

4. Metoda simplex

Deoarece știm că dacă programul în formă standard (P) are optim finit o soluție optimă va fi cu necesitate o soluție de bază și deci va fi asociată unei baze B^* , este natural să ne întrebăm cum găsim această bază optimă B^* . Traducând în termeni algebrici procedeul geometric “naiv”, descris în finalul secțiunii 3.2, rezultă următoarea procedură:

- se generează toate bazele programului (P) și se calculează soluțiile asociate acestora;
- se elimină soluțiile de bază neadmisibile și dintre cele admisibile se reține cea soluție care oferă funcției obiectiv valoarea maximă.

Nu mai insistăm asupra dezavantajelor și lipsurilor acestei scheme deoarece ele au fost deja menționate în secțiunea 3.2. “Metoda” descrisă are o alternativă care, din fericire, s-a dovedit a fi deosebit de eficientă din punct de vedere practic. Este vorba de *metoda simplex* datorată matematicianului american G.B. Dantzig (1947).

Această metodă este un *procedeu de cercetare sistematică a soluțiilor admisibile de bază* ale unui program liniar în formă standard (P). Ea presupune cunoscută o asemenea soluție, numită *soluție inițială* sau *de start* și în continuare construiește un șir de soluții admisibile de bază dealungul căruia

valoarea funcției obiectiv *crește progresiv*. Metoda simplex oferă un *test simplu de recunoaștere a optimalității* unei soluții de bază și deasemeni un test de recunoaștere a optimului infinit. *Practica numerică a arătat că numărul soluțiilor admisibile de bază efectiv generate este de regulă mult mai mic decât numărul total al acestora.*

Cu anumite precauții, ușor de îndeplinit, metoda simplex garantează convergența procesului iterativ în sensul că o bază admisibilă cercetată la un moment dat nu mai revine în iterațiile ulterioare (vezi secțiunea 4.5). Cum numărul bazelor este finit, urmează că într-un număr finit de pași se ajunge fie la soluția optimă fie la concluzia că programul are optim infinit.

Firește, în această descriere succintă, am plecat de la ipoteza cunoașterii unei soluții admisibile de bază inițiale, adică de la premiza că (P) este un program compatibil. În secțiunea 4.3 vom vedea cum se face recunoașterea incompatibilității unui program liniar.

4.1 Teoremele fundamentale ale metodei simplex

În prezentarea fundamentelor teoretice ale metodei simplex vom folosi notațiile introduse în secțiunea 3.5. În mod constant vom presupune că soluția (3.5.10), asociată bazei B , este admisibilă, adică $\bar{b}_i \geq 0, i \in I$.

Teorema A. *Dacă toți $\bar{c}_j \geq 0, j \in J$ atunci soluția (3.5.10) asociată bazei B este optimă. Dacă în plus $\bar{c}_j > 0, j \in J$, atunci ea este și unica soluție optimă a programului (P).*

Demonstrație: Fie $y = (y_1, y_2, \dots, y_n)^T \in \mathcal{A}$ o soluție admisibilă arbitrar aleasă. Deoarece $y_1 \geq 0, y_2 \geq 0, \dots, y_n \geq 0$ vom avea:

$$f(y) = \bar{f} - \sum_{j \in J} \bar{c}_j y_j \leq \bar{f} = f(\bar{x})$$

Inegalitatea de mai sus arată că, dintre toate soluțiile admisibile ale programului (P), soluția $\bar{x}$ din (3.5.10) oferă funcției obiectiv f cea mai mare valoare posibilă. Dacă costurile reduse sunt pozitive și $y \neq \bar{x}$ atunci inegalitatea de mai sus este strictă, fapt care probează unicitatea soluției optime $\bar{x}$. ■

Teorema B. *Presupunem că există indicele $k \in J$ astfel că $\bar{c}_k < 0$ și toți $\bar{a}_{ik} \leq 0, i \in I$ ($\Leftrightarrow \bar{A}^k \leq 0$). Atunci programul (P) are optim infinit.*

Demonstrație: Pornind de la soluția $\bar{x}$ din (3.5.10) construim o soluție admisibilă variabilă după cum urmează. Înlocuim în (3.5.7):

$$x_k = \theta \geq 0 ; x_j = 0, j \in J, j \neq k \quad (4.1.1)$$

Rezultă:

$$x_i = \bar{b}_i - \theta \bar{a}_{ik}, i \in I \quad (4.1.2)$$

Notăm cu $\bar{x}(\theta)$ soluția ale cărei componente sunt definite în (4.1.1) și (4.1.2). Condiția enunțului face ca $\bar{x}(\theta) \in \mathcal{A}$, $(\forall) \theta \geq 0$. Evaluăm funcția obiectiv în soluția $\bar{x}(\theta)$:

$$f(\bar{x}(\theta)) = \bar{f} - \theta \bar{c}_k \quad (4.1.3)$$

Rezultă imediat că: $\lim_{\theta \rightarrow \infty} f(\bar{x}(\theta)) = +\infty$. Deci f este nemărginită superior pe $\mathcal{A}$ și ca urmare (P) are optim infinit. ■

Teorema C Presupunem că există $k \in J$ astfel încât $\bar{c}_k < 0$ dar există și indici $i \in I$ cu $\bar{a}_{ik} > 0$. Fie $r \in I$ indicele determinat prin formula:

$$\frac{\bar{b}_r}{\bar{a}_{rk}} = \min_{i \in I, \bar{a}_{ik} > 0} \left\{ \frac{\bar{b}_i}{\bar{a}_{ik}} \right\} \quad (4.1.4)$$

Atunci grupul de coloane B' obținut din B înlocuind coloana A^r cu coloana A^k este o bază admisibilă a programului (P) și soluția $\bar{x}'$ asociată ei este cel puțin la fel de bună ca și soluția $\bar{x}$ asociată bazei B , adică $f(\bar{x}') \geq f(\bar{x})$.

Demonstrație: Din (4.1.4) rezultă că $\bar{a}_{rk} > 0$. Din $\bar{A}^k = B^{-1}A^k$ avem:

$$A^k = B\bar{A}^k = \sum_{i \in I, i \neq r} \bar{a}_{ik} A^i + \bar{a}_{rk} A^r$$

Deoarece $\bar{a}_{rk} \neq 0$ putem exprima A^r în funcție de A^i , $i \neq r$ și A^k :

$$A^r = - \sum_{i \in I, i \neq k} \frac{\bar{a}_{ik}}{\bar{a}_{rk}} A^i + \frac{1}{\bar{a}_{rk}} A^k$$

Din teorema substituției rezultă că sistemul B' , format din coloanele A^i , $i \neq r$ și A^k , este o bază a problemei (P).

Să luăm în soluția variabilă $\bar{x}(\theta)$ construită în demonstrația teoremei B:

$$\theta = \frac{\bar{b}_r}{\bar{a}_{rk}} \quad (4.1.5)$$

Din formulele (4.1.1), (4.1.2) rezultă:

$$\begin{cases} x_k = \frac{\bar{b}_r}{\bar{a}_{rk}}; x_j = 0, j \in J, j \neq k \\ x_i = \bar{b}_i - \frac{\bar{b}_r}{\bar{a}_{rk}} \bar{a}_{ik}, i \in I \end{cases}$$

Pentru $i = r$ avem $x_r = \bar{b}_r - \frac{\bar{b}_r}{\bar{a}_{rk}} \bar{a}_{rk} = 0$ așa că soluția de mai sus poate fi rescrisă astfel:

$$\begin{cases} x_j = 0, j \in J, j \neq k; x_r = 0 \\ x_i = \bar{b}_i - \frac{\bar{b}_r}{\bar{a}_{rk}} \bar{a}_{ik}, i \in I, i \neq r; x_k = \frac{\bar{b}_r}{\bar{a}_{rk}} \end{cases} \quad (4.1.6)$$

Notăm cu $\bar{x}'$ soluția ale cărei componente sunt definite în (4.1.6). Vom observa mai întâi că $\bar{x}'$ este soluție admisibilă a problemei (P), adică are toate componentele nenegative. Într-adevăr:

- dacă $\bar{a}_{ik} \leq 0$ atunci $x_i \geq 0$;

- dacă $\bar{a}_{ik} > 0$ atunci $\bar{b}_i - \frac{\bar{b}_r}{\bar{a}_{rk}} \bar{a}_{ik} \geq 0 \Leftrightarrow \frac{\bar{b}_i}{\bar{a}_{ik}} \geq \frac{\bar{b}_r}{\bar{a}_{rk}}$ care are loc

conform alegerii indicelui r (vezi (4.1.4)).

În al doilea rând remarcăm că $\bar{x}'$ este soluția asociată bazei B' . Într-adevăr, din (4.1.6) rezultă că în $\bar{x}'$ toate variabilele secundare în raport cu baza B' au valoarea 0. Afirmatia rezultă acum din unicitatea soluției asociate unei baze.

Utilizând formulele (4.1.3) și (4.1.5) obținem:

$$f(\bar{x}') = \bar{f} - \frac{\bar{b}_r}{\bar{a}_{rk}} \bar{c}_k \geq \bar{f} = f(\bar{x}) \quad (4.1.7)$$

În concluzie, $\bar{x}'$ este cel puțin la fel de bună ca și $\bar{x}$.

■

Observații: 1) Bazele B și B' apărute în enunțul teoremei C diferă una de alta printr-o singură coloană a matricii A . Două baze ale programului (P) cu această proprietate se vor numi în continuare *baze vecine* și tot vecine se vor numi și soluțiile asociate.

2) Teoremele B și C afirmă că dacă o soluție de bază admisibilă $\bar{x}$ nu satisface criteriul de optimalitate al teoremei A atunci sau (P) are optim infinit sau, printre soluțiile *vecine* cu $\bar{x}$ există cel puțin una la fel de bună ca și $\bar{x}$ dacă nu chiar mai bună.

Recapitulând materialul expus în această secțiune constatăm că testarea optimalității soluției $\bar{x}$ asociate bazei B presupune cunoașterea formei explicite (3.5.7) a problemei inițiale în raport cu baza B. Mai mult, dacă $\bar{x}$ nu verifică criteriul de optimalitate al teoremei A, construcția soluției de bază mai bune $\bar{x}'$ se face cu ajutorul elementelor aceleiași forme (3.5.7). În consecință, testarea optimalității soluției $\bar{x}'$ asociată bazei vecine B' va necesita cunoașterea formei explicite a problemei inițiale în raport cu noua bază B':

$$(P_{B'}) \begin{cases} \max f = \bar{f} - \sum_{j \in J, j \neq k} \bar{c}'_j x_j - \bar{c}'_r x_r \\ x_i + \sum_{j \in J, j \neq k} \bar{a}'_{ij} x_j + \bar{a}'_{ir} x_r = \bar{b}'_i \quad i \in I, i \neq r \\ x_k + \sum_{j \in J, j \neq k} \bar{a}'_{kj} x_j + \bar{a}'_{kr} x_r = \bar{b}'_k \\ x_1, x_2, \dots, x_n \geq 0 \end{cases} \quad (4.1.8)$$

(Vom remarca faptul că unele elemente din (4.1.8) sunt deja evaluate! Astfel, termenii liberi, adică valorile noilor variabile bazice, sunt dați de (4.1.6) în timp ce constanta $\bar{f}'$, care este valoarea funcției obiectiv în noua soluție de bază $\bar{x}'$, a fost obținută în (4.1.7).)

Firește, (4.1.8) se poate obține ca și (3.5.7) prin înmulțirea la stânga a sistemului original de restricții $Ax = b$ cu matricea inversă $(B')^{-1}$. Putem deduce (4.1.8) direct din (3.5.7) cu ajutorul următoarelor operații:

Din ecuația r a sistemului (3.5.7):

$$x_r + \sum_{j \neq k} \bar{a}_{rj} x_j + \bar{a}_{rk} x_k = \bar{b}_r$$

explicităm x_k , împărțind relația cu $\bar{a}_{rk}$:

$$x_k + \sum_{j \neq k} \frac{\bar{a}_{rj}}{\bar{a}_{rk}} x_j + \frac{1}{\bar{a}_{rk}} x_r = \frac{\bar{b}_r}{\bar{a}_{rk}} \quad (4.1.9)$$

Substituim x_k dat de (4.1.9) în celelalte ecuații ale sistemului din (3.5.7). Obținem:

$$x_i + \sum_{j \neq k} \left(\bar{a}_{ij} - \frac{\bar{a}_{rj}}{\bar{a}_{rk}} \right) x_j - \frac{\bar{a}_{ik}}{\bar{a}_{rk}} x_r = \bar{b}_i - \frac{\bar{b}_r}{\bar{a}_{rk}} \bar{a}_{ik}, \quad i \in I, i \neq r \quad (4.1.10)$$

Ecuațiile (4.1.9) , (4.1.10) reprezintă sistemul $Ax = b$ explicitat în noile variabile bazice x_i $i \in I, i \neq r$ și x_k .

Mai departe substituim același x_k și în expresia funcției obiectiv din (3.5.7). Găsim:

$$f = \left(\bar{f} - \frac{\bar{b}_r}{\bar{a}_{rk}} \bar{c}_k \right) - \sum_{j \neq k} \left(\bar{c}_j - \frac{\bar{a}_{rj}}{\bar{a}_{rk}} \bar{c}_k \right) x_j + \frac{\bar{c}_k}{\bar{a}_{rk}} x_r \quad (4.1.11)$$

Astfel, am exprimat funcția obiectiv cu ajutorul noilor variabile nebazice x_j $j \in J, j \neq k$ și x_r .

Prin urmare (4.1.9) , (4.1.10) , (4.1.11) constituie forma explicită a problemei originale în raport cu noua bază B' . Identificând coeficienții din aceste ecuații cu coeficienții corespunzători din (4.1.8) rezultă formulele:

$$\begin{aligned} \bar{a}'_{ij} &= \bar{a}_{ij} - \frac{\bar{a}_{rj}}{\bar{a}_{rk}} \bar{a}_{ik} & i \in I, i \neq r \\ & & j \in J, j \neq k \end{aligned} \quad ; \quad \bar{a}'_{ir} = -\frac{\bar{a}_{ik}}{\bar{a}_{rk}} \quad i \in I, i \neq r$$

$$\bar{a}'_{kj} = \frac{\bar{a}_{rj}}{\bar{a}_{rk}} \quad j \in J, j \neq k \quad ; \quad \bar{a}'_{kr} = \frac{1}{\bar{a}_{rk}} \quad (4.1.12)$$

$$\bar{f}' = \bar{f} - \frac{\bar{b}_r}{\bar{a}_{rk}} \bar{c}_k \quad ; \quad \bar{c}'_j = \bar{c}_j - \frac{\bar{a}_{rj}}{\bar{a}_{rk}} \bar{c}_k \quad j \in J, j \neq k \quad ; \quad \bar{c}'_r = -\frac{\bar{c}_k}{\bar{a}_{rk}}$$

cunoscute și sub numele de *formule de schimbare a bazei*.

4.2 Algoritmul simplex

Rezolvarea efectivă a unui program liniar în formă standard (P) se face cu ajutorul *algoritmului simplex*. Acesta este un *pachet invariabil de instrucțiuni logice și de calcul care, aplicate unei soluții admisibile de bază a programului (P), stabilește dacă soluția respectivă este optimă și, în caz contrar, pune în evidență situația de optim infinit sau construiește efectiv o soluție admisibilă de bază mai bună decât cea curentă*.

Să considerăm o bază admisibilă B precum și forma explicită (P_B) a programului (P) în raport cu această bază. (vezi notațiile secțiunii 3.5)

În raport cu aceste date de intrare, conținutul unei iterații simplex este următorul:

Pasul 1. (*Test de optimalitate*) Dacă toți $\bar{c}_j \geq 0, j \in J$ soluția de bază curentă (adică soluția (3.5.10), asociată bazei B) este optimă: STOP. Altminteri:

Pasul 2. Se alege indicele nebazic $k \in J$ astfel ca:

$$\bar{c}_k = \min_{j \in J} \bar{c}_j \quad (4.2.1)$$

Pasul 3. Dacă:

$$\bar{a}_{ik} \leq 0, i \in I \quad (\Leftrightarrow \bar{A}^k \leq 0)$$

programul (P) are optim infinit: STOP. Altminteri:

Pasul 4. Se determină indicele bazic $r \in I$ cu formula:

$$\frac{\bar{b}_r}{\bar{a}_{rk}} = \min_{i \in I | \bar{a}_{ik} > 0} \left(\frac{\bar{b}_i}{\bar{a}_{ik}} \right) \quad (4.2.2)$$

Pasul 5. Se construiește forma explicită a programului (P) în raport cu baza B' dedusă din B prin înlocuirea coloanei A^r cu coloana A^k . Se revine la pasul 1 în cadrul unei noi iterații.

Observații 1) La pasul 2 avem cu necesitate $\bar{c}_k < 0$. Despre coloana nebazică A^k vom spune că *intră în baza curentă*. În demonstrația teoremei C s-a arătat că variația valorii funcției obiectiv la schimbarea coloanei bazice A^r cu coloana A^k este dată de formula:

$$f(\bar{x}') - f(\bar{x}) = -\left(\frac{\bar{b}_r}{\bar{a}_{rk}}\right)\bar{c}_k \quad (4.2.3)$$

Cum $\frac{\bar{b}_r}{\bar{a}_{rk}} \geq 0$ (vezi (4.2.2)) urmează că introducerea în bază a oricărei coloane nebazice A^j cu $\bar{c}_j < 0$ îmbunătățește valoarea curentă a funcției obiectiv. Alegerea coloanei nebazice care va intra în baza curentă după formula (4.2.1) asigură funcției obiectiv *cea mai mare viteză de variație* și în general conduce la terminarea algoritmului în mai puține iterații.

2) Se poate arăta ușor că dacă situația descrisă în pasul 3 are loc atunci vectorul $w = (w_1, w_2, \dots, w_n)^T$ definit prin:

$$w_i = -\bar{a}_{ik}, i \in I; w_j = 0, j \in J, j \neq k; w_k = 1 \quad (4.2.4)$$

este o rază extremă a mulțimii poliedrale $\mathcal{A}_P$.

3) Despre coloana bazică A^r al cărei indice se determină cu formula (4.4.2) vom spune că *părăsește baza curentă*. Alegerea ei după formula amintită asigură admisibilitatea soluției asociate noii baze B' .

4) Elementul $\bar{a}_{rk}$, unde k este indicele coloanei care intră în bază iar r este indicele coloanei care iese din bază se numește *pivot* și operația de calculare a elementelor formei explicite în raport cu baza B' din elementele formei explicite în raport cu baza veche B prin aplicarea formulelor de schimbare a bazei (4.1.12) poartă numele de *pivotare gaussiană*.

5) În principiu, problemele de minimizare se reduc la cele de maximizare în baza relației (1.3.1). Algoritmul simplex descris mai sus este aplicabil și problemelor de minimizare cu următoarele mici modificări:

- în pasul 1: soluția curentă este optimă dacă $\bar{c}_j \leq 0, j \in J$;

• în pasul 2: pentru determinarea indicelui coloanei nebazice care intră în baza curentă se va utiliza formula

$$\bar{c}_k = \max_{j \in J} \bar{c}_j$$

De această dată vom avea $\bar{c}_k > 0$.

6) Să presupunem că baza curentă B este optimală (adică soluția asociată ei verifică testul de optimalitate) și că există indici nebazici $j \in J$ cu $\bar{c}_j = 0$. Formula (4.2.3) arată că introducerea în baza curentă a oricărei coloane A^k pentru care $\bar{c}_k = 0$ conduce la soluții de bază la fel de bune ca și cea curentă, deci optime. *Se poate arăta că dacă $\bar{x}^1, \bar{x}^2, \dots, \bar{x}^p$, $p \geq 2$ sunt toate soluțiile de bază optime ale programului (P) atunci acesta are o infinitate de soluții optime care au forma:*

$$\bar{x} = \alpha_1 \bar{x}^1 + \alpha_2 \bar{x}^2 + \dots + \alpha_p \bar{x}^p \quad \text{cu} \quad \alpha_1, \alpha_2, \dots, \alpha_p \geq 0 \text{ și } \alpha_1 + \alpha_2 + \dots + \alpha_p = 1$$

(altfel spus, orice soluție optimă este o combinație convexă a soluțiilor optime de bază)

7) În rezolvarea manuală a programelor liniare (firește de mici dimensiuni) se utilizează tabelele simplex (3.5.1) asociate diferitelor baze cercetate. Aceste tabele se deduc unul din altul prin *pivotare gaussiană*.

În continuare vom aplica algoritmul simplex pe o problemă simplă, în două variabile, pentru a putea ilustra grafic modul în care acționează procedura.

Exemplul 4.2.1 Considerăm programul:

$$(P) \begin{cases} \max f = 2x_1 + x_2 \\ x_1 - x_2 \leq 4 ; 3x_1 - x_2 \leq 18 ; -x_1 + 2x_2 \leq 6 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

a cărui mulțime de soluții admisibile $\mathcal{A}_P$ este vizualizată în figura 4.2.1.

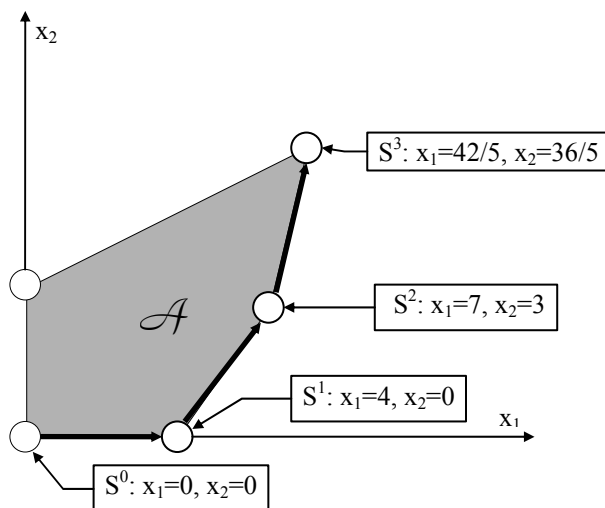


Figura 4.2.1

Aducem (P) la forma standard adăugând variabilele de abatere x_3, x_4, x_5 :

$$(FSP) \begin{cases} \max f = 2x_1 + x_2 \\ x_1 - x_2 + x_3 = 4 \\ 3x_1 - x_2 + x_4 = 18 \\ -x_1 + 2x_2 + x_5 = 6 \\ x_j \geq 0, j = 1, \dots, 5 \end{cases}$$

Se observă că matricea A a coeficienților formei standard (FSP) conține baza unitară $E = [A^3, A^4, A^5]$ a cărei soluție asociată:

$$x_1 = 0, x_2 = 0, x_3 = 4, x_4 = 18, x_5 = 6$$

este admisibilă. În continuare sunt date tabelele rezultate prin aplicarea algoritmului simplex (tabelele 4.2.1 - 4.2.4). În partea dreaptă am indicat formele explicite ale problemei (FSP) în raport cu bazele cercetate. În trei iterații s-a obținut soluția optimă $x_1^* = \frac{42}{5}$, $x_2^* = \frac{36}{5}$, valoarea maximă a funcției obiectiv fiind 24; variabilele de abatere au în soluția optimă valorile $x_3^* = \frac{14}{5}$, $x_4^* = x_5^* = 0$.

c^B	B	VVB	2	1	0	0	0
			A^1	A^2	A^3	A^4	A^5
0	A^3	4	1	-1	1	0	0
0	A^4	18	3	-1	0	1	0
0	A^5	6	-1	2	0	0	1
	f	0	-2	-1	*	*	*
22	A^1	4	1	-1	1	0	0
0	A^4	6	0	2	-3	1	0
0	A^5	10	0	1	1	0	1
	f	8	*	-3	2	*	*
2	A^1	7	1	0	-1/2	1/2	0
1	A^2	3	0	1	-3/2	1/2	0
0	A^5	7	0	0	5/2	-1/2	1
	f	17	*	*	-5/2	3/2	*
2	A^1	42/5	1	0	0	2/5	1/5
1	A^2	36/5	0	1	0	1/5	3/5
0	A^3	14/5	0	0	1	-1/5	2/5
	f	24	*	*	*	1	1

$x_1 - x_2 + x_3$	=	4
$3x_1 - x_2 + x_4$	=	18
$-x_1 + 2x_2 + x_5$	=	6
$-2x_1 - x_2 + f$	=	0
$x_1 - x_2 + x_3$	=	4
$2x_2 - 3x_3 + x_4$	=	6
$x_2 + x_3 + x_5$	=	10
$-3x_2 + 2x_3 + f$	=	8
$x_1 - 1/2x_3 + 1/2x_4$	=	7
$x_2 - 3/2x_3 + 1/2x_4$	=	3
$5/2x_3 - 1/2x_4 + x_5$	=	7
$-5/2x_3 + 3/2x_4 + f$	=	17
$x_1 + 2/5x_4 + 1/2x_5$	=	42/5
$x_2 + 1/5x_4 + 3/5x_5$	=	36/5
$x_3 - 1/5x_4 + 2/5x_5$	=	14/5
$x_4 + x_5 + f$	=	24

Tabelele 4.2.1 - 4.2.4

Soluția	Baza	Componentele soluției de bază generată de algoritm					Soluția programului (P)	Valoarea funcției obiectiv
		x_1	x_2	x_3	x_4	x_5		
S^0	$B^0 = [A^3, A^4, A^5]$	0	0	4	18	6	(0, 0)	0
S^1	$B^1 = [A^1, A^4, A^5]$	4	0	0	6	10	(4, 0)	8
S^2	$B^2 = [A^1, A^2, A^5]$	7	3	0	0	7	(7, 3)	17
S^3	$B^3 = [A^1, A^2, A^3]$	42/5	36/5	14/5	0	0	(42/5, 36/5)	24

Tabelul 4.2.5

Este util să recapitulăm într-un tabel soluțiile admisibile de bază ale programului (FSP), efectiv generate de algoritm, împreună cu soluțiile

programului (P), asociate prin corespondența Φ din finalul secțiunii 3.3 (vezi tabelul 4.2.5).

Urmărind imaginea grafică (fig. 4.2.1), deducem sensul geometric al algoritmului simplex: procedura pleacă dintr-un vârf al mulțimii soluțiilor admisibile apoi se deplasează către un vârf “vecin” mai bun ș.a.m.d. până la găsirea soluției optime.

4.3 Determinarea unei soluții admisibile de start

După cum s-a specificat, aplicarea algoritmului simplex necesită cunoașterea unei baze admisibile de start precum și a formei explicite asociate acesteia, celelalte forme explicite deducându-se una din alta prin pivotare gaussiană. Chiar și pentru probleme de dimensiuni mici, găsirea unei asemenea baze de start prin simpla inspectare a coloanelor matricii A , se dovedește a fi o treabă complicată, ne mai vorbind de faptul că este posibil ca problema să nu aibe soluții admisibile. În plus, să nu uităm că teoria metodei simplex s-a bazat esențial pe ipoteza că restricțiile problemei sunt liniar independente, lucru iarăși greu de verificat în practică. Pentru obținerea formei explicite inițiale avem nevoie de câteva pregătiri.

Vom spune că programul în formă standard (P) este în *formă bună* dacă matricea A conține o *submatrice unitate* de ordinul m (= numărul restricțiilor) iar termenii liberi sunt *nenegativi*. Dacă este așa, (P) satisface condiția (3.5.1) iar soluția asociată bazei unitare este admisibilă și poate fi considerată ca soluție de start pentru aplicarea algoritmului simplex. Dacă (P) nu este în formă bună, el se poate aduce la această formă, notată (FBP), în felul următor:

- În caz că unele restricții ale programului inițial au termeni liberi negativi înmulțim aceste restricții cu -1 ; în acest fel toți termenii liberi ai restricțiilor problemei de rezolvat vor fi ≥ 0 .
- Se aduce problema la forma standard adăugând variabile de abatere în restricțiile inegalități.
- Dacă matricea programului rezultat nu conține toate coloanele matricii unitate de ordinul m , în anumite restricții se vor adăuga noi variabile nenegative pentru crearea coloanelor lipsă; aceste noi variabile se numesc

variabile artificiale și, spre deosebire de cele de abatere, apar și în funcția obiectiv cu un coeficient comun, foarte mare în valoare absolută. Coeficientul va fi negativ dacă funcția obiectiv se maximizează și pozitiv în caz contrar.

Se observă imediat că dacă programul inițial (P) este compatibil, soluțiile sale admisibile se identifică cu acele soluții ale formei bune în care variabilele artificiale au valoarea zero. Prin faptul că variabilele artificiale sunt însoțite în expresia funcției obiectiv de niște “penalizări” foarte mari, metoda simplex este “înstruită” să caute tocmai asemenea soluții! Și astfel, rezolvarea formei bune ne conduce la unul din următoarele cazuri:

1. Forma bună are optim infinit. Atunci și programul inițial are optim infinit.

2. Forma bună are optim finit dar în soluția optimă cel puțin o variabilă artificială are valoare nenulă. Atunci programul original este incompatibil.

3. Forma bună are optim finit și în soluția optimă toate variabilele artificiale au valoarea zero. Ignorând valorile acestor variabile se obține soluția optimă a programului inițial.

Exemplul 4.3.1 Considerăm următorul program împreună cu forma sa standard:

$$(P) \begin{cases} (\max) f = 2x_1 + 3x_2 \\ 2x_1 + x_2 \geq 40 \\ x_1 + 3x_2 \geq 30 \\ x_1 + x_2 \leq 30 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \Rightarrow (FSP) \begin{cases} (\max) f = 2x_1 + 3x_2 \\ 2x_1 + x_2 - x_3 = 40 \\ x_1 + 3x_2 - x_4 = 30 \\ x_1 + x_2 + x_5 = 30 \\ x_j \geq 0, j = 1, \dots, 5 \end{cases}$$

Se constată că (FSP) nu este în forma bună neconținând decât un singur vector al matricii unitare de ordinul 3. Vom crea o asemenea matrice introducând în primele două restricții variabilele artificiale x_6 și x_7 . Obținem programul:

$$\begin{aligned}
 & (max) f = 2x_1 + 3x_2 - Mx_6 - Mx_7 \\
 (FBP) \left\{ \begin{array}{l}
 2x_1 + x_2 - x_3 + x_6 = 40 \\
 x_1 + 3x_2 - x_4 + x_7 = 30, \quad M \gg 0 \\
 x_1 + x_2 + x_5 = 30 \\
 x_j \geq 0, j = 1, \dots, 7
 \end{array} \right.
 \end{aligned}$$

Putem aplica algoritmul simplex programului (FBP) plecând de la baza unitară $E = [A^6, A^7, A^5]$ și de la soluția asociată acesteia:

$$x_1 = x_2 = x_3 = x_4 = 0 \quad x_5 = 30, x_6 = 40, x_7 = 30$$

După trei iterații (vezi tabelele 4.3.1 - 4.3.4) se obține soluția optimă a programului (FBP) în care variabilele artificiale x_6, x_7 au valoarea zero. Ignorând aceste valori se obține soluția optimă a formei standard (FSP) și implicit soluția optimă a programului original (P):

$$x_1^* = 10, x_2^* = 20 \quad f_{max} = 80$$

variabilele de abatere având valorile: $x_3^* = 0, x_4^* = 40, x_5^* = 0$.

c^B	B	VVB	2 A^1	3 A^2	0 A^3	0 A^4	0 A^5	-M A^6	-M A^7
-M	A^6	40	2	1	-1	0	0	1	0
-M	A^7	30	1	3	0	-1	0	0	1
0	A^5	30	1	1	0	0	1	0	0
	f	-70M	-3M-2	-4M-3	M	M	*	*	*
-M	A^6	30	5/3	0	-1	1/3	0	1	-1/3
3	A^2	10	1/3	1	0	-1/3	0	0	1/3
0	A^5	20	2/3	0	0	1/3	1	0	-1/3
	f	-30M+30	-5/3M-1	*	M	-M/3-1	*	*	4M/3+1
2	A^1	18	1	0	-3/5	1/5	0	3/5	-1/5
3	A^2	4	0	1	1/5	-2/5	0	-1/5	2/5
0	A^5	8	0	0	2/5	1/5	1	-2/5	-1/5
	f	48	*	*	-3/5	-4/5	*	M+3/5	M+4/5
2	A^1	10	1	0	-1	0	-1	1	0
3	A^2	20	0	1	1	0	2	-1	0
0	A^4	40	0	0	2	1	5	-2	-1
	f	80	*	*	1	*	4	M-1	M

Tabelele 4.3.1 - 4.3.4

Punctele din $\mathbb{R}^2$ corespunzătoare celor patru soluții generate de algoritm sunt $S^0 = (0,0)$, $S^1 = (0,10)$, $S^2 = (18,4)$, $S^3 = (10,20)$.

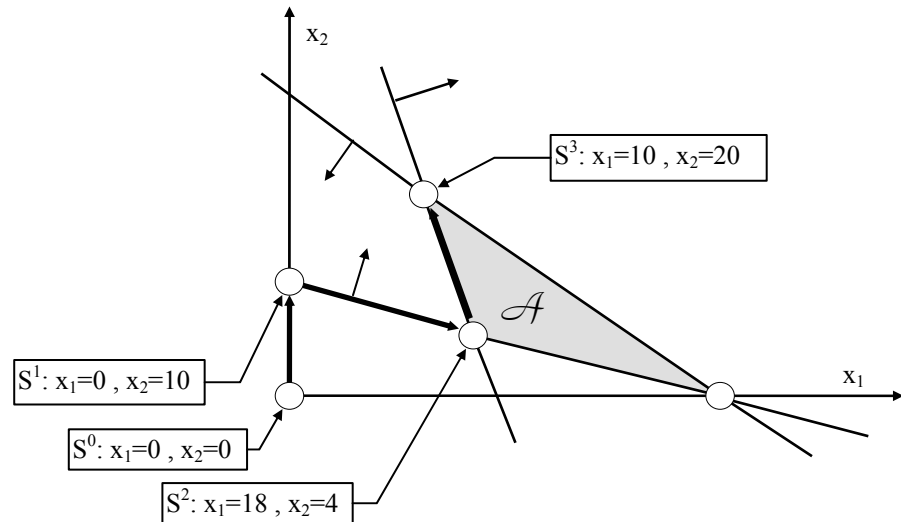


Figura 4.3.1

Urmărind fig. 4.3.1 se constată că punctele S^0 și S^1 , ce corespund unor soluții ale programului (FBP) în care cel puțin o variabilă artificială are valoare nenulă, sunt în afara mulțimii $\mathcal{A}_P$ în timp ce S^2 și S^3 , corespunzătoare unor soluții în care $x_6 = x_7 = 0$, sunt vârfuri ale acestei mulțimi.

Pentru determinarea unei soluții admisibile de bază de start, în situația în care programul inițial (P) nu este în formă bună, se poate aplica și așa numita *metodă a celor două faze*:

- se introduc în restricții variabile artificiale, bineînțeles acolo unde este cazul, în scopul formării unei baze unitare de start (termenii liberi ai restricțiilor se presupun a fi nenegativi);
- în faza I se minimizează suma w a variabilelor artificiale introduse. Deoarece și aceste variabile sunt supuse condiției de nenegativitate, urmează că $(\min)w \geq 0$. În caz că $(\min)w > 0$ este clar că programul inițial (P) este incompatibil; dacă $(\min)w = 0$ se trece la:

• *faza a II-a, în care se optimizează funcția obiectiv a programului inițial (P) plecând de la soluția de bază rezultată la finele fazei I. (Atenție, la începutul acestei faze, vom avea grijă să recalculăm costurile reduse $\bar{c}_j$ în raport cu coeficienții funcției obiectiv din (P)!)*

4.4. Inversa bazei curente. Soluția optimă a problemei duale

Să considerăm un program liniar (P) în formă bună. Renumeroțind convenabil variabilele programului, matricea coeficienților sistemului de restricții are forma:

$$A = [A', E]$$

unde E este matricea unitate de ordinul $m = \text{numărul restricțiilor din (P)}$. După cum am văzut, E se ia ca bază de start în procesul rezolvării programului (P)

prin algoritmul simplex. Dacă B este o bază cercetată de algoritm, atunci matricea formei explicite asociată bazei B va fi:

$$B^{-1}A = [B^{-1}A', B^{-1}]$$

Cum matricea $B^{-1}A$ apare în “corpul mare” al tabelului simplex corespunzător bazei B, deducem următoarea concluzie importantă:

La fiecare iterație, algoritmul simplex pune în evidență inversa B^{-1} a bazei curente. Ea este formată din coloanele $\bar{A}^j$ corespunzătoare coloanelor unitare A^j care au format baza de start.

Exemplul 4.4.1 Ne referim la programul (P) rezolvat în exemplul 4.3.1. Deoarece baza de start a fost matricea unitate $E = [A^6, A^7, A^5]$, inversa bazei optime $B = [A^1, A^2, A^4]$ se “citește” din ultimul tabel simplex:

$$B^{-1} = [\bar{A}^6, \bar{A}^7, \bar{A}^5] = \begin{bmatrix} 1 & 0 & -1 \\ -1 & 0 & 2 \\ -2 & -1 & 5 \end{bmatrix}$$

O calitate remarcabilă a algoritmului simplex este aceea că produce nu numai soluția optimă a problemei căreia i se aplică ci și soluția optimă a problemei duale.

Să considerăm un cuplu de probleme în dualitate în care una este în formă standard (vezi secțiunea 2.2):

$$(P) \begin{cases} \max f(x) = cx \\ Ax = b \\ x \geq 0 \end{cases} \quad (Q) \begin{cases} \min g(u) = ub \\ uA \geq c \\ u \text{ oarecare} \end{cases}$$

Cu notațiile secțiunii 3.5 avem următoarea;

Teorema 4.4.1 *Presupunem că (P) are o soluție optimă x^* asociată unei baze B. Atunci:*

$$u^* = c^B B^{-1} \quad (4.4.1)$$

este o soluție optimă a problemei duale (Q).

Demonstrație: Vom arăta că u^* satisface restricțiile $uA^j \geq c_j$, $j = 1, \dots, n$ ale problemei duale (Q). Într-adevăr $u^*A^j - c_j = c^B B^{-1}A^j - c_j = \bar{c}_j$, conform (3.5.6). Avem $\bar{c}_j \geq 0$, $j = 1, \dots, n$, deoarece x^* este prin ipoteză soluția optimă a programului (P), astfel că u^* este o soluție a dualei (Q). Mai departe $f(x^*) = g(u^*) = c^B B^{-1}b$. Concluzia teoremei decurge acum din teorema 2.3.1. ■

Practic, pentru determinarea soluției optime duale se procedează astfel. Am arătat că rezolvarea unei probleme în formă standard se reduce la rezolvarea unei probleme de același tip a cărei matrice conține o submatrice unitate având ordinul egal cu numărul restricțiilor.

Atunci inversa bazei optimale B^{-1} se citește în tabelul simplex asociat acestei baze pe coloanele $\bar{A}^j$ corespunzătoare vectorilor care au format baza unitară de start!

Extrăgând B^{-1} din tabel se poate aplica formula (4.4.1).

Procedeul descris este valabil și pentru un cuplu general de probleme în dualitate (P,Q). Într-adevăr, după cum am văzut deja, aducerea problemei (P) la o formă (P_1) convenabilă aplicării algoritmului simplex se face prin adăugare

de variabile: de abatere și/sau artificiale. În consecință, duala (Q_1) a problemei (P_1) va avea aceleași variabile și aceeași funcție obiectiv ca și duala (Q) a problemei inițiale (P) , diferența constând în numărul restricțiilor și în condițiile impuse variabilelor. Se poate arăta fără dificultate că această “diferență” nu înseamnă altceva decât două modalități de scriere a uneia și aceleiași probleme, altfel spus (Q) și (Q_1) , în esență, coincid! În acest fel, teorema de dualitate 2.3.3 este probată.

Exemplul 4.4.2 Ilustrăm cele de mai sus, determinând soluția optimă a programului liniar:

$$(Q) \begin{cases} \min g(u) = 40u_1 + 30u_2 + 30u_3 \\ 2u_1 + u_2 + u_3 \geq 2 ; u_1 + 3u_2 + u_3 \geq 3 \\ u_1 \leq 0, u_2 \leq 0, u_3 \geq 0 \end{cases}$$

care este dualul programului (P) rezolvat în exemplul 4.3.1. După adăugarea variabilelor de abatere și a celor artificiale, din (P) s-a obținut programul $(P_1) = (FBP)$ al cărui dual este:

$$(Q_1) \begin{cases} \min g(u) = 40u_1 + 30u_2 + 30u_3 \\ 2u_1 + u_2 + u_3 \geq 2 \\ u_1 + 3u_2 + u_3 \geq 3 \\ -u_1 \geq 0 \\ -u_2 \geq 0 \\ u_3 \geq 0 \\ u_1 \geq -M \\ u_2 \geq -M \\ u_1, u_2, u_3 \text{ f.r.s.} \end{cases}$$

Restricția $u_3 \geq 0$ este de fapt condiția de nenegativitate impusă variabilei u_3 în (Q) iar $-u_1 \geq 0$, $-u_2 \geq 0$ înseamnă $u_1 \leq 0$, $u_2 \leq 0$. Ultimele două restricții din (Q₁) sunt superflue pentru că M este prin definiție $\gg 0$. Prin urmare (Q) și (Q₁) coincid. Din tabelul simplex 4.3.4 extragem inversa bazei optimale comune $B = [A^1, A^2, A^4]$ a programelor (P) și (P₁) - vezi exemplul precedent - astfel că soluția optimă a problemei duale (Q) este :

$$u^* = c^B B^{-1} = \begin{bmatrix} 2 & 3 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ -1 & 0 & 2 \\ -2 & -1 & 5 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 4 \end{bmatrix}$$

adică: $u_1^* = -1, u_2^* = 0, u_3^* = 4$.

Observație: Folosind notațiile din 3.5 punem în evidență componentele vectorului linie $c^B B^{-1}$:

$$\pi_j = c^B \bar{A}^j = \sum_{i \in I} c_i \bar{a}_{ij} \quad j \in J$$

Pentru coloanele unitare $\bar{A}^i$ corespunzătoare vectorilor A^i din baza curentă punem:

$$\pi_i = c_i \quad i \in I$$

Din (3.5.6) rezultă atunci că:

$$\bar{c}_j = \pi_j - c_j \quad j=1, \dots, n$$

și deci mărimile π_j sunt efectiv calculate în procesul evaluării coeficienților $\bar{c}_j$! Acesta este și motivul pentru care de multe ori mărimile π_j sunt puse în evidență într-o linie separată plasată în tabelul simplex deasupra liniei test. Cu aceste pregătiri, formula (4.4.1) arată că:

Soluția optimă a problemei duale este dată de coeficienții π_j din tabelul simplex al problemei primale, corespunzători coloanelor unitare care au format baza de start.

4.5 Convergența algoritmului simplex

Se arată ușor că dacă minimul din (4.2.2) *nu este unic* atunci noua soluție de bază $\bar{x}'$ este *degenerată*, adică are un număr de componente nenule $< m$. Această situație este delicată prin faptul că poate implica neconvergența algoritmului. Mai precis, este posibil ca algoritmul să genereze un șir de soluții de bază neoptimale $\bar{x}^1, \bar{x}^2, \dots, \bar{x}^p$, dealungul căruia valoarea funcției obiectiv staționează, adică $f(\bar{x}^1) = f(\bar{x}^2) = \dots = f(\bar{x}^p)$, astfel încât $\bar{x}^p = \bar{x}^1$!! Fenomenul descris, numit *ciclare*, deși teoretic posibil, nu a fost întâlnit în nici o aplicație practică, existența lui fiind probată doar prin câteva exemple “artificial” construite.

Exemplul 4.5.1 [Beale] Se consideră programul liniar:

$$\begin{cases} x_1 + \frac{1}{4}x_4 - 8x_5 - x_6 + 9x_7 = 0 \\ x_2 + \frac{1}{2}x_4 - 12x_5 - \frac{1}{2}x_6 + 3x_7 = 0 \\ x_3 + x_6 = 1 \\ (\max)f = \frac{3}{4}x_4 - 20x_5 + \frac{1}{2}x_6 - 6x_7 \end{cases}$$

O bază admisibilă de start este $B^0 = E = [A^1, A^2, A^3]$ a cărei soluție asociată x^0 este degenerată. Într-adevăr, variabilele bazice au valorile:

$$x_1 = 0, x_2 = 0, x_3 = 1 \text{ (toate celelalte variabile fiind nule)}$$

Propunem cititorului să arate că succesiunea de baze admisibile:

$$B^1 = [A^4, A^2, A^3], B^2 = [A^4, A^5, A^3], B^3 = [A^6, A^5, A^3], B^4 = [A^6, A^7, A^3]$$

$$B^5 = [A^6, A^2, A^3], B^6 = [A^4, A^2, A^3]$$

poate fi dedusă din B^0 prin aplicarea regulilor algoritmului simplex. Se constată fără dificultate că valoarea funcției obiectiv în soluțiile de bază asociate este zero și că nici una din aceste soluții, toate degenerate, nu verifică criteriul de optimalitate. Pe de altă parte observăm că baza B^6 este identică cu baza B^1 și deci algoritmul intră într-un ciclu infinit fără a putea determina soluția optimă (care există și este unică după cum vom vedea).

Pentru problemele ale căror soluții de bază admisibile sunt toate nedegenerate convergența algoritmului este asigurată de următoarea :

Teorema 4.5.1 *Dacă programul în formă standard (P) este compatibil și toate soluțiile sale admisibile de bază sunt nedegenerate atunci aplicarea algoritmului simplex descris în 4.2 se termină într-un număr finit de iterații, fie cu găsirea soluției optime, fie cu concluzia că programul are optim infinit.*

Demonstrație: Fie $\bar{x}^0, \bar{x}^1, \bar{x}^2, \dots$ soluțiile cercetate în cursul aplicării algoritmului, $\bar{x}^0$ fiind soluția inițială. Avem $f(\bar{x}^0) \leq f(\bar{x}^1) \leq f(\bar{x}^2) \leq \dots$. Ipoteza nedegenerării precum și formula (4.2.3) arată că de fapt $f(\bar{x}^0) < f(\bar{x}^1) < f(\bar{x}^2) \dots$ și deci nici o soluție admisibilă de bază nu va fi cercetată de două ori. Concluzia teoremei rezultă acum din faptul că numărul soluțiilor de bază este finit (vezi secțiunea 3.5). ■

Din fericire, există proceduri de evitare a ciclării care constau într-o mică modificare a regulii (4.2.2). Ele sunt încorporate în orice pachet de programe destinat rezolvării problemelor de programare liniară. O asemenea metodă va fi descrisă în cele ce urmează. Sunt necesare câteva pregătiri.

Definiție Spunem că vectorul $x = (x_1, x_2, \dots, x_n)$ este lexicografic pozitiv și scriem $x \succ 0$ dacă $x \neq 0$ și prima componentă nenulă a lui x este pozitivă.

În mulțimea vectorilor din R^n introducem relația binară:

$$x \succ y \text{ dacă } x - y \succ 0$$

Se verifică ușor afirmațiile:

- Dacă $x \succ y$ și $y \succ z$ atunci $x \succ z$ (tranzitivitate).
- Oricare ar fi $x, y \in R^n$ una și numai una din următoarele relații are loc: $x = y$, $x \succ y$, $y \succ x$.

Prin urmare relația $\succ$ este o relație de ordine totală pe R^n care extinde relația de ordine naturală din R . În plus ea este compatibilă cu structura liniară a spațiului R^n în sensul că:

- $x \succ y \Rightarrow x + z \succ y + z$
- $x \succ y$ și $\lambda > 0 \Rightarrow \lambda x \succ \lambda y$

Să considerăm acum problema de programare liniară în formă standard:

$$(P) \begin{cases} Ax = b \geq 0 \\ x \geq 0 \\ (\max) f = cx \end{cases}$$

Putem presupune că problema (P) este în formă bună, altfel spus că matricea A conține o submatrice unitate de ordinul m , unde m este, ca de obicei, numărul restricțiilor. Renumerotând convenabil variabilele problemei putem presupune că primele m coloane din A sunt unitare, astfel că A are forma $A = [E, S]$.

Atunci tabelul simplex asociat unei baze oarecare B, nu neapărat admisibilă, are următorul conținut:

	VVB		⋮
B	$\bar{b}$	B^{-1}	$B^{-1}S$

Fie I mulțimea indicilor vectorilor A^i care formează baza B. Pentru fiecare $i \in I$ notăm:

$$q_i^B = [\bar{b}_i, \bar{a}_{i1}, \bar{a}_{i2}, \dots, \bar{a}_{im}] \quad (4.5.1)$$

unde $\bar{a}_{i1}, \bar{a}_{i2}, \dots, \bar{a}_{im}$ sunt componentele liniei i a matricii B^{-1} .

Definiție Vom spune că baza B este lexicografic admisibilă dacă $q_i^B \succ 0$ pentru orice $i \in I$.

Este clar că dacă B este lexicografic admisibilă atunci B este admisibilă în sensul de până acum adică $\bar{b}_i \geq 0, i \in I$. Reciproc, dacă B este admisibilă și soluția asociată este nedegenerată, adică $\bar{b}_i > 0, i \in I$ atunci evident, B este lexicografic admisibilă.

Baza de start $B^0 = E$ este lexicografic admisibilă. Într-adevăr:

$$q_i^{B^0} = [b_i, 0, \dots, 1, \dots, 0] \quad i = 1, \dots, m$$

și este clar că toți $q_i^{B^0} \succ 0$. Introducem vectorul $q^B = (\bar{f}, \bar{c})$ în care $\bar{f}$ este valoarea funcției obiectiv în soluția asociată bazei B iar $\bar{c}$ este vectorul costurilor reduse corespunzătoare (vezi secțiunea 3.5).

Definiție Spunem că baza B' este lexicografic mai bună decât baza B (în sensul maximizării funcției obiectiv f) și vom scrie $B' \succ B$ dacă $q^{B'} \succ q^B$.

Evident $B' \succ B$ este o relație de ordine totală în mulțimea bazelor problemei (P). Se constată imediat că dacă B' este lexicografic mai bună decât B atunci B' este cel puțin la fel de bună ca și B în sensul obișnuit adică $\bar{f}' \geq \bar{f}$.

Ideea de modificare a algoritmului simplex descris în 4.2 constă în a pleca de la o bază B^0 lexicografic admisibilă și de a genera în continuare numai baze lexicografic admisibile $B^1, B^2, \dots$ care să fie din ce în ce mai bune în sens lexicografic, adică:

$$B^0 \prec B^1 \prec B^2 \prec \dots$$

Având în vedere că în șirul de mai sus nu pot exista două baze identice și că numărul bazelor admisibile este finit urmează că ideea de modificare a algoritmului simplex propusă mai sus garantează convergența procedurii.

Trecem acum la concretizarea ideii de modificare. Fie B o bază lexicografic admisibilă care nu verifică criteriul de optimalitate al algoritmului simplex. Determinăm cu criteriul (4.2.1) coloana A^k care intră în bază și presupunem că A^k are și componente pozitive (cu alte cuvinte nu suntem în cazul optimului infinit). Pentru determinarea indicelui coloanei A^r care părăsește baza curentă B , în locul criteriului (4.2.2) vom folosi formula:

$$\frac{1}{\bar{a}_{rk}} q_r^B = \min_{i, \bar{a}_{ik} > 0} \left\{ \frac{1}{\bar{a}_{ik}} q_i^B \right\} \quad (4.5.2)$$

în care $q_i^B, i \in I$ sunt dați de (4.5.1) iar minimul din membrul drept este luat în sens lexicografic. Acest minim se calculează astfel:

- se determină mulțimea I_0 a indicilor $r \in I$ astfel încât:

$$\frac{\bar{b}_r}{\bar{a}_{rk}} = \min_{i, \bar{a}_{ik} > 0} \left\{ \frac{\bar{b}_i}{\bar{a}_{ik}} \right\}$$

Dacă I_0 se reduce la un singur indice r atunci $\frac{1}{\bar{a}_{rk}} q_r^B$ realizează minimul lexicografic din (4.5.2). Prin urmare dacă minimul din (4.2.2) este unic cele două criterii de ieșire din bază (4.2.2) și (4.5.2) coincid.

Dacă I_0 se compune din cel puțin două elemente, se determină mulțimea $I_1 \subseteq I_0$ a indicilor r cu proprietatea că:

$$\frac{\bar{a}_{r1}}{\bar{a}_{rk}} = \min_{i \in I_0} \frac{\bar{a}_{i1}}{\bar{a}_{ik}}$$

Dacă nici I_1 nu se reduce la un singur element, se determină în continuare sub mulțimea $I_2 \subseteq I_1$ formată din indicii r pentru care:

$$\frac{a_{r2}}{\bar{a}_{rk}} = \min_{i \in I_1} \frac{a_{i2}}{\bar{a}_{ik}}$$

ș.a.m.d.

Se generează astfel un șir de submulțimi:

$$I_0 \supseteq I_1 \supseteq I_2 \supseteq \dots$$

ultima din ele având cu siguranță un singur element ce definește minimul lexicografic din (4.5.2). Într-adevăr, dacă ultima mulțime generată ar conține cel puțin doi indici diferiți r și r' , aceasta va fi I_m și în consecință:

$$\frac{\bar{a}_{ri}}{\bar{a}_{rk}} = \frac{\bar{a}_{r'i}}{\bar{a}_{r'k}} \quad \text{pentru toți } i = 1, \dots, m$$

ceea ce ar implica proporționalitatea a două linii din matricea B^{-1} , fapt imposibil. Prin urmare, minimul din (4.5.2) este unic.

Să arătăm acum că noua bază B' dedusă din B înlocuind A^k cu A^r :

- este lexicografic admisibilă, adică:

$$q_i^{B'} > 0, i \in I, i \neq r \quad \text{și} \quad q_k^{B'} > 0;$$

- este lexicografic mai bună decât B , adică $q^{B'} \succ q^B$.

În baza formulelor de schimbare a bazei (4.1.12) putem scrie relațiile:

$$q_i^{B'} = q_i^B - \frac{\bar{a}_{ik}}{\bar{a}_{rk}} q_r^B \quad i \neq r; \quad q_k^{B'} = \frac{1}{\bar{a}_{rk}} q_r^B$$

Deoarece prin ipoteză $q_r^B > 0$ și $\bar{a}_{rk} > 0$ rezultă $q_k^{B'} > 0$. Fie $i \neq r$. Dacă $\bar{a}_{ik} \leq 0$ atunci în mod clar $q_i^{B'} > 0$. Dacă $\bar{a}_{ik} > 0$ atunci:

$$q_i^{B'} = \bar{a}_{ik} \left(\frac{1}{\bar{a}_{ik}} q_i^B - \frac{1}{\bar{a}_{rk}} q_r^B \right) > 0$$

în virtutea alegerii lui r , conform (4.5.2). În fine: $q^{B'} = q^B - \frac{\bar{c}_k}{\bar{a}_{rk}} q_r^B$ și deoarece

$$\bar{c}_k < 0 \text{ rezultă } q^{B'} \succ q^B.$$

Aplicarea în calculul manual a criteriului de ieșire din bază modificat (4.5.2) este foarte simplă:

Dacă minimul rapoartelor $\frac{\bar{b}_i}{\bar{a}_{ik}}$, $\bar{a}_{ik} > 0$ din criteriul uzual de ieșire din bază nu este unic și se atinge pe o submulțime de indici I_0 se caută minimul rapoartelor $\frac{\bar{a}_{i1}}{\bar{a}_{ik}}$, $i \in I_0$ în care numărătorii sunt luați din prima coloană a inversei bazei curente. Dacă nici acesta nu este unic și se atinge pe o submulțime $I_1 \subseteq I_0$ se caută minimul rapoartelor $\frac{\bar{a}_{i2}}{\bar{a}_{ik}}$, $i \in I_1$ cu numărătorii din a doua coloană a inversei bazei ș.a.m.d. După cum am văzut acest proces se termină într-un număr finit de pași cu găsirea unui unic indice r . Vectorul A^r părăsește baza.

Considerațiile de mai sus vor fi aplicate problemei din ex.4.5.1 (vezi tabelele 4.5.1 - 4.5.3)

La prima iterație intră în bază A^4 . Se observă că:

$$\min \left\{ \frac{\bar{b}_1}{\bar{a}_{14}}, \frac{\bar{b}_2}{\bar{a}_{24}} \right\} = \min \{0, 0\}$$

nu este unic. Fie $I_0 = \{1, 2\}$. Prima coloană a inversei bazei curente este $\bar{A}^1$ și ca urmare calculăm:

$$\min \left\{ \frac{\bar{a}_{11}}{\bar{a}_{14}}, \frac{\bar{a}_{21}}{\bar{a}_{24}} \right\} = \min \left\{ \frac{1}{1/4}, \frac{0}{1/2} \right\} = 0 = \frac{\bar{a}_{21}}{\bar{a}_{24}}$$

În consecință vectorul A^2 părăsește baza ș.a.m.d.

			0	0	0	3/4	-20	1/2	-6
c^B	B	VVB	$\bar{A}^1$	$\bar{A}^2$	$\bar{A}^3$	$\bar{A}^4$	$\bar{A}^5$	$\bar{A}^6$	$\bar{A}^7$
0	A^1	0	1	0	0	1/4	-8	-1	9
0	A^2	0	0	1	0	1/2	-12	-1/2	3
0	A^3	1	0	0	1	0	0	1	0
	f	0	*	*	*	-3/4	20	-1/2	6
0	A^1	0	1	-1/2	0	0	-2	-3/4	15/2
3/4	A^4	0	0	2	0	1	-24	-1	6
0	A^3	1	0	0	1	0	0	1	0
	f	0	*	3/2	*	*	2	-5/4	9/2
0	A^1	3/4	1	-1/2	3/4	0	-2	0	15/2
3/4	A^4	1	0	2	1	1	-24	0	6
1/2	A^6	1	0	0	1	1	0	1	0

	f	5/4	*	3/2	5/4	*	2	*	9/2
--	-----	-----	---	-----	-----	---	---	---	-----

Tabelele 4.5.1 - 4.5.3

4.6 Alte exemple numerice

Aceste exemple au rolul de a ilustra diferitele fapte teoretice prezentate în secțiunile anterioare.

Exemplul 4.6.1 (Optim infinit, raze extreme).

Vom considera programul (P) împreună cu forma sa standard (FSP):

$$(P) \begin{cases} (\max) f = 3x_1 + 4x_2 \\ -3x_1 + 4x_2 \leq 12 \\ -2x_1 + x_2 \leq 2 \\ x_1 - 2x_2 \leq 2 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \Rightarrow (\text{FSP}) \begin{cases} (\max) f = 3x_1 + 4x_2 \\ -3x_1 + 4x_2 + x_3 = 12 \\ -2x_1 + x_2 + x_4 = 2 \\ x_1 - 2x_2 + x_5 = 2 \\ x_j \geq 0, j = 1, \dots, 5 \end{cases}$$

Plecând cu baza unitară admisibilă $E = [A^3, A^4, A^5]$, în două iterații se ajunge la tabelul 4.6.1:

			3	4	0	0	0
C^B	B	VVB	$\bar{A}^1$	$\bar{A}^2$	$\bar{A}^3$	$\bar{A}^4$	$\bar{A}^5$
3	A^1	4/5	1	0	1/5	-4/5	0
4	A^2	18/5	0	1	2/5	-3/5	0
0	A^5	42/5	0	0	3/5	-2/5	1
	f	84/5	*	*	11/5	-24/5	*

Tabelul 4.6.1

Conform pasului 3 al algoritmului simplex programul (P) are optim infinit deoarece:

$$\bar{c}_4 < 0 \text{ iar } \bar{A}^4 \leq 0$$

Vectorul $w^1 = (4/5, 3/5, 0, 1, 2/5)^T$ este o rază extremă a mulțimii $\mathcal{A}_{\text{FSP}}$ a soluțiilor admisibile ale formei standard (FSP), vezi (4.2.4). Prin "proiecție", vectorul $w^1 = (4/5, 3/5)^T$ este o rază extremă a mulțimii $\mathcal{A}_P$ a soluțiilor admisibile ale programului inițial (P). Din fig. 4.6.1 rezultă că $\mathcal{A}_P$ mai are o rază extremă $w^2 = (2, 1)^T$. Această rază se poate determina tot cu ajutorul algoritmului simplex

cu condiția ca la prima iterație, în bază, să fie introdusă coloana A^1 .
Propunem cititorului să efectueze calculele necesare.

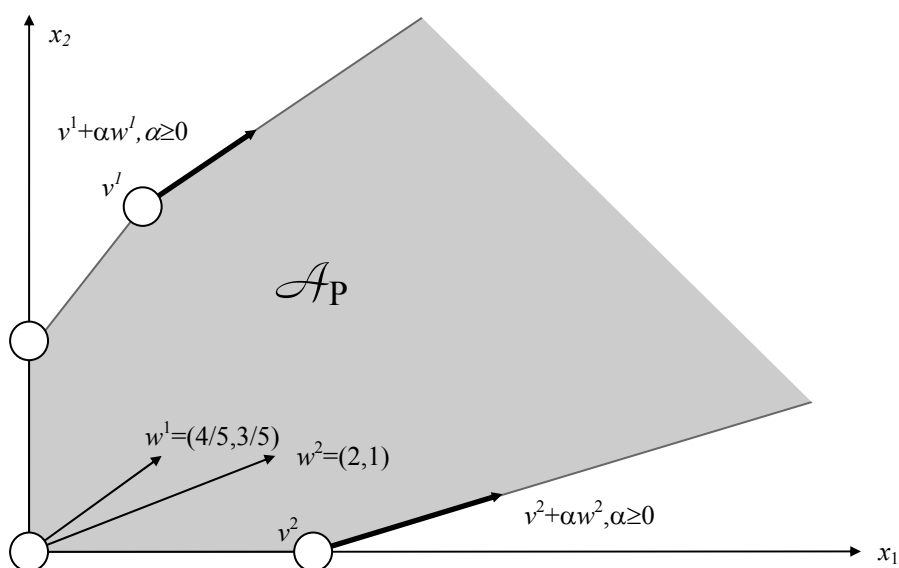


Figura 4.6.1

Exemplu 4.6.2 (Incompatibilitate) Considerăm programul:

$$(P) \begin{cases} (\min) f = 6x_1 + x_2 - 2x_3 \\ 5x_1 - x_2 + x_3 \geq 6 \\ 2x_1 + x_2 + 3x_3 \leq 7 \\ -3x_1 + x_2 + 2x_3 = 2 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

După introducerea variabilelor de abatere x_4, x_5 și a variabilelor artificiale x_6 și x_7 (în prima și respectiv a treia restricție) se obține programul în formă bună:

$$(FBP) \begin{cases} (\min) f = 6x_1 + x_2 - 2x_3 + Mx_6 + Mx_7, \quad M \geq 0 \\ 5x_1 - x_2 + x_3 - x_4 + x_6 = 6 \\ 2x_1 + x_2 + 3x_3 + x_5 = 7 \\ -3x_1 + x_2 + 2x_3 + x_7 = 2 \\ x_j \geq 0 \quad 1 \leq j \leq 7 \end{cases}$$

(Deoarece funcția obiectiv se minimizează, variabilele artificiale au fost incluse în funcția obiectiv cu un coeficient pozitiv M foarte mare!)

Pornind de la baza unitară $E=[A^6, A^5, A^7]$, în două iterații algoritmul simplex produce tabelul 4.6.2, ce conține soluția optimă a programului (FBP).

				6	1	-2	0	0	M
c^B	B	VVB	$\bar{A}^1$	$\bar{A}^2$	$\bar{A}^3$	$\bar{A}^4$	$\bar{A}^5$	$\bar{A}^6$	$\bar{A}^7$
M	A^6	1	0	-1	0	-1	-1	1	1
6	A^1	8/13	1	-1/13	0	0	2/13	0	-3/13
-2	A^3	25/13	0	5/13	1	0	3/13	0	2/13
	f	M - 2/13	*	- M - 29/13	*	- M	-M + 6/13	*	- 22/13

Tabelul 4.6.2

Deoarece în această soluție variabila artificială x_6 are valoare nenulă, programul în forma standard și implicit programul original nu au soluții admisibile.

Exemplul 4.6.3 (Programe liniare cu variabile fără restricție de semn)

Condiția de nenegativitate impusă variabilelor unui program liniar (P) nu este deloc restrictivă. Într-adevăr să presupunem că în (P) ar exista o

variabilă x_j ce poate lua orice valoare reală. Înlocuim x_j cu diferența a două variabile nenegative:

$$x_j = x_j' - x_j'' \quad \text{cu } x_j' \geq 0, \quad x_j'' \geq 0$$

În consecință, în locul coloanei A^j a coeficienților variației x_j vor apare două coloane:

$$(A^j)' = A^j \quad (A^j)'' = -A^j$$

iar în funcția obiectiv coeficientul c_j se înlocuiește cu : $c_j' = c_j$ și $c_j'' = -c_j$

Deoarece $(A^j)' + (A^j)'' = 0$, coloanele $(A^j)'$ și $(A^j)''$ sunt liniar dependente și prin urmare nu vor putea apare simultan în nici o bază a programului transformat. După rezolvarea programului transformat, valoarea variabilei originale x_j în soluția optimă va fi $x_j^* = (x_j')^* - (x_j'')^*$. Cum x_j' și x_j'' nu pot fi simultan variabile bazice vom avea: $x_j^* = (x_j')^* \geq 0$ sau $x_j^* = -(x_j'')^* \leq 0$ după cum, în

soluția optimă a programului transformat x'_j , respectiv x''_j , a fost variabilă bazică. Pentru ilustrare să considerăm problema:

$$(P) \begin{cases} \min f = 3x_1 + 2x_2 \\ -x_1 + 2x_2 \geq 2 \\ -3x_1 + x_2 \leq 3 \\ x_1 \text{ fara restrictii de semn}, x_2 \geq 0 \end{cases}$$

Înlocuim: $x_1 = x'_1 - x''_1$, adăugăm variabilele de abatere x_3, x_4 precum și variabila artificială x_5 în scopul obținerii unei baze unitare de start. Rezultă următorul program transformat:

$$(P') \begin{cases} \min f = 3x'_1 - 3x''_1 + 2x_2 + Mx_5, M \gg 0 \\ -x'_1 + x''_1 + 2x_2 - x_3 + x_5 = 2 \\ -3x'_1 + 3x''_1 + x_2 + x_4 = 3 \\ x'_1, x''_1, x_2, x_3, x_4, x_5 \geq 0 \end{cases}$$

În continuare dăm elementele ultimului tabel simplex:

			3	-3	2	0	0	M
C^B	B	VVB	$(A^1)'$	$(A^1)''$	A^2	A^3	A^4	A^5
2	A^2	3/5	0	0	1	-3/5	-1/5	3/5
-3	$(A^1)''$	4/5	-1	1	0	1/5	2/5	-1/5
	f	-6/5	0	*	*	-9/5	-8/5	9/5 - M

Tabelul 4.6.3

Soluția optimă a programului (P') este deci:

$$(x'_1)^* = 0, (x''_1)^* = 4/5, x_2^* = 3/5, f_{\min} = -6/5$$

Urmează că soluția optimă a programului original va fi :

$$x_1^* = -4/5, x_2^* = 3/5$$

4.7 Interpretarea economică a algoritmului simplex

Să considerăm cazul în care problema de maximizare în formă standard:

$$\begin{cases} \max f = cx \\ Ax = b \\ x \geq 0 \end{cases}$$

reprezintă modelul de optimizare a activității unei firme (secțiunea 1.1, exemplul 1). Se cere determinarea combinației de bunuri ce urmează a fi realizate precum și a cantităților în care acestea vor fi produse astfel încât venitul firmei să fie maxim cu condiția consumării integrale a resurselor disponibile. Ultima cerință nu este deloc restrictivă întrucât în lista activităților productive putem include la nevoie un număr de activități fictive ale căror nivele să reprezinte resursele neconsumate (aceste activități fictive corespund variabilelor de abatere).

O primă concluzie care se desprinde din teoria metodei simplex este aceea că în orice soluție optimă a modelului numărul bunurilor ce vor fi efectiv realizate este cel mult egal cu numărul resurselor folosite.

Aceasta rezultă din faptul că în orice soluție de bază numărul componentelor nenule nu depășește numărul restricțiilor. În consecință, realizarea unor bunuri ce nu sunt incluse în "lista optimă" implică adăugarea unor noi restricții în model, fapt care duce la diminuarea optimului inițial.

Să considerăm acum un program de producție "de bază", adică o soluție admisibilă de bază, asociată unei baze B . Activitățile i corespunzătoare coloanelor A^i din B vor fi numite în continuare *activități de bază*, iar celelalte *activități secundare*. Conform definiției, acest program nu prevede folosirea activităților secundare iar cantitățile de bunuri realizate în activitățile de bază trebuie astfel dimensionate încât să se asigure consumarea întregului stoc de resurse. În notațiile secțiunii 3.5 :

$$Bx^B = b, \quad x^S = 0 \quad \text{de unde } x^B = B^{-1}b = \bar{b}$$

După cum se vede, lista activităților de bază determină în mod unic cantitățile de bunuri ce pot fi produse și ca atare detectarea unui program mai bun se poate face numai studiind oportunitatea utilizării unor activități secundare care

să înlocuiască o parte din activitățile bazice curente. Pentru aceasta avem nevoie de un criteriu care să permită compararea unei activități secundare j cu grupul activităților de bază. Să examinăm coloana $\bar{A}^j$ a tabelului simplex asociat bazei B . Conform (3.5.3) $\bar{A}^j = B^{-1}A^j$ de unde $A^j = B \cdot \bar{A}^j$ sau:

$$A^j = \bar{a}_{1j}A^1 + \bar{a}_{2j}A^2 + \dots + \bar{a}_{mj}A^m \quad (4.7.1)$$

în ipoteza că $B = [A^1, A^2, \dots, A^m]$. Sensul economic al egalității (4.7.1) este următorul: din punctul de vedere al consumului de resurse producerea unei unități din bunul j este echivalentă cu producerea cantităților $\bar{a}_{1j}, \bar{a}_{2j}, \dots, \bar{a}_{mj}$ din bunurile activităților de bază. În consecință, dacă se dorește producerea unei unități din bunul j , producția activităților de bază trebuie diminuată cu cantitățile $\bar{a}_{1j}, \bar{a}_{2j}, \dots, \bar{a}_{mj}$. Analiza oportunității introducerii în fabricație a bunului j se va face prin compararea aportului său la creșterea venitului firmei cu aportul valoric al cantităților $\bar{a}_{1j}, \bar{a}_{2j}, \dots, \bar{a}_{mj}$. Astfel, realizarea unei unități din bunul j determină creșterea valorii curente a funcției obiectiv cu prețul său c_j în timp ce renunțarea la producerea cantităților $\bar{a}_{1j}, \bar{a}_{2j}, \dots, \bar{a}_{mj}$ înseamnă diminuarea aceleiași valori cu suma $c_1\bar{a}_{1j} + c_2\bar{a}_{2j} + \dots + c_m\bar{a}_{mj}$. Prin urmare dacă diferența:

$$\bar{c}_j = c_1\bar{a}_{1j} + c_2\bar{a}_{2j} + \dots + c_m\bar{a}_{mj} - c_j$$

este ≥ 0 realizarea bunului j nu este rentabilă deoarece nu duce la creșterea valorii producției asigurate prin programul curent. Dacă $\bar{c}_j \geq 0$ pentru toate activitățile secundare, programul de fabricație curent este optim. Am regăsit astfel criteriul de optimalitate din teorema A secțiunea 4.1. Dacă $\bar{c}_j < 0$, utilizarea activității secundare j duce la o sporire a venitului realizabil prin programul curent, viteza de creștere fiind $-\bar{c}_j$. Interpretarea criteriului de intrare în bază (4.2.1) este acum clară: dacă mai multe activități secundare sunt rentabile în raport cu grupul activităților de bază este preferată activitatea care asigură cea mai ridicată viteză de creștere a valorii curente a producției. Fie k această activitate. Ultima problemă constă în stabilirea cantității din bunul k ce poate fi realizată în condițiile date. Conform celor de mai sus producerea unei cantități θ din acest bun implică micșorarea producției din bunurile activităților de bază cu cantitățile $\theta\bar{a}_{1k}, \theta\bar{a}_{2k}, \dots, \theta\bar{a}_{mk}$:

$$x_1 = \bar{b}_1 - \theta\bar{a}_{1k}, x_2 = \bar{b}_2 - \theta\bar{a}_{2k}, \dots, x_m = \bar{b}_m - \theta\bar{a}_{mk}$$

Deoarece desfășurarea unei activități la un nivel negativ este lipsită de sens va trebui să avem $\bar{b}_i - \theta \bar{a}_{ik} \geq 0 \quad i = 1, \dots, m$ de unde:

$$\theta \leq \theta_0 = \min_{i, \bar{a}_{ik} > 0} \frac{\bar{b}_i}{\bar{a}_{ik}}$$

(excludem cazul optimului infinit, nesemnificativ din punct de vedere economic). Dacă $\theta_0 = \bar{b}_r / \bar{a}_{rk}$ și $\theta = \theta_0$ obținem creșterea maximă a valorii curente a producției prin utilizarea activității k . În acest caz activitatea k nu mai este folosită întrucât $x_r = \bar{b}_r - \theta_0 \bar{a}_{rk} = \bar{b}_r - \frac{\bar{b}_r}{\bar{a}_{rk}} \bar{a}_{rk} = 0$. Am obținut din nou criteriul de ieșire din bază (4.2.2).

4.8 Versiunea revizuită a algoritmului simplex

Reluăm programul liniar în formă standard (P):

$$(P) \begin{cases} (\max) f = cx \\ x \geq 0 \\ Ax = b \end{cases}$$

cu notațiile și terminologia introduse în secțiunile 3.4 - 3.5. Considerăm o iterație oarecare a algoritmului simplex în care se cercetează soluția asociată unei baze B. Pentru efectuarea iterației sunt necesar următoarele elemente:

$$1) \text{ Costurile reduse: } \bar{c}_j = c^B B^{-1} A^j - c_j, \quad j \in J$$

Dacă testul de optimalitate nu este îndeplinit se determină $k \in J$ astfel ca:

$$\bar{c}_k = \min_{j \in J} \bar{c}_j$$

Coloana A^k intră în baza curentă.

2) Componentele coloanei:

$$\bar{A}^k = B^{-1} A^k$$

3) Valorile variabilelor bazice curente x_i , $i \in I$ reunite în vectorul:

$$\bar{b} = B^{-1}b$$

Admițând că nu are loc cazul optimului infinit se determină $i \in I$ astfel încât:

$$\frac{\bar{b}_r}{a_{rk}} = \min_{a_{ik} > 0} \frac{\bar{b}_i}{a_{ik}}$$

Coloana A^k părăsește baza curentă.

După efectuarea operațiilor necesare evaluării elementelor amintite, tabelul simplex curent se pivotează cu pivotul $\bar{a}_{rk}$.

Se observă că aceste mărimi numerice se pot calcula direct din datele inițiale A, b, c ale programului (P) cunoscând inversa B^{-1} a bazei curente.

Dacă avem în vedere soluționarea problemelor practice de programare liniară caracterizate în primul rând prin numărul mare de restricții și variabile următoarele probleme trebuie să ne dea de gândit:

- Din mulțimea de coloane calculate $\bar{A}^j = B^{-1}A^j, j \in J$ doar una este efectiv folosită la aflarea pivotului și anume $\bar{A}^k$; toate celelalte servesc la calculul costurilor reduse conform formulei:

$$\bar{c}_j = c^B \bar{A}^j - c_j, j \in J$$

Cum în practică problemele au multe variabile, astfel spus matricea A are multe coloane, evaluarea coloanelor $\bar{A}^j$ necesită timp și efort de calcul.

- Exceptând primul tabel simplex, celelalte se deduc unul din altul prin pivotare gaussiană, facilitând propagarea și amplificarea erorilor de rotunjire inerente calculului cu numere reale. Este posibil ca în final aceste erori să

compromită grav rezultatele (de exemplu o problemă compatibilă să fie declarată incompatibilă).

- Chiar dacă matricea inițială A este rară (adică densitatea elementelor nenule este mică), densitatea elementelor nenule în tabelele simplex generate crește vertiginos, cu efecte negative asupra vitezei de calcul.

Este de la sine înțeles că aceste neajunsuri nu sunt vizibile în calculul manual. Aceste probleme apar în contextul rezolvării programelor de dimensiuni mari când utilizarea calculatorului este inevitabilă!

Versiunea revizuită a algoritmului simplex elimină în mare parte aceste neajunsuri. În esență, această versiune propune calcularea mărimilor amintite mai înainte într-o altă ordine și apelarea la datele inițiale ale problemei. La fiecare iterație este necesară cunoașterea inversei bazei curente.

Fie B o bază a programului (P) . Componentele vectorului (linie):

$$\pi = c^B B^{-1}$$

se numesc *multiplicatori simplex asociați bazei B* (vezi observația din finalul secțiunii 4.4)

Inserăm elementele B^{-1} , π , $\bar{b} = B^{-1}b$ și $\bar{f} = c^B \bar{b} = \pi b$ într-un tabel:

B	$\bar{b}$	B^{-1}
f	$\bar{f}$	π

numit *tabel simplex redus*.

Să presupunem cunoscută o soluție admisibilă de bază a programului (P) și tabelul simplex redus al acesteia. Cu aceste pregătiri, conținutul unei iterații în versiunea revizuită este următorul:

Pasul 1. Se calculează costurile reduse:

$$\bar{c}_j = \pi A^j - c_j, j \in J$$

Dacă toți $\bar{c}_j \geq 0$ soluția curentă este optimă. Altminteri:

Pasul 2. Se determină indicele nebazic $k \in J$ cu proprietatea :

$$\bar{c}_k = \min_{j \in J} \bar{c}_j \quad (\bar{c}_k < 0 !)$$

Coloana A^k intră în baza curentă. Se calculează $\bar{A}^k = B^{-1} A^k$

Pasul 3. Dacă $\bar{A}^k \leq 0$ programul dat are optim infinit. Altminteri:

Pasul 4. Se determină indicele bazei $r \in I$ cu proprietatea:

$$\frac{\bar{b}_r}{a_{rk}} = \min_{a_{ik} > 0} \frac{\bar{b}_i}{a_{ik}}$$

Coloana A' părăsește baza curentă.

Pasul 5. *Se pivotează tabelul simplex redus curent cu pivotul $\bar{a}_{rk} > 0$ (se presupune că la tabelul simplex redus curent au fost "atașate" coloana $\bar{A}^k$ și costul redus $\bar{c}_k$)*

Față de algoritmul simplex standard, versiunea revizuită are o serie de avantaje:

- *utilizarea la fiecare iterație a datelor inițiale ale problemei (pentru calcularea costurilor reduse $\bar{c}_j$ și a coloanei $\bar{A}^k$) asigură un control mai bun asupra propagării erorilor de rotunjire ca și o viteză de calcul mai mare dacă se are în vedere faptul că densitatea elementelor nenule în matricea A este de regulă mică (este știut faptul că o operație de înmulțire se efectuează în calculator numai dacă ambii operanzi sunt nenuli)*

- *volumul de calcule este în general mai mic, mai cu seamă în situația în care numărul variabilelor este cu mult mai mare decât cel al restricțiilor. Este evident faptul că aceste avantaje depind de "calitatea" inversei B^{-1} a bazei curente. Într-adevăr, inversele diferitelor baze cercetate de algoritmul revizuit se calculează ca și în algoritmul standard "una din alta" prin pivotare. Am amintit deja "riscurile numerice" care decurg din această situație ce nu poate fi evitată: pe de o parte propagarea și amplificarea erorilor de rotunjire cu impact negativ asupra acurateții evaluării diferitelor mărimi necesare*

efectuării unei iterații și pe de alta creșterea timpului de calcul ca urmare a "îndesirii" elementelor nenule din B^{-1} . În principiu, atenuarea acestor neajunsuri se face prin reinversarea periodică a bazei curente și "stocarea" inversei B^{-1} sub forma unui produs de matrici foarte simple ca structură.

Pentru determinarea unei soluții admisibile de bază de start, în situația în care programul inițial (P) nu este în formă bună, se aplică *metoda celor două faze* (vezi secțiunea 4.3), cu observația că la începutul fazei II, vor trebui recalculați multiplicatorii simplex ce compun vectorul $\pi = c^B B^{-1}$ în raport cu coeficienții funcției obiectiv din (P)!

Următorul exemplu are menirea de a ilustra diferențele de organizare a calculelor introduse algoritmul simplex revizuit și nu de a pune în evidență avantajele de ordin "numeric" în raport cu algoritmul standard; este și

imposibil de făcut acest lucru deoarece în calculul manual se lucrează "exact". Oricum, în rezolvarea manuală a programelor liniare de dimensiuni mici se recomandă versiunea standard.

Exemplu 4.8.1 Se consideră programul:

$$(P) \begin{cases} (\max) f = 3x_1 + 4x_2 + x_3 + 2x_4 + 2x_5 + x_6 \\ 2x_1 - 4x_3 + 2x_4 - 2x_6 \leq 5 \\ 2x_1 - 4x_2 + 4x_4 - 2x_5 + 2x_6 \geq 1 \\ x_2 + x_3 = 1 \\ x_4 + x_5 + x_6 = 1 \\ x_j \geq 0, 1 \leq j \leq 6 \end{cases}$$

Introducem variabilele de abatere x_7, x_8 în primele două restricții obținând forma standard după care introducem variabilele artificiale x_9, x_{10}, x_{11} în restricțiile 2, 3 și 4 în scopul formării bazei unitare de start. Obținem sistemul liniar:

$$\begin{cases} 2x_1 - 4x_3 + 2x_4 - 2x_6 + x_7 = 5 \\ 2x_1 - 4x_2 + 4x_4 - 2x_5 + 2x_6 - x_8 + x_9 = 1 \\ x_2 + x_3 + x_{10} = 1 \\ x_4 + x_5 + x_6 + x_{11} = 1 \\ x_j \geq 0, j = 1, \dots, 11 \end{cases}$$

Faza I. Atașăm sistemului de mai sus funcția obiectiv:

$$w = x_9 + x_{10} + x_{11}$$

pe care o minimizăm. Luăm ca bază de start $B = E = [A^7, A^9, A^{10}, A^{11}]$ căreia îi corespunde soluția: $x_1 = x_2 = x_3 = x_4 = x_5 = x_6 = x_8 = 0$, $x_7 = 5$, $x_9 = 1$, $x_{10} = 1$, $x_{11} = 1$. În raport cu această bază $c^B = [0, 1, 1, 1]$ astfel că $\pi = c^B B^{-1} = [0, 1, 1, 1]$, $\bar{w} = \pi b = 3$. Primul tabel simplex redus arată astfel:

A^7	5	1	0	0	0
A^9	1	0	1	0	0
A^{10}	1	0	0	1	0
A^{11}	1	0	0	0	1
w	3	0	1	1	1

Tabelul 4.8.1

Calculăm costurile reduse $\bar{c}_j = \pi A^j - c_j$. Costurile reduse corespunzătoare diferitelor iterații ale fazei I sunt date în tabelul 4.8.6; de fiecare dată un cost încadrat arată coloana care intră în baza curentă.

$(A^4)^T \rightarrow$	2	4	0	1	$\bar{A}^4$	
A^7	5	1	0	0	0	2
A^9	1	0	1	0	0	4
A^{10}	1	0	0	1	0	0
A^{11}	1	0	0	0	1	1
w	3	0	1	1	1	$5 = \bar{c}_4$

Tabelul 4.8.2

$(A^2)^T \rightarrow$	0	-4	1	0	$\bar{A}^2$	
A^7	9/2	1	-1/2	0	0	2
A^4	1/4	0	1/4	0	0	-1
A^{10}	1	0	0	1	0	1
A^{11}	3/4	0	-1/4	0	1	1
w	7/4	0	-1/4	1	1	$2 = \bar{c}_2$

$(A^3)^T \rightarrow$	-4	0	1	0	$\bar{A}^3$	
A^7	3	1	0	0	-2	-4
A^4	1	0	0	0	1	0
A^{10}	1/4	0	1/4	1	-1	1
A^2	3/4	0	-1/4	0	1	0
w	1/4	0	1/4	1	-1	$1 = \bar{c}_3$

Tabelul 4.8.3

Tabelul 4.8.4

La prima iterație intră în bază A^4 . Calculăm $\bar{A}^4 = B^{-1}A^4$; pentru aceasta (vezi tabelul 4.8.2 care coincide cu 4.8.1) scriem A^4 în linie, deasupra tabelului

c^B	$(A^1)^T \rightarrow$	2	2	0	0	$\bar{A}^1$	
0	A^7	4	1	1	4	-6	4
2	A^4	1	0	0	0	1	0
1	A^3	1/4	0	1/4	1	-1	1/2
4	A^2	3/4	0	-1/4	0	1	-1/2
	w	0					
	f	21/4	0	-3/4	1	5	$-9/2 = \bar{c}_1$

Tabelul 4.8.5

și facem produsele scalare ale lui $(A^4)^T$ cu liniile matricei B^{-1} ; rezultatele le înscrîm în dreapta tabelului 4.8.2. Tot în dreapta jos înscrîm costul redus $\bar{c}_4$. Parcurgînd celelalte etape al algoritmului găsim că A^9 părăsește baza. Se efectuează pivotarea tabelului 4.8.2 cu pivotul încadrat și se obține tabelul (4.8.3). Tabelele 4.8.4 și 4.8.5 aparțin deasemeni fazei I.

Coef. funcției ob. f	3	4	1	2	2	1	0	0			
Coef. funcției ob. w	0	0	0	0	0	0	0	0	1	1	1
	A ¹	A ²	A ³	A ⁴	A ⁵	A ⁶	A ⁷	A ⁸	A ⁹	A ¹⁰	A ¹¹
Matricea A →	2	0	-4	2	0	-2	1	0	0	0	0
Costuri reduse la iterația ↓	2	-4	0	4	-2	2	0	-1	1	0	0
	0	1	1	0	0	0	0	0	0	1	0
	0	0	0	1	1	1	0	0	0	0	1
FAZA I	1	2	-3	1	5	-1	3	*	-1	*	*
	2	-1/2	2	1	*	3/2	1/2	*	1/4	-5/4	*
	3	1/2	*	1	*	-3/2	-1/2	*	-1/4	-3/4	-2
FAZA II	1	-9/2	*	*	*	9/2	5/2	*	3/4		
	2	*	*	9	*	-9	-2	*	-3/2		
	3	*	*	-9	*	*	-13/2	9/4	3/4		
	4	*	*	*	9/2	*	1/4	9/8	-3/8		
	5	*	3	*	3	*	-2	3/2	*		
	6	*	3	*	3	2	*	3/2	*		

Tabelul 4.8.6

La iterația 3 (tabelul 4.8.5) s-a obținut $w = 0$, ceea ce înseamnă obținerea unei soluții admisibile de bază pentru forma standard a programului (P). Nu s-au mai calculat multiplicatorii simplex relativi la funcția obiectiv w și ca urmare nu s-a mai urmărit verificarea testului de optimalitate în raport cu această funcție! În schimb s-a "actualizat" $c^B = [0, 2, 1, 4]$, în raport cu funcția obiectiv originală f și s-au calculat multiplicatorii simplex în raport cu această funcție.

În alte cinci iterații se obține soluția optimă a programului dat:

$$x_1^* = \frac{11}{2}, x_2^* = 0, x_3^* = 1, x_4^* = x_5^* = 0, x_6^* = 1; \quad (\max)f = \frac{37}{2}$$

În continuare se dau tabelele simplex reduse ale fazei a II-a.

$$(A^5)^T \rightarrow \begin{array}{cccccc|c} 0 & -2 & 0 & 1 & \bar{A}^5 \end{array}$$

$$(A^3) \rightarrow \begin{array}{cccccc|c} -4 & 0 & 1 & 0 & (\bar{A}^3) \end{array}$$

A ⁷	2	1	-1	-4	2	4
A ⁴	1	0	0	0	1	1
A ¹	1/2	0	1/2	2	-2	-3
A ²	1	0	0	1	0	0
f	15/2	0	3/2	10	-4	-9 = $\bar{c}_5$

A ⁵	1/2	1/4	-1/4	-1	1/2	-2
A ⁴	1/2	-1/4	1/4	1	1/2	2
A ¹	2	3/4	-1/4	-1	-1/2	-4
A ²	1	0	0	1	0	1
f	12	9/4	-3/4	1	1/2	-9 = $\bar{c}_3$

Tabelul 4.8.7

$(A^8)^T \rightarrow$	0	-1	0	0	$\bar{A}^8$
A^5	1	0	0	0	1
A^3	1/4	-1/8	1/8	1/2	1/4
A^1	3	1/4	1/4	1	1/2
A^2	3/4	1/8	-1/8	1/2	-1/4
f	57/4	9/8	3/8	11/2	11/4
					$-3/8 = \bar{c}_8$

Tabelul 4.8.9

A^6	1	0	0	0	1
A^3	1	0	0	1	0
A^1	11/2	1/2	0	2	1
A^8	12	1	-1	4	4
f	37/2	3/2	0	7	4

Tabelul 4.8.11

În considerațiile de până acum am avut în vedere în exclusivitate numai aspectele teoretice legate de rezolvarea unui program liniar. Rezolvarea efectivă, mai cu seamă a programelor de dimensiuni mari (rezultate, de exemplu, din modelarea unor situații economice reale), de neconceput fără utilizarea unui calculator, pune însă și alte probleme, de natură numerică cum ar fi controlul propagării erorilor de rotunjire (inerente calculului cu numere reale), volumul de memorie necesar stocării diferitelor informații numerice utilizate în procesul rezolvării, timpul de rulare până la obținerea rezultatului final etc.

Reamintim cu acest prilej că, nu de puține ori, o metodă de optimizare bine fundamentată teoretic s-a dovedit a fi total ineficientă în practică din cauza ignorării dificultăților numerice semnalate mai sus.

De regulă creșterea performanțelor numerice ale unei metode teoretice se face prin :

- Reorganizarea procesului de calcul de așa manieră încât să se asigure diminuarea efectelor nefaste ale propagării erorilor de rotunjire ca și o viteză de calcul crescută. În acest fel s-a ajuns, de exemplu, la versiunea revizuită a algoritmului simplex, versiune care stă la baza celor mai multe pachete de programe utilizate în rezolvarea programelor liniare.

Tabelul 4.8.8

$(A^6)^T \rightarrow$	-2	2	0	1	$\bar{A}^6$
A^5	1	0	0	0	1
A^3	1	0	0	1	0
A^1	9/2	1/2	0	2	0
A^2	6	1	-1	4	-2
f	33/2	3/2	0	7	2
					$-2 = \bar{c}_6$

Tabelul 4.8.10

- Exploatarea structurii problemei de rezolvat care, în cazul unor dimensiuni apreciabile are de obicei anumite proprietăți "speciale". Un

exemplu edificator în acest sens îl constituie diferitele *metode de descompunere sau de relaxare* din programarea liniară. În principiu, aceste metode reduc rezolvarea unei probleme de dimensiuni mari, cu structură specială la rezolvarea mai multor probleme mai mici și implicit mai ușor de manevrat.

Referindu-ne la eficacitatea practică a algoritmului simplex, după o jumătate de veac de experimente numerice și perfecționări teoretice, se poate spune că acesta s-a dovedit a fi o procedură foarte robustă fiind capabilă să rezolve programe liniare de dimensiuni impresionante conținând sute de restricții și mii de variabile.

Este de la sine înțeles că aceste performanțe n-ar fi fost posibile fără utilizarea unor calculatoare din ce în ce mai puternice.