A. KRASTIŅŠ

MATEMĀTISKĀ PROGRAMMĒŠANA

1976

A. KRASTIŅŠ

MATEMĀTISKĀ PROGRAMMĒŠANA

Latvijas PSR Augstākās un vidējās speciālās izglītības ministrija atļāvusi lietot par mācību grāmatu Latvijas PSR augstskolu ekonomisko specialitāšu studentiem matemātiskās programmēšanas kursa apgūšanai

IZDEVNIECĪBA «ZVAIGZNE»
RĪGA 1976

Grāmata atbilst augstskolu ekonomisko specialitāšu mācību programmai disciplīnā «Matemātiskā programmēšana».

Pirmā, visplašākā grāmatas daļa uzrakstīta ar tādu metodisku pieeju, lai to varētu izmantot ne vien ekonomisko fakultāšu studenti, bet arī plašs prakses darbinieku loks, kuriem, organizējot un plānojot ražošanu, ir nepieciešams pielietot lineārās programmēšanas un tīklveida plānošanas elementus. Grāmatas otrās un trešās daļas materiāla izprašanai vajadzīgas priekšzināšanas augstskolu ekonomikas specialitāšu augstākās matemātikas kursa apjomā.

Книга соответствует учебной программе курса «Математическое программирование» для экономических специальностей высших учебных заведений.

Первая, самая обширная часть книги написана с таким методическим подходом, чтобы ею могли бы пользоваться не только студенты экономических факультетов, но также и широкий круг работников практики, у которых возникла необходимость применения элементов линейного программирования в организации и планировании производства. Для понимания содержания второй и третьей части книги необходима предварительная подготовка в объеме курса высшей математики для экономических специальностей вузов.

Рис. 27.

К $\frac{20204-183}{М\ 802(11)-76}$ 139—76

IEVADS

0.1. §. Matemātiskās programmēšanas uzdevumi. Vārds *programmēšana* ieviesies matemātiskajā literatūrā pēdējos gadu desmitos, turklāt ar divējādu nozīmi. Vienā nozīmē šo vārdu lieto tā darba apzīmēšanai, kurš jāpilda cilvēkam vai mašīnai, lai sastādītu kāda uzdevuma risināšanai vajadzīgo automātiskās skaitļošanas mašīnas darba programmu, t. i., lai sistematizētu izpildāmo komandu (pavēļu, instrukciju) virkni. Šajā nozīmē vārds «programmēšana» sastopams arī ar apzīmētājiem, piemēram, automātiskā programmēšana vai programmēšana algoritmiskajās valodās. Pilnīgi citā nozīmē vārds «programmēšana» ieviesies savienojumos ar apzīmētāju «matemātiskā» vai arī ar tādiem apzīmētājiem kā «lineārā», «nelineārā», «kvadrātiskā», «izliektā», «separablā», «diskrētā», «stohastiskā» u. c. Ar nosaukumu *matemātiskā programmēšana* apzīmē matemātikas nozari, kura nodarbojas ar dotās n argumentu funkcijas vislielākās vai vismazākās vērtības atrašanu, ja šīs funkcijas argumentiem jāapmierina kāda dota ierobežojumu sistēma. Atkarībā no tā, kādas ir dotās n argumentu funkcijas īpašības, kādi ir ierobežojumi (nosacījumi), kuri jāapmierina dotās funkcijas argumentiem, kā arī atkarībā no uzdevuma risināšanai lietotās metodes izšķir dažādus matemātiskās programmēšanas apakšnozarojumus. Tādi ir, piemēram, lineārā, nelineārā, kvadrātiskā, izliektā, separablā, diskrētā, stohastiskā programmēšana u. c.

Matemātiskās programmēšanas straujā attīstība pēdējos gadu desmitos cieši saistīta ar kibernētikas un elektroniskās skaitļošanas tehnikas attīstību. Visplašāk izpētītā un arī praktiski visvairāk pielietotā starp visām matemātiskās programmēšanas apakšnozarēm ir lineārā programmēšana. Šo matemātikas nozari, kura pilnīgi pamatojas uz lineāro algebru, pēdējā laikā izdotajā literatūrā mēdz apzīmēt arī ar nosaukumu lineārā optimizācija. Dažādās optimizācijas metodes, tātad arī lineārā optimizācija, ir matemātisks palīglīdzeklis tādu lēmumu pieņemšanai, kuru realizēšana atbilstoši izraudzītam kritērijam dod optimālo ekonomisko efektu. Šādu lēmumu pieņemšana ir sarežģīts uzdevums, jo iespējamo lēmumu skaits var būt ļoti liels. Tā, piemēram, ja būtu jāpieņem plāns, kādu secību paredzēt ceļojumā apmeklējamām 8 vietām, ja ceļojums sākas un beidzas vienā no tām, tad iespējami

$8!=1\cdot2\cdot3\cdot4\cdot5\cdot6\cdot7\cdot8=40\,320$ dažādi lēmumi, jo tāds ir iespējamo plāna variantu skaits. Katra atsevišķa plāna efektivitāti var novērtēt pēc dažādiem kritērijiem. Viens no tādiem kritērijiem varētu būt, piemēram, visīsākais ceļa garums kilometros. Ja pēc tāda kritērija novērtētu visus 40 320 dažādos plānus, tad noskaidrotos, ka pastāv viens plāns vai arī neliels alternatīvu plānu skaits ar vismazāko ceļa garumu. Viena no tādu uzdevumu atrisināšanas metodēm ir dinamiskās programmēšanas metode, kuru aplūkosim šīs grāmatas noslēgumā.

0.2.§. Īsas vēsturiskas ziņas. Pirmā publikācija par lineāro programmēšanu mūsdienu izpratnē ir L. Kantoroviča 1939. gadā publicētā grāmata par matemātiskajām metodēm ražošanas organizācijā un plānošanā.* Šajā grāmatā tās autors devis aplūkoto uzdevumu atrisināšanas algoritmu un atzīmējis, ka šādas metodes pirmām kārtām ir noderīgas plānsaimniecības apstākļos. Diemžēl šis L. Kantoroviča darbs ilgu laiku neradīja vajadzīgo atsaucību, kam par iemeslu lielā mērā bija tā laika skaitļošanas tehnikas zemais attīstības līmenis.

Kad otrā pasaules kara laika apstākļos saimniecība spieda arī Rietumvalstis pievērsties zināmā līmenī centralizētai saimniecības vadībai, sākās ar šādiem uzdevumiem saistīti pētījumi (ASV). 1941. gadā L. Hičkoks un no viņa neatkarīgi 1945. gadā T. Koopmans atrisināja transporta uzdevumu. Vispārīgais lineārās programmēšanas uzdevumu atrisināšanas algoritms izveidojās 1947. gadā G. Danciga zinātniskās darbības rezultātā, bet viņa proponētā simpleksa metode kļuva plaši pazīstama tikai 1951. gadā.

Vienlaicīgi ar lineārās programmēšanas uzdevumu risināšanu attīstījās pētījumi arī nelineārajā programmēšanā. 1951. gadā H. Kūns un A. Takers publicēja savus pētījumus par nepieciešamajiem un pietiekamajiem nelineāru uzdevumu optimālā atrisinājuma eksistences nosacījumiem.

Dinamiskās programmēšanas vēsture cieši saistīta ar R. Bellmana vārdu un viņa 1960. gadā publicēto darbu par dinamiskās programmēšanas metodi.

Starp literatūras avotiem, kas izmantoti šīs grāmatas sarakstīšanā, pirmās 4 vietas ieņem šādi darbi:

1. Зуховицкий С. И., Авдеева Л. И. Линейное и выпуклое программирование. М., 1967. 460 с.

2. Хедли Дж. Нелинейное и динамическое программирование. М., 1967. 506 с.

3. Вагнер Г. Основы исследования операций. В 3-х т. М., 1972–1973.

4. Kreko B. Lehrbuch der linearen Optimierung. Berlin, 1969. 410 S.

---

* Канторович Л. В. Математические методы организации и планирования производства. Л., 1939.

I DAĻA

LINEĀRĀS PROGRAMMĒŠANAS UN TĪKLVEIDA PLĀNOŠANAS ELEMENTI

1. nodaļa

DAŽU LINEĀRĀS PROGRAMMĒŠANAS PIEMĒRU EKONOMISKĀ INTERPRETĀCIJA UN MATEMĀTISKAIS FORMULĒJUMS

1.1. §. Transporta uzdevums. Ar nosaukumu «transporta uzdevums» pazīstams viens no vēsturiski pirmajiem un relatīvi visvienkāršākajiem matemātiskās programmēšanas uzdevumiem.

Aplūkosim skaitlisku piemēru, pieņemot, ka ir dota šāda sākotnējā informācija:

	B_1	B_2	B_3	B_4	B_5	Krājumi
A_1	8	7	6	9	5	40
A_2	4	10	8	3	6	54
A_3	2	3	6	5	3	36
A_4	5	4	8	9	7	50
Pieprasījumi	32	48	62	11	27	180

(1.1.1)

Tabulā (1.1.1) ar A_i $(i=1, 2, 3, 4)$ apzīmētas noliktavas, bet ar B_j $(j=1, 2, 3, 4, 5)$ — patēriņa vietas. Noliktavās A_1, A_2, A_3 un A_4 atrodas kravas vienību krājumi atbilstoši $a_1=40$, $a_2=54$, $a_3=36$ un $a_4=50$, kopā 180 vienību. Šīs kravas vienības jānogādā patēriņa vietās B_1, B_2, B_3, B_4 un B_5 atbilstoši to pieprasījumiem $b_1=$ $=32$, $b_2=48$, $b_3=62$, $b_4=11$ un $b_5=27$, kopā 180 kravas vienību.

Pārējie sākotnējās informācijas tabulā (1.1.1) dotie 20 skaitļi, kas atbilstoši noliktavu un patēriņa vietu skaitam sakārtoti 4 rindās un 5 kolonnās, veido taisnstūrveida tabulu

$$\boldsymbol{C}=\begin{bmatrix} 8 & 7 & 6 & 9 & 5 \\ 4 & 10 & 8 & 3 & 6 \\ 2 & 3 & 6 & 5 & 3 \\ 5 & 4 & 8 & 9 & 7 \end{bmatrix} \qquad (1.1.2)$$

Apzīmējot tabulu (1.1.2) ar vienu burtu $\boldsymbol{C}$, sauksim to par dotā transporta uzdevuma tarifu *matricu.* Katrs šīs tabulas skaitlis *(matricas $\boldsymbol{C}$ elements)* c_{ij} ir noteiktās mērvienībās izteikts tarifs jeb maksa par vienas kravas vienības pārvešanu no i-tās noliktavas uz j-to patēriņa vietu. Tā, piemēram, par vienas kravas vienības pārvešanu pa maršrutu (1, 1), t. i., par kravas vienības pārvešanu no noliktavas A_1 uz patēriņa vietu B_1 jāmaksā $c_{11}=8$ rubļi, ja tarifu matricas elementu mērvienība ir rublis; maksa par vienas kravas vienības pārvešanu pa maršrutu (1, 2) ir $c_{12}=7$ utt.

Uzdevums 1.1.1°. Ievērojot informāciju, kas dota ar tabulu (1.1.1), sastādīt optimālu transporta plānu. Par optimālu uzskatāms plāns, kam atbilst vismazākā transporta izmaksu kopsumma.

Uzdevuma matemātiskais modelis. Apzīmēsim ar x_{ij} to nezināmo kravas vienību skaitu, kas jāpārved pa maršrutu (i, j), t. i., to kravas vienību skaitu, kas no i-tās noliktavas jāpārved uz j-to patēriņa vietu. Tādu nezināmo šajā uzdevumā ir $4\times5=20$. Sakārtojot nezināmos tabulā pēc maršrutiem, rodas nezināmo matrica

$$\boldsymbol{X}=\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} \\ x_{31} & x_{32} & x_{33} & x_{34} & x_{35} \\ x_{41} & x_{42} & x_{43} & x_{44} & x_{45} \end{bmatrix} \qquad (1.1.3)$$

Tā kā par katras kravas vienības pārvešanu pa maršrutu (i, j) jāmaksā c_{ij} rubļi, tad par x_{ij} kravas vienību pārvešanu jāmaksā $c_{ij}x_{ij}$ rubļi. Tā, piemēram, par x_{22} vienību pārvešanu jāmaksā $10x_{22}$ rubļi, bet par x_{25} vienību pārvešanu — $6x_{25}$ rubļi. Ja sareizina atbilstošos matricu $\boldsymbol{C}$ un $\boldsymbol{X}$ elementus un visus tā dabūtos reizinājumus saskaita, tad iegūst transporta izmaksu kopsummu, kas ir šī uzdevuma *mērķa funkcija*

$$\begin{aligned} z=8x_{11}&+7x_{12}+6x_{13}+9x_{14}+5x_{15}+\\ +4x_{21}&+10x_{22}+8x_{23}+3x_{24}+6x_{25}+\\ +2x_{31}&+3x_{32}+6x_{33}+5x_{34}+3x_{35}+\\ +5x_{41}&+4x_{42}+8x_{43}+9x_{44}+7x_{45}. \end{aligned} \qquad (1.1.4)$$

Funkcijas (1.1.4) argumenti x_{ij} $(i=1, 2, 3, 4;\ j=1, 2, 3, 4, 5)$ var pieņemt tikai tādas nenegatīvas vērtības, kuras apmierina šādus nosacījumus:

$$\left.\begin{aligned} &x_{11}+x_{12}+x_{13}+x_{14}+x_{15}=40,\\ &x_{21}+x_{22}+x_{23}+x_{24}+x_{25}=54,\\ &x_{31}+x_{32}+x_{33}+x_{34}+x_{35}=36,\\ &x_{41}+x_{42}+x_{43}+x_{44}+x_{45}=50,\\ &x_{11}+x_{21}+x_{31}+x_{41}=32,\\ &x_{12}+x_{22}+x_{32}+x_{42}=48,\\ &x_{13}+x_{23}+x_{33}+x_{43}=62,\\ &x_{14}+x_{24}+x_{34}+x_{44}=11,\\ &x_{15}+x_{25}+x_{35}+x_{45}=27. \end{aligned}\right\} \qquad (1.1.5)$$

Nosacījumi (1.1.5) veido lineāru algebrisku vienādojumu sistēmu ar $4+5=9$ vienādojumiem un $4\cdot 5=20$ nezināmajiem. Pirmie 4 vienādojumi šajā sistēmā nodrošina visu noliktavās esošo kravas vienību aizvešanu no noliktavām, bet pēdējie 5 vienādojumi nodrošina to, ka patēriņa vietās nonāks tieši tur pieprasītais kravas vienību daudzums.

Uzdevuma 1.1.1° matemātisko modeli tagad varam formulēt šādi.

Dota lineāra mērķa funkcija (1.1.4) un lineāru algebrisku vienādojumu (ierobežojumu) sistēma (1.1.5), kā arī nezināmo nenegativitātes nosacījumi

$$x_{ij} \geqslant 0 \quad (i=1,2,3,4;\ j=1,2,3,4,5). \qquad (1.1.6)$$

Atrast minimumu* funkcijai (1.1.4) pie ierobežojumiem (1.1.5), (1.1.6).

Formulējot lineārās programmēšanas uzdevumus vispārīgā veidā, skaitļu un to reizinājumu summu saīsinātai apzīmēšanai bieži lieto summas simbolu. Tā, piemēram, skaitļu $a_1, a_2, \ldots, a_m$ summu $a_1+a_2+\ldots+a_m$ saīsināti pieraksta ar simbolu

$$\sum_{i=1}^{m} a_i.$$

Lieto arī divkāršu un vairākkārtīgu summu simbolus, piemēram,

$$\begin{aligned}\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} = c_{11}+c_{12}+\ldots+c_{1n}+ \\ +c_{21}+c_{22}+\ldots+c_{2n}+ \\ \ldots\ldots\ldots\ldots\ldots \\ +c_{m1}+c_{m2}+\ldots+c_{mn}.\end{aligned}$$

Vienādojumu $a_{i1}x_1+a_{i2}x_2+\ldots+a_{in}x_n=b_i$ saīsināti var pierakstīt šādi:

$$\sum_{j=1}^{n} a_{ij}x_j = b_i.$$

Koeficients pie katra nezināmā x_j apzīmēts ar a_{ij}, kur indekss i ir vienādojuma, bet j — nezināmā numurs.

---

* Termins *minimums (maksimums)* šajā grāmatā lietots termina *vismazākā (vislielākā) vērtība* vietā visur tur, kur nav vajadzības atšķirt jēdzienu *lokāls minimums (maksimums)* no jēdziena *globāls minimums (maksimums)*, resp., *vismazākā (vislielākā) vērtība*. Šie jēdzieni precizēti grāmatas III daļā.

Klasiskajā transporta uzdevumā vispārīgā gadījumā figurē m noliktavas un n patēriņa vietas. Tarifu matricai

$$C=\begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \dots & \dots & \dots & \dots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix}$$

vispārīgā gadījumā ir $m \cdot n$ elementi. Tikpat daudz elementu ir arī nezināmo lielumu matricai

$$X=\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix}.$$

Apzīmēsim m noliktavās $A_1, A_2, \dots, A_m$ esošos kravas vienību daudzumus atbilstoši ar $a_1, a_2, \dots, a_m$ un patēriņa vietu $B_1, B_2, \dots$ $\dots, B_n$ pieprasītos kravas vienību daudzumus atbilstoši ar $b_1, b_2, \dots, b_n$. Tad klasiskā transporta uzdevuma matemātisko modeli vispārīgā veidā var formulēt šādi:

U z d e v u m s 1.1.2°. Dota lineāra funkcija

$$z=\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{1.1.7}$$

un lineāri algebriski vienādojumi (ierobežojumi)

$$\sum_{j=1}^{n} x_{ij}=a_i \quad (i=1, 2, \dots, m), \tag{1.1.8}$$

$$\sum_{i=1}^{m} x_{ij}=b_j \quad (j=1, 2, \dots, n), \tag{1.1.9}$$

kur

$$\sum_{i=1}^{m} a_i=\sum_{j=1}^{n} b_j \tag{1.1.10}$$

un

$$x_{ij}\geqslant 0 \quad (i=1,\ 2,\ \dots,\ m;\ j=1,\ 2,\ \dots,\ n). \tag{1.1.11}$$

Atrast minimumu lineārajai funkcijai (1.1.7) pie ierobežojumiem (1.1.8)—(1.1.11).

Ja kravas vienību daudzums noliktavās pārsniedz patēriņa vietās pieprasītos daudzumus, tad vienādības (1.1.10) vietā pastāv nevienādība

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

Tādā gadījumā uzdevumā paredz papildu patēriņa vietu ar pieprasīto kravas vienību daudzumu

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

Visi papildu patēriņa vietai atbilstošie tarifu matricas elementi ir nulles. Tā arī šajā gadījumā ir izmantojams iepriekš formulētais klasiskā transporta uzdevuma matemātiskais modelis.

Pieņemsim, piemēram, ka sākotnējā informācija kādam transporta uzdevumam ir dota ar šādu tabulu:

	B_1	B_2	B_3	B_4	Piedāvājumi
A_1	6	7	5	5	100
A_2	10	2	5	6	150
A_3	8	10	20	4	50
Pieprasījumi	75	80	60	50	265 \ 300

(1.1.12)

Pēc tabulā (1.1.12) esošās informācijas piedāvājumi pārsniedz pieprasījumus $\sum_{j=1}^{4} b_j=265$ par

$$b_5=\sum_{i=1}^{3} a_i-\sum_{j=1}^{4} b_j=300-265=35$$

vienībām. Ievedot fiktīvu patērētāju B_5 ar tā pieprasījumu $b_5=35$, tabulu (1.1.12) var atvietot ar tabulu

	B_1	B_2	B_3	B_4	B_5	Piedāvājumi
A_1	6	7	5	5	0	100
A_2	10	2	5	6	0	150
A_3	8	10	20	4	0	50
Pieprasījumi	75	80	60	50	35	300

(1.1.13)

Turpmāk, aplūkojot transporta uzdevuma atrisināšanas metodes, vienmēr iepriekš nodrošināsim vienādību (1.1.10), ja dotajā sākuma informācijā tā nebūtu spēkā.

Šādu uzdevumu atrisināšanas metodes aplūkosim 6. nodaļā.

1.2. §. Ierobežotu ražošanas resursu un ražošanas tehnoloģisko procesu izmantošanas optimizācija. Pieņemsim, ka divu produkcijas veidu B_1 un B_2 ražošanu ierobežo tikai četri ražošanas resursi A_1, A_2, A_3 un A_4, kas mērāmi dažādās mērvienībās. Konkrētā informācija dota ar tabulu

Ražošanas resursi	Mērvienība	Daudzums	Patēriņa normas	
			B_1	B_2
A_1	h(stunda)	84	1	7
A_2	kg	69	2	5
A_3	kW	105	5	5
A_4	gab.	72	4	0
Gatavās produkcijas vienības realizācijas cena (rbļ.)			30	40

(1.2.1)

Tabulā (1.2.1) dotās patēriņa normas ir skaitļi, kas rāda, cik attiecīgā resursa vienību jāizmanto, lai ražotu vienu produkcijas vienību.

Uzdevums 1.2.1°. Pamatojoties uz tabulā (1.2.1) doto informāciju, sastādīt tādu ražošanas plānu, lai, realizējot gatavo produkciju, iegūtu maksimālo ienākumu.

Uzdevuma matemātiskais modelis. Apzīmēsim pagaidām nezināmo produkcijas B_1 vienību skaitu ar x_1, bet produkcijas B_2 vienību skaitu ar x_2. Saskaņā ar ievesto apzīmējumu jēgu x_1 un x_2 nevar būt negatīvi. Gatavās produkcijas vērtība ir šo nezināmo lielumu lineāra funkcija

$$z=30x_1+40x_2, \tag{1.2.2}$$

kuru sauksim par šī uzdevuma mērķa funkciju. Abu produkcijas veidu ražošanai vajadzīgais resursa A_1 vienību skaits x_1+7x_2 nevar pārsniegt 84 h. Analogi resursa A_2 vienību skaits $2x_1+5x_2$ nevar pārsniegt 69 kg, A_3 vienību skaits $5x_1+5x_2$ nevar pārsniegt 105 kW un resursa A_4 vienību skaits $4x_1$, kas nepieciešams produkcijas B_1 ražošanai, nevar pārsniegt 72 gabalus.

Pierakstīsim visus ierobežojumus nevienādību formā:

$$\left.\begin{aligned} x_1+7x_2&\leqslant 84,\\ 2x_1+5x_2&\leqslant 69,\\ 5x_1+5x_2&\leqslant 105,\\ 4x_1\qquad&\leqslant 72, \end{aligned}\right\} \tag{1.2.3}$$

$$x_1\geqslant 0;\quad x_2\geqslant 0. \tag{1.2.4}$$

Uzdevuma 1.2.1° matemātisko modeli tagad varam formulēt šādi.

Atrast maksimumu lineārajai funkcijai (1.2.2) pie ierobežojumiem (1.2.3), (1.2.4).

Vispārīgā gadījumā sākotnējās informācijas tabulā var būt m resursi un n dažādi produkcijas veidi. Bez tam katra produkcijas veida ražošanai var būt k dažādas tehnoloģijas. Ierobežotu ražošanas resursu izmantošanas optimizācijas (optimāla ražošanas plāna sastādīšanas) uzdevumu vispārīgā gadījumā var formulēt šādi.

U z d e v u m s 1.2.2°. Maksimizēt lineāru funkciju

$$z=\sum_{j=1}^{n} c_j x_j \tag{1.2.5}$$

pie ierobežojumiem

$$\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i \quad (i=1, 2, \ldots, m), \tag{1.2.6}$$

$$x_j \geqslant 0 \quad (j=1, 2, \ldots, n). \tag{1.2.7}$$

Šādu uzdevumu atrisināšanas metodes aplūkosim 4. nodaļā.

1.3. §. Barības devu optimizācija. Dažādām uzturvielām, kā zināms, ir dažāds barības komponentu (ingredientu) daudzums. Pieņemsim, ka dzīvnieka dienas barības devā jānodrošina m ingredienti, kuri atrodas n dažādās uzturvielās.

Kā piemērs tabulā (1.3.1) dota infomācija par vienības cenu un ingredientu daudzumu katrā uzturvielas vienībā, kā arī minimālo ingredientu daudzumu, kuriem jābūt dzīvnieka dienas barības devā.

(1.3.1)

Ingredients	Mērvienība	Minimālais daudzums dienas barības devā	Uzturvielas vienībā esošais ingredienta daudzums	
			1. uzturviela (rauši)	2. uzturviela (milti)
A_1 (piemēram, barības vienības)	kg	2,5	1,00	1,25
A_2 (piemēram, sagremojamais proteīns)	g	240	400	80
Uzturvielas vienības cena (kap.)			5	4
Uzturvielas vienību skaits dienas barības devā			x_1	x_2

U z d e v u m s 1.3.1°. Ņemot vērā tabulā (1.3.1) doto informāciju, sastādīt ekonomiski optimālo dzīvnieka dienas barības devu.

Uzdevuma matemātiskais modelis. Minimizēt lineāru funkciju (dienas barības devas izmaksu)

$$z=5x_1+4x_2 \tag{1.3.2}$$

pie ierobežojumiem

$$\begin{aligned} x_1 + 1{,}25x_2 &\geqslant 2{,}5, \\ 400x_1 + 80x_2 &\geqslant 240, \end{aligned} \tag{1.3.3}$$

$$x_1 \geqslant 0, \quad x_2 \geqslant 0. \tag{1.3.4}$$

Ierobežojumu (1.3.4) jēga ir acīm redzama, jo uzturvielas vienību skaits dienas barības devā nevar būt negatīvs. Ierobežojumu sistēmas (1.3.3) pirmā nevienādība nodrošina, ka ingredients A_1 (barības vienības) dienas barības devā būs vismaz 2,5 kg. Sistēmas (1.3.3) otrā nevienādība savukārt nodrošina, ka ingredients A_2 (sagremojamais proteīns) dienas barības devā būs vismaz 240 g.

Barības devas sastādīšanas uzdevumu vispārīgā veidā var formulēt šādi.

Uzdevums 1.3.2°. Minimizēt lineāru funkciju

$$z \Rightarrow \sum_{j=1}^{n} c_j x_j \tag{1.3.5}$$

pie ierobežojumiem

$$\sum_{j=1}^{n} a_{ij} x_j \geqslant b_i \quad (i=1, 2, \ldots, m), \tag{1.3.6}$$

$$x_j \geqslant 0 \quad (j=1, 2, \ldots, n). \tag{1.3.7}$$

1.4. §. Ražošanas jaudu izmantošanas optimizācija. Pieņemsim, ka laika posmā, kura ilgums nepārsniedz 5000 minūtes, jāsaražo 3 dažādi produkti:

A — ne mazāk kā 500 vienības
B — „ „ „ 300 „
C — tieši 450 vienības

Šāda uzdevuma veikšanai var izmantot jebkuru no divām dažādām tehnoloģijām. Tehnoloģiju jaudu raksturo tabula (1.4.1):

Tehnoloģija	Produkts		
	A	B	C
I	4	10	10
II	6	8	20

(1.4.1)

Tabulā (1.4.1) uzrādītie skaitļi ir minūtēs izteikts laiks, kāds nepieciešams vienas produkta vienības ražošanai pēc dotās tehnoloģijas.

U z d e v u m s 1.4.1°. Sastādīt tādu ražošanas jaudu izmantošanas plānu, lai minimālā laika posmā veiktu doto ražošanas uzdevumu.

Uzdevuma matemātiskais modelis. Apzīmēsim ar x_{11}, x_{12} un x_{13} attiecīgi produkta A, produkta B un produkta C vienību daudzumu, kas ražojams pēc I tehnoloģijas. Atbilstošos nezināmos II tehnoloģijai apzīmēsim ar x_{21}, x_{22} un x_{23}. Tad ražošanas uzdevuma veikšanai nepieciešamo koplaiku izteic lineāra funkcija

$$z=4x_{11}+10x_{12}+10x_{13}+6x_{21}+8x_{22}+20x_{23}. \tag{1.4.2}$$

Lai izpildītu uzdevumā dotos ražošanas nosacījumus, nepieciešami šādi ierobežojumi:

$$\left.\begin{aligned} 4x_{11}+10x_{12}+10x_{13}&\leqslant 5000,\\ 6x_{21}+\ 8x_{22}+20x_{23}&\leqslant 5000,\\ x_{11}+\ x_{21}&\geqslant 500,\\ x_{12}+\ x_{22}&\geqslant 300,\\ x_{13}+\ x_{23}&= 450, \end{aligned}\right\} \tag{1.4.3}$$

$$x_{ij}\geqslant 0 \quad (i=1,\ 2;\ j=1,\ 2,\ 3). \tag{1.4.4}$$

Ierobežojumi (1.4.4) ir acīm redzami, jo produkcijas vienību skaits nevar būt negatīvs. Ierobežojumu sistēmas (1.4.3) pirmās divas nevienādības nodrošina to, ka nevienu tehnoloģiju nenoslogos vairāk par 5000 minūtēm. Pēdējās trīs nevienādības šajā sistēmā nodrošina to, ka ražos uzdevumā pieprasīto produktu daudzumu.

Uzdevuma matemātisko modeli var formulēt šādi: *minimizēt* lineāro funkciju (1.4.2) pie ierobežojumiem (1.4.3)—(1.4.4).

2. nodaļa

LINEĀRĀS PROGRAMMĒŠANAS VISPĀRĪGIE JĒDZIENI

2.1. §. Lineārās programmēšanas vispārīgais uzdevums un tā modifikācijas. Jebkuru lineārās programmēšanas uzdevumu vispārīgā veidā var formulēt šādi:

U z d e v u m s 2.1.1°. *Dota lineāra mērķa funkcija*

$$z=\sum_{j=1}^{n} c_j x_j \tag{2.1.1}$$

un lineāras nevienādības vai vienādojumi (ierobežojumi)

$$\sum_{j=1}^{n} a_{ij}x_j\leqslant b_i \quad (i=1, 2, \ldots, p<h), \tag{2.1.2}$$

$$\sum_{j=1}^{n} a_{ij}x_j \geqslant b_i \quad (i=p+1, p+2, \ldots, h<m), \tag{2.1.3}$$

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \quad (i=h+1, h+2, \ldots, m), \tag{2.1.4}$$

$$x_j \geqslant 0 \quad (j=1, 2, \ldots, n), \tag{2.1.5}$$

kur a_{ij}, $b_i \geqslant 0$ *un* c_j *ir doti reāli skaitļi, bet* x_j *— nezināmi reāli lielumi.*

Atrast maksimumu lineārajai funkcijai (2.1.1) pie ierobežojumiem (2.1.2)—(2.1.5).

Uzdevuma formulējumā nezināmo x_j nenegativitātes ierobežojumi (2.1.5) ir ierobežojumu (2.1.3) apakšgadījums. Praktiskos lineārās programmēšanas modeļos nezināmo nenegativitāte ļoti bieži izriet no šo nezināmo lielumu ekonomiskās vai fizikālās jēgas. Tādēļ ir lietderīgi nezināmo nenegativitātes nosacījumus neietilpināt ierobežojumos (2.1.3), bet dot kā īpašu ierobežojumu grupu (2.1.5). Tas, kā vēlāk redzēsim, ļauj arī vienveidot un vienkāršot uzdevuma risināšanas gaitu.

Ja lineārās programmēšanas uzdevuma formulējumā starp dotajiem ierobežojumiem nav (2.1.2) un (2.1.3) tipa izteiksmju, tad saka, ka uzdevums ir *kanoniskajā formā.* Tādā formā doti, piemēram, uzdevumi 1.1.1° un 1.1.2°.

Ja uzdevuma formulējumā nav (2.1.3) un (2.1.4) tipa izteiksmju, tad saka, ka lineārās programmēšanas uzdevums ir *normālformā.* Tādā formā doti, piemēram, uzdevumi 1.2.1° un 1.2.2°.

Ja uzdevuma formulējumā nav (2.1.3) tipa izteiksmju, tad saka, ka uzdevums ir *modificētajā normālformā.*

Lineārās programmēšanas uzdevumu kanoniskā forma un normālforma acīm redzami ir modificētās normālformas apakšgadījumi.

2.2. §. Lineārās programmēšanas uzdevumu algebriskās transformācijas. Šajā paragrāfā, sadalot to piecos apakšnodalījumos, aplūkosim dažus vienkāršus paņēmienus, kuri ļauj vārdiem un matemātiskiem simboliem formulētu lineārās programmēšanas uzdevuma modeli pārveidot citā, ar citiem vārdiem un citiem simboliem izteiktā, bet dotajam uzdevumam ekvivalentā lineārās programmēšanas uzdevuma modelī.

1°. Jebkuru lineārās programmēšanas uzdevumu, kurā jāatrod lineārās funkcijas minimums, var atvietot ar tam ekvivalentu maksimizācijas uzdevumu (un otrādi), nemainot nevienu no ierobežojumiem. Lineārās funkcijas $z=\sum_{j=1}^{n} c_jx_j$ minimizācija ir ekvivalenta ar funkcijas $\bar{z}=\sum_{j=1}^{n} (-c_j)x_j$ maksimizāciju. Ja $\max \bar{z}=z^*$, tad $\min z = -z^*$, un otrādi: $\max z = -\min(\bar{z})$. Ja, piemēram, jāatrod lineārās funkcijas $z = 2x_1 - 3x_2 + 5x_3$ minimums, tad šī uzdevuma vietā, nemainot nevienu ierobežojumu, var meklēt maksimumu lineārajai funkcijai $\bar{z} = -2x_1 + 3x_2 - 5x_3$. Atrodot, ka $\max \bar{z}$ pie dotajiem ierobežojumiem ir, piemēram, -230, secinām, ka $\min z = 230$.

2°. Tā kā, reizinot kādas nevienādības abas puses ar -1, mainās nevienādības zīme, tad nevienādība

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant -b_i$$

ir ekvivalenta ar nevienādību

$$\sum_{j=1}^{n} (-a_{ij}) x_j \geqslant b_i.$$

Tā, piemēram, nevienādība $3x_1 - 2x_2 \leqslant -3$ ir ekvivalenta ar nevienādību $-3x_1 + 2x_2 \geqslant 3$.

3°. Jebkuru vispārīgā veidā dotu lineārās programmēšanas uzdevumu var pārveidot modificētajā normālformā vai kanoniskajā formā, ja ierobežojumu sistēmā ieved nenegatīvus papildu nezināmos. Nevienādību (2.1.2) vietā ierobežojumu sistēmā varam rakstīt vienādojumus

$$y_i + \sum_{j=1}^{n} a_{ij} x_j = b_i \quad (i = 1, 2, \ldots, p),$$

kur $y_i \geqslant 0$ $(i = 1, 2, \ldots, p)$ ir *papildu nezināmie*. Papildu nezināmajiem katrā konkrētā uzdevumā var būt sava ekonomiskā interpretācija. Tā, piemēram, ja nevienādību sistēmas (1.2.3) vietā uzdevumā 1.2.1° raksta vienādojumus

$$\left.\begin{aligned} y_1 + x_1 + 7x_2 &= 84, \\ y_2 + 2x_1 + 5x_2 &= 69, \\ y_3 + 5x_1 + 5x_2 &= 105, \\ y_4 + 4x_1 \qquad &= 72, \end{aligned}\right\} \qquad (2.2.1)$$

tad y_1, y_2, y_3 un y_4 ir ražošanā neizmantoto resursu atlikumi.

Nevienādību (2.1.3) vietā ierobežojumu sistēmā varam rakstīt vienādojumus

$$-y_i + \sum_{j=1}^{n} a_{ij} x_j = b_i \quad (i = p+1,\ p+2, \ldots, h),$$

kur $y_i \geqslant 0$ $(i = p+1,\ p+2, \ldots, h)$ ir papildu nezināmie. Tā, piemēram, nevienādību sistēmas (1.3.3) vietā uzdevumā 1.3.1° varam rakstīt vienādojumus

$$\left.\begin{aligned} -y_1+\quad x_1+1{,}25x_2&=2{,}5,\\ -y_2+200x_1+\ 80x_2&=240.\end{aligned}\right\}$$

Nezināmie $y_1\geqslant 0$ un $y_2\geqslant 0$ šajā piemērā ir ingredientu daudzumi, kas dienas barības devā pārsniedz ingredientu minimālos daudzumus.

4°. Jebkuru lineāru algebrisku vienādojumu sistēmu var atvietot ar tai ekvivalentu nevienādību sistēmu, jo jebkuru vienādību, piemēram, $v_1=5$, var atvietot ar divām nevienādībām: $v_1\geqslant 5$ un $v_1\leqslant 5$. Ja ir dota, piemēram, triju vienādojumu sistēma

$$\left.\begin{aligned} v_1&=-x_1\qquad\quad+\ 4x_3=16,\\ v_2&=\ 4x_1+2x_2-13x_3=\ 2,\\ v_3&=\ 2x_1-\ x_2-\ 3x_3=\ 5,\end{aligned}\right\}\tag{2.2.2}$$

tad jebkuru no šīs sistēmas vienādojumiem var atvietot ar divām nevienādībām. Arī visu triju vienādojumu summa $s=v_1+v_2+$ $+v_3=23$ ir ekvivalenta ar divām nevienādībām $s\geqslant 23$ un $s\leqslant 23$. Ja $s\leqslant 23$, tad $v_1+v_2+v_3\leqslant 23$ un $v_1\leqslant 23-(2+5)=16$. Tad arī $v_2\leqslant 23-(5+16)=2$ un $v_3\leqslant 23-(16+2)=5$. Tādēļ vienādojumu sistēma (2.2.2) ir ekvivalenta ar nevienādību sistēmu

$$\left.\begin{aligned} -x_1\qquad\quad+\ 4x_3&\leqslant 16,\\ 4x_1+2x_2-13x_3&\leqslant\ 2,\\ 2x_1-\ x_2-\ 3x_3&\leqslant\ 5,\\ 5x_1+\ x_2-12x_3&\geqslant 23.\end{aligned}\right\}$$

Vispārīgā gadījumā lineāru algebrisku vienādojumu sistēma

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

ir ekvivalenta ar nevienādību sistēmu

$$\left.\begin{aligned} &\sum_{j=1}^{n} a_{ij}x_j\leqslant b_i\quad (i=1,2,\ldots,m),\\ &\sum_{j=1}^{n} s_jx_j\geqslant b_{m+1},\end{aligned}\right\}$$

kur

$$s_j=\sum_{i=1}^{m} a_{ij}\quad (j=1,2,\ldots,n),$$

$$b_{m+1}=\sum_{i=1}^{m} b_i.$$

Tas nozīmē, ka jebkuru kanoniskā formā dotu lineārās programmēšanas uzdevumu var pārveidot vispārīgajā formā vai modificētajā normālformā.

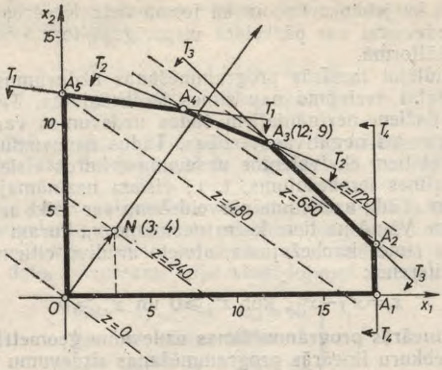
5°. Formulējot lineārās programmēšanas uzdevumus, var gadīties, ka daļai nezināmo nav zīmes ierobežojuma. Tas nozīmē, ka vismaz dažiem nezināmajiem šādos uzdevumos var būt gan pozitīvas, gan arī negatīvas vērtības. Tādus uzdevumus vienmēr var pārveidot tiem ekvivalentos uzdevumos, kuros visiem nezināmajiem ir zīmes ierobežojums, t. i., visiem nezināmajiem jābūt nenegatīviem. Tādu uzdevuma pārveidošanu var veikt ar dažādiem paņēmieniem. Vienā no tiem katru nezināmo x_j, kuram dotajā uzdevumā nav zīmes ierobežojuma, atvieto ar divu citu nenegatīvu nezināmo diferenci:

$$x_j = x'_j - x''_j, \text{ kur } x'_j \geqslant 0 \text{ un } x''_j \geqslant 0.$$

2.3. §. Lineārās programmēšanas uzdevumu ģeometriskā interpretācija. Jebkuru lineārās programmēšanas uzdevumu var viegli interpretēt ģeometriski un atrisināt grafiski, ja ierobežojumu sistēmā ir tikai divi nezināmie lielumi x_1 un x_2. Kā zināms, katru divu skaitļu pāri var interpretēt kā punktu vai arī kā vektoru taisnleņķa koordinātu sistēmā plaknē. Šādā koordinātu sistēmā katrs lineārs algebrisks vienādojums $a_{i1}x_1 + a_{i2}x_2 = b_i$ ir taisnes vienādojums. Katra taisne sadala koordinātu plakni divās pusplaknēs. Ja kāda brīvi izraudzīta punkta P_0 $(x_1; x_2)$ koordinātes ievieto dotajā taisnes vienādojumā, tad dabū skaitli $u_i = b_i - a_{i1}x_1 - a_{i2}x_2$. Šis skaitlis $u_i = 0$ tad un tikai tad, ja punkts P_0 $(x_1; x_2)$ atrodas uz dotās taisnes. Katrā citā gadījumā $u_i > 0$ vai $u_i < 0$. Tādēļ katru ierobežojumu $a_{i1}x_1 + a_{i2}x_2 \leqslant b_i$ apmierina tie un tikai tie punkti, kuri atrodas uz taisnes $a_{i1}x_1 + a_{i2}x_2 = b_i$ vai vienā (bet ne otrā) no pusplaknēm, kurās šī taisne sadala koordinātu plakni. Ierobežojumu sistēmā atsevišķi izdalītie nezināmo nenegativitātes nosacījumi ģeometriski nozīmē to, ka jebkuru no sistēmas ierobežojumiem var apmierināt tikai tie punkti, kuri atrodas koordinātu sistēmas pirmajā kvadrantā. Pusplaknēm citai citu pārsedzot (šķeļoties), rodas to punktu ģeometriskā vieta (punktu kopa), kurai piederīgie punkti apmierina visus dotos ierobežojumus, ja vien dotā ierobežojumu sistēma ir saderīga, t. i., nav pretrunīga. Šo punktu kopu sauc par lineārās programmēšanas uzdevuma mērķa funkcijas *definīcijas apgabalu* vai ierobežojumu sistēmas *atrisinājumu apgabalu*.

Ja ierobežojumu sistēma ir saderīga, tās atrisinājumu apgabalā ir bezgalīgi daudz punktu vai arī viens vienīgs punkts. Ja sistēma nav saderīga, tad nav neviena tāda punkta, kas apmierinātu visus dotos ierobežojumus. Pēdējā gadījumā saka, ka atrisinājumu apgabals ir tukša kopa. Tādam uzdevumam nav neviena reāla atrisinājuma.

Kā piemēru aplūkosim uzdevuma 1.2.1° ģeometrisko interpretāciju, kas attēlota zīmējumā 2.1. Šajā piemērā ierobežojumu



Zīm. 2.1.

sistēmas atrisinājumu apgabals ir punktu kopa, kuru veido sešas pusplaknes, citai citu pārsedzot, resp., tas ir sešu pusplakņu šķēlums, jo ierobežojumu sistēmā (1.2.3), (1.2.4) ir pavisam 6 ierobežojumi.

Zīmējumā 2.1 taisnleņķa koordinātu sistēmā attēlotā taisne T_1T_1, kuras vienādojums ir $x_1+7x_2=84$, saskaņā ar ierobežojumu sistēmas (1.2.3) pirmo nevienādību $x_1+7x_2\leqslant 84$ sadala koordinātu plakni divās *pusplaknēs (pustelpās)* — *pozitīvajā* un *negatīvajā*. Jebkura pozitīvajai pusplaknei T_1^+ piederīga punkta koordinātes apmierina doto nevienādību, bet to neapmierina neviena negatīvajai pusplaknei T_1^- piederīga punkta koordinātes.

Lai noteiktu, kura no abām pusplaknēm (pustelpām) ir pozitīvā, var ievietot tajā nevienādībā, kas sadala plakni (telpu) pusplaknēs (pustelpās) T^+ un T^-, jebkura brīvi izraudzīta punkta koordinātes. Ja šī nevienādība izpildās, izraudzītais punkts atrodas pustelpā T^+, pretējā gadījumā — pustelpā T^-.

Tā pusplakne, kurai piederīgie punkti apmierina ierobežojumu sistēmas (1.2.3) pirmo ierobežojumu $x_1+7x_2\leqslant 84$, apzīmēta ar bultiņām pie taisnes T_1T_1. Analogi attēloti visi pārējie ierobežojumi un norādītas tās pusplaknes, kurām piederīgie punkti apmierina attiecīgo ierobežojumu. Pēc tam zīmējumā ar resnu līniju norobežota koordinātu plaknes to punktu kopa, kuri pieder visām 6 pusplaknēm. Tie ir punkti, kuri atrodas daudzstūrī $OA_1A_2A_3A_4A_5$ un uz tā malām.

Mērķa funkciju $z=c_1x_1+c_2x_2$ jebkurai fiksētai z vērtībai, piemēram, $z=c_0$, ģeometriski var interpretēt kā taisni x_1Ox_2 plaknē. Katra šāda taisne ir perpendikulāra pret taisni, kas iet caur koordinātu sākumu un punktu $(\lambda c_1;\ \lambda c_2)$, kur λ — jebkurš pozitīvs skaitlis. Citādi sakot, mērķa funkciju ģeometriski var interpretēt kā tādu taišņu (līmeņa līniju) saimi, kas perpendikulāras pret vektoru $N=(\lambda c_1;\ \lambda c_2)$. Šis vektors N ģeometriski nozīmē orientētu taisnes nogriezni, kura sākumpunkts sakrīt ar koordinātu sākumu un galapunkts ir punktā $(\lambda c_1;\ \lambda c_2)$. Vektora vērsums norāda to virzienu definīcijas apgabalā, kurā mērķa funkcijas vērtība aug.

Uzdevuma 1.2.1° mērķa funkcija ir $z=30x_1+40x_2$. Tātad katra šīs mērķa funkcijas līmeņa līniju saimes taisne ir perpendikulāra pret vektoru $N=(30\lambda;\ 40\lambda)$. Ņemot, piemēram, $\lambda=0{,}1$, zīmējumā 2.1 attēlots vektors $N=(3;\ 4)$ un dažas tam perpendikulāras mērķa funkcijas līmeņa līnijas. Zīmējumā redzams, ka mērķa funkcija z visaugstāko līmeni (vislielāko vērtību) sasniedz punktā A_3 (12; 9), resp., $\max z=30\cdot 12+40\cdot 9=720$.

Zīmējumā 2.2 attēlota uzdevuma 1.3.1° ģeometriskā interpretācija, iepriekš pārveidojot ierobežojumu sistēmu (1.3.3) citā, dotajai sistēmai ekvivalentā ierobežojumu sistēmā

$$\left.\begin{aligned} 4x_1+5x_2 &\geqslant 10, \\ 5x_1+x_2 &\geqslant 3. \end{aligned}\right\} \qquad (2.3.1)$$

Ierobežojumu sistēma (2.3.1) ir ekvivalenta ar sistēmu (1.3.3), jo nevienādība nemainās, ja tās abas puses reizina ar vienu un to pašu pozitīvu skaitli. Šajā gadījumā sistēmas (1.3.3) pirmā nevienādība reizināta ar 4, bet otrā — ar 1/80. Dotā mērķa funkcija $z=5x_1+4x_2$ aug virzienā, kuru norāda vektors $N=(5\lambda;\ 4\lambda)$. Vektors N zīmējumā attēlots, izvēloties $\lambda=\frac{1}{5}$. Zīmējumā redzams,

Zīm. 2.2.

kā mērķa funkcija šajā piemērā var augt neierobežoti, turpretī no apakšas tā ir ierobežota un sasniedz minimumu punktā $A_2\left(\frac{5}{21}; \frac{38}{21}\right)$, kas ir aprēķinātais taišņu T_1T_1 un T_2T_2 krustpunkts. Aprēķinot šim punktam atbilstošo mērķa funkcijas z vērtību, atrodam, ka $\min z = 5 \cdot \frac{5}{21} + 4 \cdot \frac{38}{21} = \frac{59}{7}$.

Uzdevums 2.3.1°. Grafiski atrast maksimumu un minimumu lineārai funkcijai $z = x_1 - x_2$ pie ierobežojumiem

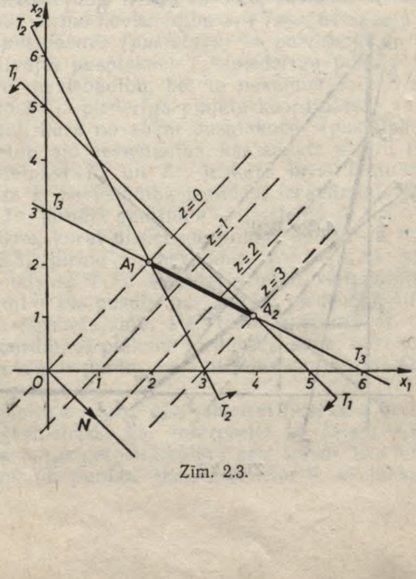
$$\left.\begin{aligned} x_1 + x_2 &\leqslant 5, \\ 2x_1 + x_2 &\geqslant 6, \\ x_1 + 2x_2 &= 6, \end{aligned}\right\}$$

$$x_1 \geqslant 0; \quad x_2 \geqslant 0.$$

Atrisinājums. Uzdevuma ģeometriskā interpretācija dota zīmējumā 2.3. Ierobežojumu sistēmas atrisinājumu apgabals šajā uzdevumā ir taisnes T_3T_3 nogrieznis, kas atrodas starp taisnēm T_1T_1 un T_2T_2. Zīmējumā nolasāms, ka lineārajai funkcijai ir minimums punktā A_1 (2; 2) un maksimums punktā A_2 (4; 1). Tātad $\max z = 3$ un $\min z = 0$.

Uzdevums 2.3.2°. Grafiski atrast maksimumu un minimumu lineārai funkcijai

$$z = x_1 - x_2 \qquad (2.3.2)$$



Zīm. 2.3.

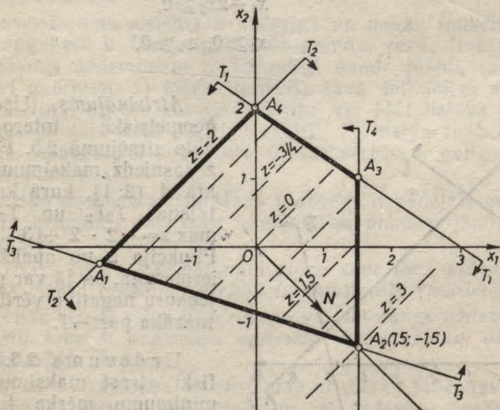
pie ierobežojumiem

$$\begin{aligned} 2x_1+3x_2&\leqslant 6,\\ -x_1+\ x_2&\leqslant 2,\\ -x_1-3x_2&\leqslant 3,\\ -x_1\qquad &\geqslant -1{,}5. \end{aligned} \qquad (2.3.3)$$

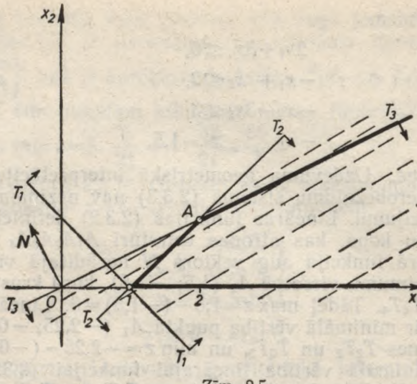
Atrisinājums. Uzdevuma ģeometriskā interpretācija dota zīmējumā 2.4. Ierobežojumu sistēmā (2.3.3) nav nezināmo nenegativitātes nosacījumu. Lineārās funkcijas (2.3.2) definīcijas apgabals ir punktu kopa, kas atrodas četrstūrī $A_1A_2A_3A_4$ un uz tā malām. Lineārā funkcija aug vektora $\boldsymbol{N}$ norādītajā virzienā un sasniedz maksimumu virsotnē A_2 (1,5; −1,5), kurā krustojas taisnes T_3T_3 un T_4T_4. Tādēļ $\max z=1{,}5-(-1{,}5)=3$. Lineārajai funkcijai (2.3.2) ir minimālā vērtība punktā A_1 (−2,25; −0,25), kurā krustojas taisnes T_2T_2 un T_3T_3, un $\min z=-2{,}25-(-0{,}25)=-2$. Tāda pati minimālā vērtība lineārajai funkcijai (2.3.2) ir arī punktā A_4 (0; 2), kurā krustojas taisnes T_1T_1 un T_2T_2. Funkcijas vērtība ir $\min z=-2$ acīm redzami arī jebkurā punktā, kas atrodas uz taisnes nogriežņa A_1A_4. Tas nozīmē, ka dotajai lineārajai funkcijai ir minimums jebkurā punktā, kura koordinātes aprēķināmas pēc formulām

$$x_1=-2{,}25(1-\lambda)+0\cdot\lambda,$$
$$x_2=-0{,}25(1-\lambda)+2\cdot\lambda.$$

Te λ ir jebkurš reāls skaitlis no intervāla $0\leqslant\lambda\leqslant 1$.



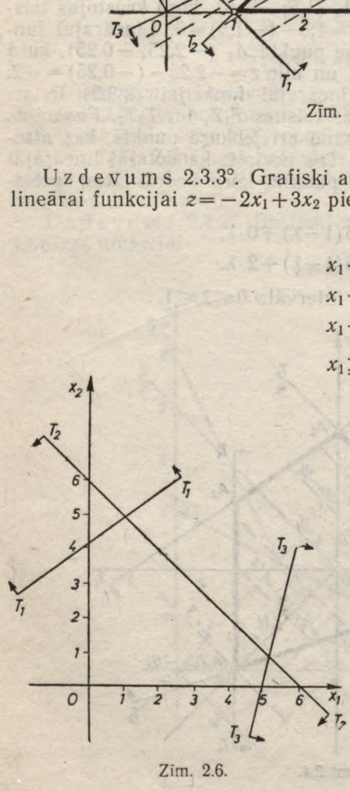
Zīm. 2.4.



Zīm. 2.5.

Uzdevums 2.3.3°. Grafiski atrast maksimumu un minimumu lineārai funkcijai $z=-2x_1+3x_2$ pie ierobežojumiem

$$x_1+x_2\geqslant 1,$$
$$x_1-x_2\geqslant 1,$$
$$x_1-2x_2\geqslant 0,$$
$$x_1\geqslant 0,\ x_2\geqslant 0.$$



Zīm. 2.6.

Atrisinājums. Uzdevuma ģeometriskā interpretācija dota zīmējumā 2.5. Funkcija z sasniedz maksimumu punktā A (2; 1), kurā krustojas taisnes T_2T_2 un T_3T_3, un $\max z=-2\cdot 2+3\cdot 1=-1$. Funkcija z no apakšas nav ierobežota, un tā var pieņemt jebkuru negatīvu vērtību, kas mazāka par -1.

Uzdevums 2.3.4°. Grafiski atrast maksimumu un minimumu mērķa funkcijai $z=c_1x_1+c_2x_2$ pie ierobežojumiem

$$x_1 \geqslant 0,\ x_2 \geqslant 0,$$
$$-2x_1+3x_2 \geqslant 12,$$
$$x_1+x_2 \leqslant 6,$$
$$4x_1-x_2 \geqslant 20.$$

Atrisinājums. Uzdevuma ierobežojumi ģeometriski interpretēti zīmējumā 2.6. No zīmējuma redzams, ka ierobežojumu sistēma nav saderīga un mērķa funkcijas definīcijas apgabals ir tukša punktu kopa. Tādēļ šim uzdevumam atrisinājums neeksistē.

2.4. §. Plāna, optimālā plāna un atbalsta plāna jēdzieni. Lineārajā programmēšanā par *plānu* sauc jebkuru nezināmo vērtību sakārtotu kopu, kas apmierina visus dotā uzdevuma ierobežojumus. Ģeometriskajā interpretācijā plāns ir katrs tāds punkts, kas pieder pie mērķa funkcijas definīcijas apgabala, resp., pie ierobežojumu sistēmas atrisinājumu apgabala. Tādēļ dotajam lineārās programmēšanas uzdevumam ir vai nu bezgalīgi daudz plānu, vai viens vienīgs plāns, vai arī nav neviena plāna. Pēdējā gadījumā dotā ierobežojumu sistēma ir nesaderīga. Ģeometriski tas nozīmē, ka mērķa funkcijas definīcijas apgabals ir tukša punktu kopa.

Ja ierobežojumu sistēma ir gan saderīga, bet tās atrisinājumu apgabalā ietilpst tikai viens vienīgs punkts, tad uzdevumam ir tikai viens vienīgs plāns un mērķa funkcijai ir viena vienīga vērtība. Tādā gadījumā nav lineārās programmēšanas uzdevuma, jo nav plāna variantu.

Ja ierobežojumu sistēma ir saderīga un mērķa funkcijas definīcijas apgabalā ir bezgalīgi daudz punktu, resp., lineārās programmēšanas uzdevumam ir bezgalīgi daudz plānu, tad iespējami divi gadījumi: 1) mērķa funkcija savā definīcijas apgabalā nav ierobežota, un tās absolūtā vērtība var kļūt lielāka par jebkuru iepriekš dotu lielu pozitīvu skaitli; 2) mērķa funkcija savā definīcijas apgabalā ir ierobežota, un tās vērtība ir galīgs lielums jebkuram plānam.

Praktisku uzdevumu risināšanā nozīmīgs ir pēdējais gadījums, kad starp bezgalīgi daudziem iespējamiem plāniem jāatrod optimālais plāns.

D e f i n ī c i j a 1°. *Par optimālo plānu sauc tādu plānu, kuram atbilstošā mērķa funkcijas vērtība ir maksimālā (minimālā).*

D e f i n ī c i j a 2°. *Par lineārās programmēšanas uzdevuma atrisinājumu sauc šī uzdevuma optimālo plānu ar tam atbilstošo mērķa funkcijas vērtību.*

Atrast dotā uzdevuma atrisinājumu ar mērķa funkcijas vērtību salīdzināšanas metodi nav iespējams, jo saderīga lineārās programmēšanas uzdevuma plānu skaits ir bezgalīgi liels. Kā vēlāk redzēsim, optimālais plāns nav jāmeklē starp bezgalīgi daudziem iespējamiem plāniem. Pietiek, ja atrod vienu no t. s. atbalsta plāniem. To pakāpeniski uzlabojot, var atrast optimālo plānu, ja tāds dotajam uzdevumam vispār eksistē.

Definīcija 3°. *Par bāzes mainīgo (pretstatā nebāzes mainīgajam) kanoniskā formā pārveidotā lineārās programmēšanas uzdevumā sauc tādu šī uzdevuma nenegatīvu nezināmo lielumu, kas izslēgts no mērķa funkcijas un no visiem ierobežojumu sistēmas vienādojumiem, izņemot vienu vienādojumu, kurā šim nezināmajam ir koeficients 1, kad vienādojumam ir nenegatīva labā puse (brīvais loceklis).*

Ierobežojumu sistēmā, kurā ir m sanumurēti vienādojumi, nevar būt vairāk kā m bāzes mainīgo, turklāt, ja bāzes mainīgos sanumurē ar indeksiem $1, 2, \ldots, m$, tad bāzes mainīgais ar indeksu i atrodas tajā un tikai tajā vienādojumā, kura numurs ir i $(i=1, 2, \ldots, m)$.

Definīcija 4°. *Par atbalsta plānu sauc tādu lineārās programmēšanas uzdevuma ierobežojumu sistēmas atrisinājumu, kurā visi bāzes mainīgie vienādi ar tiem atbilstošo ierobežojumu labajām pusēm, bet visi nebāzes mainīgie vienādi ar nulli.*

Definīcija 5°. *Lineārās programmēšanas uzdevumā atbalsta plānu ar tam atbilstošo mērķa funkcijas vērtību sauc par šī uzdevuma bāzes atrisinājumu.*

Tā, piemēram, ierobežojumu sistēmā (2.2.1) bāzes mainīgie ir nezināmie y_1, y_2, y_3 un y_4, nebāzes mainīgie — x_1 un x_2. Viens no uzdevuma 1.2.1° atbalsta plāniem ir $x_1=x_2=0$, $y_1=84$, $y_2=69$, $y_3=105$, $y_4=72$. Šim atbalsta plānam atbilstošā mērķa funkcijas vērtība ir $z=0$.

3. nodaļa

DAŽAS LINEĀRU ALGEBRISKU VIENĀDOJUMU SISTĒMU SKAITLISKĀS ATRISINĀŠANAS METODES

3.1. §. Lineāru algebrisku vienādojumu sistēma. Jebkura kanoniskā formā pārveidota lineārās programmēšanas uzdevuma ierobežojumu sistēma ir lineāra m vienādojumu sistēma ar n nezināmajiem:

$$\left.\begin{array}{l} a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n=b_1, \\ a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n=b_2, \\ \cdots\cdots\cdots\cdots\cdots \\ a_{m1}x_1+a_{m2}x_2+\ldots+a_{mn}x_n=b_m \end{array}\right\} \qquad (3.1.1)$$

jeb saīsinātā pierakstā

$$a_{i1}x_1+a_{i2}x_2+\ldots+a_{in}x_n=b_i \quad (i=1, 2, \ldots, m).$$

Lietojot summas simbolu, sistēmas (3.1.1) pierakstu var vēl saīsināt:

$$\sum_{j=1}^{n} a_{ij}x_j=b_i \quad (i=1, 2, \ldots, m).$$

Vienādojumu sistēmas *koeficienti* a_{ij} un *brīvie locekļi* b_i ir doti reāli skaitļi. Katram vienādojumu sistēmas koeficientam sistēmas vispārīgajā pierakstā, kā redzams, ir divi indeksi, ar kuriem šie koeficienti savstarpēji atšķiras. Pirmais indekss sakrīt ar vienādojuma numuru sistēmā, bet otrs — ar tā nezināmā numuru, pie kura dotais koeficients atrodas. Tā, piemēram, a_{35} ir trešā vienādojuma piektā nezināmā x_5 koeficients, šī vienādojuma brīvais loceklis ir b_3.

Lineāru algebrisku vienādojumu sistēmu skaitliskā atrisināšanā iespējami 3 gadījumi: a) sistēma ir nesaderīga, resp., tai nav neviena atrisinājuma; b) sistēmai ir viens vienīgs atrisinājums; c) tai ir bezgalīgi daudz atrisinājumu. Kuram no šiem gadījumiem atbilst dotā vienādojumu sistēma, — tas noskaidrojas, meklējot sistēmas atrisinājumu ar kādu no lineārās algebras skaitliskajām metodēm. No tādām metodēm aplūkosim t. s. modificēto Žordāna izslēgšanas metodi un Gausa—Žordāna metodi.

3.2. §. Modificētā Žordāna izslēgšanas metode. Lai risinātu m lineāru algebrisku vienādojumu sistēmu (3.1.1) ar n nezināmajiem pēc modificētās Žordāna izslēgšanas metodes, papildināsim katru dotās sistēmas vienādojumu ar vienu jaunu nezināmo un aplūkosim iegūto paplašināto m vienādojumu sistēmu ar $m+n$ nezināmajiem:

$$\left.\begin{array}{llllll} y_1 & +a_{11}x_1 & +a_{12}x_2 & +\ldots & +a_{1n}x_n & =b_1, \\ \quad y_2 & +a_{21}x_1 & +a_{22}x_2 & +\ldots & +a_{2n}x_n & =b_2, \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \qquad y_m & +a_{m1}x_1 & +a_{m2}x_2 & +\ldots & +a_{mn}x_n & =b_m. \end{array}\right\} \quad (3.2.1)$$

Te nezināmos y_i ($i=1, 2, \ldots, m$) sauksim par sistēmas (3.2.1) *bāzes mainīgajiem* pretstatā nezināmajiem x_j ($j=1, 2, \ldots, n$), kurus sauksim par šīs sistēmas *nebāzes mainīgajiem*. Sistēmai (3.2.1) ir bezgalīgi daudz atrisinājumu, jo katrai brīvi izraudzītai nebāzes mainīgo vērtību kopai atbilst viena, pilnīgi noteikta bāzes mainīgo vērtību kopa, kas apmierina doto vienādojumu sistēmu. Izvēloties, piemēram, visus nebāzes mainīgos vienādus ar nulli, visi bāzes mainīgie ir identiski vienādi ar sistēmas (3.2.1) labajām pusēm.

Acīm redzami, vienādojumu sistēma (3.1.1) ir saderīga tad un tikai tad, ja var uzrādīt kaut vienu tādu nezināmo $x_1, x_2, \ldots, x_n$ vērtību kopu, kas apmierina vienādojumu sistēmu (3.2.1) pie nosacījuma $y_1=y_2=\ldots=y_m=0$.

Vienādojumu sistēmu (3.2.1) var pierakstīt tabulas formā:

y_1	y_2	$\dots$	y_m	x_1	x_2	$\dots$	x_n	$+1$
1	0	$\dots$	0	a_{11}	a_{12}	$\dots$	a_{1n}	b_1
0	1	$\dots$	0	a_{21}	a_{22}	$\dots$	a_{2n}	b_2
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
0	0	$\dots$	1	a_{m1}	a_{m2}	$\dots$	a_{mn}	b_m

(3.2.2)

Tabulas (3.2.2) vietā bieži lieto tās kanonisko formu:

	1	-1	x_1	x_2	$\dots$	$\boxed{x_k}$	$\dots$	x_n	Rindu numuri
	y_1	b_1	a_{11}	a_{12}	$\dots$	a_{1k}	$\dots$	a_{1n}	1
	y_2	b_2	a_{21}	a_{22}	$\dots$	a_{2k}	$\dots$	a_{2n}	2
	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.
*	y_r	b_r	a_{r1}	a_{r2}	$\dots$	$\boxed{a_{rk}}$	$\dots$	a_{rn}	r
	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.	.
	y_m	b_m	a_{m1}	a_{m2}	$\dots$	a_{mk}	$\dots$	a_{mn}	m

(3.2.3)

Tā kā katrā lineārā algebriskā vienādojumā vismaz viens koeficients nav nulle, tad katrā tabulas (3.2.3) rindā vismaz viens skaitlis nav nulle, un, tā kā katrs nezināmais ietilpst vismaz vienā sistēmas vienādojumā, tad arī katrā tabulas kolonnā, neieskaitot brīvo locekļu kolonnu, vismaz viens skaitlis nav nulle.

Jebkurš no tabulas (3.2.3) skaitļiem atrodas kādas rindas un kolonnas krustojumā. Ja izvēlas tādu kolonnu un rindu, kuru krustojumā novietotais elements nav nulle, piemēram, $a_{rk} \neq 0$, tad ir iespējams izslēgt nezināmo x_k no nebāzes mainīgo skaita, apmainot to lomām ar bāzes mainīgo y_r. To panāk, izsakot no sistēmas (3.2.1) r-tā vienādojuma

$$x_k = \frac{b_r}{a_{rk}} - \sum_{j=1}^{k-1} \frac{a_{rj}}{a_{rk}} x_j - \frac{y_r}{a_{rk}} - \sum_{j=k+1}^{n} \frac{a_{rj}}{a_{rk}} x_j \tag{3.2.4}$$

un ievietojot šo x_k izteiksmi visos pārējos sistēmas (3.2.1) vienādojumos. Tad iegūst jaunu vienādojumu sistēmu, kas ekvivalenta ar sistēmu (3.2.1). Jauno vienādojumu sistēmu pierakstīsim šādas tabulas formā:

1	-1	x_1	x_2	$\cdots$	y_r	$\cdots$	x_n	Nr.
y_1	b'_1	a'_{11}	a'_{12}	$\cdots$	$-\gamma a_{1k}$	$\cdots$	a'_{1n}	1
y_2	b'_2	a'_{21}	a'_{22}	$\cdots$	$-\gamma a_{2k}$	$\cdots$	a'_{2n}	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_k	γb_r	γa_{r1}	γa_{r2}	$\cdots$	$\gamma\cdot 1$	$\cdots$	γa_{rn}	r
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
y_m	b'_m	a'_{m1}	a'_{m2}	$\cdots$	$-\gamma a_{mk}$	$\cdots$	a'_{mn}	m

(3.2.5)

Tabulā (3.2.5)

$$\gamma=\frac{1}{a_{rk}}, \tag{3.2.6}$$

$$a'_{ij}=a_{ij}-\frac{a_{ik}\cdot a_{rj}}{a_{rk}} \quad (i\neq r;\ j\neq k), \tag{3.2.7}$$

$$b'_i=b_i-\frac{b_r\cdot a_{ik}}{a_{rk}} \quad (i\neq r). \tag{3.2.8}$$

Tiešām, ievietojot x_k izteiksmi (3.2.4) visos pārējos sistēmas (3.2.1) vienādojumos, iegūst

$$y_i=b_i-\sum_{j=1}^{k-1}a_{ij}x_j-\frac{a_{ik}}{a_{rk}}\left(b_r-\sum_{j=1}^{k-1}a_{rj}x_j-y_r-\sum_{j=k+1}^{n}a_{rj}x_j\right)-\sum_{j=k+1}^{n}a_{ij}x_j$$

jeb

$$y_i-b'_i+\sum_{j=1}^{k-1}a'_{ij}x_j-\gamma a_{ik}y_r+\sum_{j=k+1}^{n}a'_{ij}x_j=0$$

$$(i=1, 2, \ldots, r-1, r+1, \ldots, m). \tag{3.2.9}$$

Izraudzīto sākuma tabulas skaitli $a_{rk}\neq 0$, kas atrodas r-tās rindas un k-tās kolonnas krustojumā, sauc par šīs tabulas *galveno elementu.* Jebkuru vienādojumu sistēmas koeficientu, kas nav nulle, var vai nu pilnīgi brīvi, vai arī, vadoties no kādiem īpašiem apsvērumiem, izvēlēties par galveno elementu. To rindu un kolonnu, kuru krustojumā atrodas tabulas galvenais elements, sauc par *galveno rindu* un *galveno kolonnu.*

Par *modificētās Žordāna izslēgšanas metodes vienu soli* ar galveno elementu $a_{rk}\neq 0$ sauc r-tā bāzes mainīgā y_r apmaiņu lomām ar nebāzes mainīgo x_k.

Tātad izpildīt vienu modificētās Žordāna izslēgšanas metodes soli ar galveno elementu a_{rk} nozīmē atvietot sākuma tabulu (3.2.3) ar jaunu tabulu (3.2.5). Vienā solī ietilpst šādas darbības:

1) atvietot sākuma tabulas galveno elementu ar tā apgriezto lielumu

$$\gamma=\frac{1}{a_{rk}};$$

2) atvietot pārējos galvenās rindas elementus ar skaitļiem, kurus iegūst, reizinot šīs rindas elementus ar γ;

3) atvietot pārējos galvenās kolonnas elementus ar skaitļiem, kurus iegūst, reizinot šīs kolonnas elementus ar $-\gamma$;

4) atvietot visus citus sākuma tabulas elementus, kas neatrodas šīs tabulas galvenajā rindā vai galvenajā kolonnā, ar skaitļiem, kurus aprēķina pēc formulām (3.2.7) vai (3.2.8).

Formulu (3.2.7) un (3.2.8) vai to paveidu atvieglotai iegaumēšanai pastāv dažādas kārtulas. Viena no tām ir «taisnstūra kārtula». Lietojot šo kārtulu, jāievēro, ka katrā no formulām (3.2.7), (3.2.8) iesaistīti četri sākuma tabulas skaitļi: 1) atvietojamais elements, 2) galvenais elements, 3) tas galvenās rindas elements, kas atrodas vienā kolonnā ar atvietojamo elementu, un 4) tas galvenās kolonnas elements, kas atrodas vienā rindā ar atvietojamo elementu. Grafiski šie 4 skaitļi katrā konkrētā gadījumā atrodas kāda taisnstūra virsotnēs. Taisnstūra galvenā diagonāle ir tā, kas savieno atvietojamo elementu ar galveno elementu. Otrā, blakus diagonāle savieno pārējos divus elementus. Ievērojot to, taisnstūra kārtulu var formulēt šādi:

Jaunās tabulas elementu a'_{ij} $(i\neq r, j\neq k)$ iegūst, ja no vecās tabulas elementa a_{ij} atņem blakus diagonāles galos novietoto skaitļu reizinājumu, kas dalīts ar galveno elementu.

Piemērs 3.2.1°. Atrisināt vienādojumu sistēmu

$$\left.\begin{aligned} x_1-\ x_2-\ x_3&=0,\\ 3x_1-2x_2-\ x_3&=4,\\ -2x_1+3x_2+3x_3&=5. \end{aligned}\right\} \qquad (3.2.10)$$

Atrisinājums. Ievedīsim katrā sistēmas (3.2.10) vienādojumā vienu jaunu nezināmo ar koeficientu $+1$ un aplūkosim sistēmas (3.2.10) vietā paplašinātu vienādojumu sistēmu

$$\left.\begin{aligned} y_1+\ x_1-\ x_2-\ x_3&=0,\\ y_2+3x_1-2x_2-\ x_3&=4,\\ y_3-2x_1+3x_2+3x_3&=5, \end{aligned}\right\} \qquad (3.2.11)$$

kurā y_1, y_2 un y_3 ir bāzes mainīgie pretstatā nebāzes mainīgajiem x_1, x_2 un x_3.

Šo vienādojumu sistēmu var pārrakstīt tabulas formā:

	1	-1	$\boxed{x_1}$	x_2	x_3	(I)
	y_1	0	1	-1	-1	1
	y_2	4	3	-2	-1	2
*	y_3	5	$\boxed{-2}$	3	3	3

Ja par galveno kolonnu un galveno rindu izvēlamies attiecīgi nebāzes mainīgā x_1 koeficientu kolonnu un 3. rindu, tad galvenais elements ir skaitlis -2, kas tabulā apvilkts ar četrstūri. Izpildot ar šo galveno elementu vienu modificētās Žordāna izslēgšanas metodes soli, iegūst jaunu tabulu (II), kurā mainīgie x_1 un y_3 apmainīti lomām:

	1	-1	y_3	x_2	$\boxed{x_3}$	(II)
*	y_1	2,5	0,5	0,5	$\boxed{0{,}5}$	1
	y_2	11,5	1,5	2,5	3,5	2
	x_1	$-2{,}5$	$-0{,}5$	$-1{,}5$	$-1{,}5$	3

Aizpildot jauno tabulu (II), vispirms apmaina vietām x_1 un y_3 un ieraksta galvenā elementa vietā apgriezto lielumu $-0{,}5$. Pēc tam atvieto galvenās rindas pārējos elementus ar skaitļiem $-2{,}5$; $-1{,}5$; $-1{,}5$, kurus iegūst, dalot galvenās rindas elementus 5; 3 un 3 ar galveno elementu -2 vai reizinot ar tā apgriezto lielumu $\gamma=-0{,}5$. Pēc tam atvieto galvenās kolonnas pārējos elementus 1 un 3 ar skaitļiem 0,5 un 1,5. Tos iegūst, reizinot 1 un 3 ar $-\gamma=$ $=0{,}5$. Beidzot visus pārējos tabulas (I) elementus atvieto ar skaitļiem

$$b'_1=0-\frac{1\cdot 5}{-2}=2{,}5;\ b'_2=4-\frac{3\cdot 5}{-2}=11{,}5;$$

$$a'_{12}=-1-\frac{1\cdot 3}{-2}=0{,}5;\ a'_{22}=-2-\frac{3\cdot 3}{-2}=2{,}5;$$

$$a'_{13}=a'_{12}=0{,}5;\ a'_{23}=-1-\frac{3\cdot 3}{-2}=3{,}5,$$

kurus iegūst pēc taisnstūra kārtulas.

Ar to noslēdzas pirmais modificētās Žordāna izslēgšanas metodes solis.

Otrajā solī izvēlēsimies par galveno elementu 0,5, kas tabulā (II) atbilstoši iezīmēts. Izpildot otro soli ar šo galveno elementu, iegūst tabulu (III):

1	-1	y_3	$\boxed{x_2}$	y_1	(III)
x_3	5	1	1	2	1
* y_2	-6	-2	$\boxed{-1}$	-7	2
x_1	5	1	0	3	3

Pēdējā, trešā izslēgšanas soļa galvenais elements ir -1. Izpildot ar šo galveno elementu trešo modificētās Žordāna izslēgšanas metodes soli, iegūst tabulu (IV):

1	-1	y_3	y_2	y_1	(IV)
x_3	-1	-1	1	-5	1
x_2	6	2	-1	7	2
x_1	5	1	0	3	3

Tabulā (IV) nolasāma šāda vienādojumu sistēma:

$$\left.\begin{aligned} x_3+1-\ y_3+y_2-5y_1&=0,\\ x_2-6+2y_3-y_2+7y_1&=0,\\ x_1-5+\ y_3\qquad\ +3y_1&=0. \end{aligned}\right\} \tag{3.2.12}$$

Vienādojumu sistēma (3.2.12) ir ekvivalenta ne vien ar sistēmu (3.2.11), bet arī ar doto vienādojumu sistēmu (3.2.10), ja $y_1 = y_2 = y_3 = 0$. Tādēļ dotā vienādojumu sistēma ir ekvivalenta ar vienādojumu sistēmu

$$\left.\begin{aligned} x_3+1&=0,\\ x_2-6&=0,\\ x_1-5&=0, \end{aligned}\right\}$$

un tās vienīgais atrisinājums ir $x_1=5$, $x_2=6$, $x_3=-1$, kas nolasāms tabulā (IV).

P i e m ē r s 3.2.2°. Atrisināt vienādojumu sistēmu

$$\left.\begin{aligned} x_1+3x_2-\ x_3&=2,\\ 2x_1+\ x_2+\ x_3&=-1,\\ -x_1+7x_2-5x_3&=8. \end{aligned}\right\} \tag{3.2.13}$$

Atrisinājums.

	1	−1	$\boxed{x_1}$	x_2	x_3	(I)
*	y_1	2	$\boxed{1}$	3	−1	1
	y_2	−1	2	1	1	2
	y_3	8	−1	7	−5	3

	1	−1	y_1	$\boxed{x_2}$	x_3	(II)
	x_1	2	1	3	−1	1
*	y_2	−5	−2	$\boxed{-5}$	3	2
	y_3	10	1	10	−6	3

1	−1	y_1	y_2	x_3	(III)
x_1	−1	−0,2	0,6	0,8	1
x_2	1	0,4	−0,2	−0,6	2
y_3	0	−3,0	2,0	0	3

Tabula (III), kura iegūta pēc diviem modificētās Žordāna izslēgšanas metodes soļiem, atbilst vienādojumu sistēmai

$$\left.\begin{aligned} x_1+1-0{,}2y_1+0{,}6y_2+0{,}8x_3&=0,\\ x_2-1+0{,}4y_1-0{,}2y_2-0{,}6x_3&=0,\\ y_3\ -\ 3y_1+\ 2y_2\qquad\quad &=0.\end{aligned}\right\} \tag{3.2.14}$$

Tā ir ekvivalenta ar doto sistēmu (3.2.13), ja $y_1=y_2=y_3=0$. Tas nozīmē, ka dotā vienādojumu sistēma ir ekvivalenta ar sistēmu

$$\left.\begin{aligned} x_1&=-1-0{,}8x_3,\\ x_2&=\ \ 1+0{,}6x_3.\end{aligned}\right\} \tag{3.2.15}$$

Vienādības (3.2.15) ir dotās *vienādojumu sistēmas vispārīgais atrisinājums*. Tā kā šajā piemērā $y_3=3y_1-2y_2$, tad $y_1=\frac{2}{3}y_2+$ $+\frac{1}{3}y_3$ un $y_2=\frac{3}{2}y_1-\frac{1}{2}y_3$. Tas nozīmē, ka vienādojumu sistēmā (3.2.13) jebkurš vienādojums ir pārējo divu vienādojumu secinājums. Tā, piemēram, otro vienādojumu iegūst, ja pirmo vienādojumu reizina ar $\frac{3}{2}$, trešo ar $-\frac{1}{2}$ un iegūtos vienādojumus saskaita. Tādēļ saka, ka šajā vienādojumu sistēmā ir tikai divi *lineāri neatkarīgi vienādojumi,* bet trešais ir pārējo divu vienādojumu *lineāra kombinācija.*

Risinot vienādojumu sistēmu ar Žordāna metodi, no pēdējā soļa tabulas redzams, vai sistēma ir saderīga, un nolasāms ne vien saderīgas sistēmas vispārīgais atrisinājums, bet atklājas arī iepriekš nepamanītas lineāras sakarības starp sistēmas vienādojumiem.

Piemērs 3.2.3°. Atrisināt vienādojumu sistēmu

$$\left.\begin{aligned} x_1+3x_2-x_3&=2,\\ 2x_1+x_2+x_3&=-1,\\ -x_1+7x_2-5x_3&=1. \end{aligned}\right\} \tag{3.2.16}$$

Atrisinājums.

	1	−1	$\boxed{x_1}$	x_2	x_3	(I)
*	y_1	2	$\boxed{1}$	3	−1	1
	y_2	−1	2	1	1	2
	y_3	1	−1	7	−5	3

	1	−1	y_1	$\boxed{x_2}$	x_3	(II)
	x_1	2	1	3	−1	1
*	y_2	−5	−2	$\boxed{-5}$	3	2
	y_3	3	1	10	−6	3

1	−1	y_1	y_2	x_3	(III)
x_1	−1	−1/5	3/5	4/5	1
x_2	1	2/5	−1/5	−3/5	2
y_3	−7	−3	2	0	3

No pēdējās tabulas redzams, ka vienādojumu sistēma (3.2.16) ir nesaderīga, jo izteiksmē $y_3=-7+3y_1-2y_2$ nav iespējama nosacījuma $y_1=y_2=y_3=0$ izpildīšanās.

3.3. §. Lineāru vienādojumu sistēmas risināšanas gaitas kontrole. Par lineāru vienādojumu sistēmas atrisinājuma pareizību var un vajag pārliecināties, ievietojot atrasto vispārīgo atrisinājumu dotajā vienādojumu sistēmā. Ja tas vienādojumu sistēmu neapmierina, tad risināšanas gaitā ir pieļauta kļūda. Katras vienādojumu sistēmas risināšana prasa relatīvi daudz laika un pūļu, un kļūdīšanās iespēja ir liela, jo sevišķi, kad jārisina daudzu vienādojumu sistēmas. Tādēļ ir lietderīgi kontrolēt katru izslēgšanas soli. Viens no vienkāršākajiem kontroles paņēmieniem ir kolonnu summu kontrole.

Ja vienādojumu sistēmas (3.1.1) beigās pievieno vienādojumu

$$(\sigma_1-1)x_1+(\sigma_2-1)x_2+\ldots+(\sigma_n-1)x_n=\sigma_0-1,$$

kur $\sigma_0=\sum_{i=1}^{m} b_i$; $\sigma_j=\sum_{i=1}^{m} a_{ij}$ $(j=1,2,\ldots,n)$, un līdz ar to tabulai (3.2.3) pievieno $(m+1)$-mo rindu, kurā ieraksta katrā kolonnā esošo skaitļu summu, atņemot no tās 1, tad, izpildot vienu izslēgšanas soli, noapaļošanas kļūdu robežās ir spēkā vienādības

$$(\sigma_j-1)'=\sigma'_j-1,\ (j=1,2,\ldots,n).$$

Te $(\sigma_j-1)'$ ir skaitļi, ar kuriem atvieto vecās tabulas $(m+1)$-ās rindas elementus, izpildot vienu Žordāna izslēgšanas soli, bet σ'_j ir skaitļi, kurus iegūst, summējot visus j-tās kolonnas skaitļus jaunajā tabulā.

P i e m ē r s 3.3.1°. Izdarot kolonnu summu kontroli, atrisināt vienādojumu sistēmu ar saīsināto Žordāna izslēgšanas metodi.

$$\left.\begin{aligned} 2x_1+3x_2+11x_3+5x_4&=2,\\ x_1+x_2+5x_3+2x_4&=1,\\ 2x_1+x_2+3x_3+2x_4&=-3,\\ x_1+x_2+3x_3+4x_4&=-3. \end{aligned}\right\}$$

Atrisinājums. Pārrakstīsim doto vienādojumu sistēmu tabulas formā, pievienojot tai kolonnu summu kontroles rindu:

	1	−1	$\boxed{x_1}$	x_2	x_3	x_4	(I)
*	y_1	2	$\boxed{2}$	3	11	5	1
	y_2	1	1	1	5	2	2
	y_3	−3	2	1	3	2	3
	y_4	−3	1	1	3	4	4
	σ	−4	5	5	21	12	5

Apmainot lomām mainīgos y_1 un x_1, t. i., izdarot vienu modificētās Žordāna izslēgšanas metodes soli, iegūst jaunu tabulu, kurā atmetīsim jaunā nebāzes mainīgā y_1 koeficientu kolonnu.

1	−1	$\boxed{x_2}$	x_3	x_4	(II)
x_1	1	3/2	11/2	5/2	1
* y_2	0	$\boxed{-1/2}$	−1/2	−1/2	2
y_3	−5	−2	−8	−3	3
y_4	−4	−1/2	−5/2	3/2	4
σ	−9	−5/2	−13/2	−1/2	5

Analogi, izpildot nākošos izslēgšanas soļus, iegūstam

1	−1	$\boxed{x_3}$	x_4	(III)
x_1	1	4	1	1
x_2	0	1	1	2
* y_3	−5	$\boxed{-6}$	−1	3
y_4	−4	−2	2	4
σ	−9	−4	2	5

1	−1	$\boxed{x_4}$	(IV)
x_1	7/3	1/3	1
x_2	−5/6	5/6	2
x_3	5/6	1/6	3
* y_4	−7/3	$\boxed{7/3}$	4
σ	−17/3	8/3	5

1	−1	(V)
x_1	−2	1
x_2	0	2
x_3	1	3
x_4	−1	4
σ	−3	5

Vienādojumu sistēmas atrisinājums ir $x_1=-2$, $x_2=0$, $x_3=1$, $x_4=-1$.

Šī piemēra atrisināšanas gaitā katrā izslēgšanas solī tika atmesta iepriekšējās tabulas galvenā kolonna. Tādu metodi sauc par saīsināto Žordāna izslēgšanas metodi.

3.4.§. Gausa—Žordāna metode. Risinot piemērus 3.2.1°—3.2.3° ar modificēto Žordāna izslēgšanas metodi, vienādojumu sistēmas pierakstīšanai lieto kanonisko tabulu (3.2.3). Turpretim Gausa—

Žordāna metodē lieto pilno tabulu (3.2.2). Ja vienādojumu sistēmu ekvivalentos pārveidojumus pieraksta pilno tabulu formā, resp., vienādojumu sistēmu risina ar Gausa—Žordāna metodi, tad mainās 3.2.§ aprakstītā kārtula, pēc kuras doto tabulu atvieto ar jaunu, ekvivalentās vienādojumu sistēmas pieraksta tabulu. Izmaiņas skar gan tikai dotās tabulas galveno kolonnu. Dotās tabulas galvenā elementa vietā jaunajā tabulā jāieraksta skaitlis 1 un visi pārējie galvenās kolonnas elementi jāatvieto ar nullēm. Visus citus jaunās tabulas elementus iegūst pēc formulām

$$a'_{rj}=\frac{a_{rj}}{a_{rk}}\ (j=0, 1, \ldots, n), \tag{3.4.1}$$

$$a_{i0}=b_i\ (i=1, 2, \ldots, m),$$

$$a'_{ij}=a_{ij}-\frac{a_{ik}a_{rj}}{a_{rk}}\ (i\neq r), \tag{3.4.2}$$

kur a_{rk} ir galvenais elements. 3.3.§ aprakstītā kolonnu summu kontrole noderīga arī tad, ja vienādojumu sistēmu risina ar Gausa—Žordāna metodi. Lietojot šo metodi, papildu nezināmos (sākuma bāzes mainīgos) parasti apzīmē ar tiem pašiem burtiem, ar kuriem apzīmēti sākuma nezināmie, un visi nezināmie cits no cita atšķiras vienīgi ar to indeksiem. Salīdzināšanas nolūkā atkārtosim vienādojumu sistēmas (3.2.10) risinājumu, lietojot modificētās Žordāna izslēgšanas metodes vietā Gausa—Žordāna metodi.

	x_1	x_2	x_3	x_4	x_5	x_6	-1
	1	−1	−1	1	0	0	0
I	3	−2	−1	0	1	0	4
	$\boxed{-2}$	3	3	0	0	1	5
	1	−1	0	0	0	0	
	0	0,5	$\boxed{0,5}$	1	0	0,5	2,5
II	0	2,5	3,5	0	1	1,5	11,5
	1	−1,5	−1,5	0	0	−0,5	−2,5
	0	0,5	1,5	0	0	0,5	10,5
	0	1	1	2	0	1	5
III	0	$\boxed{-1}$	0	−7	1	−2	−6
	1	0	0	3	0	1	5
	0	−1	0	−3	0	−1	3
	0	0	1	−5	1	−1	−1
IV	0	1	0	7	−1	2	6
	1	0	0	3	0	1	5
	0	0	0	4	−1	1	9

Atrisinājums. $x_1=5$, $x_2=6$, $x_3=-1$.

4. nodaļa

SIMPLEKSA METODE LINEĀRAJĀ PROGRAMMĒŠANĀ

4.1.§. Uzdevuma normālforma. Atgādināsim, ka lineārās programmēšanas uzdevums ir normālformā, ja ir jāatrod maksimums lineārai funkcijai

$$z=c_1x_1+c_2x_2+\ldots+c_nx_n \tag{4.1.1}$$

pie ierobežojumiem

$$a_{i1}x_1+a_{i2}x_2+\ldots+a_{in}x_n\leqslant b_i \quad (i=1,2,\ldots,m), \tag{4.1.2}$$

$$x_j\geqslant 0 \quad (j=1,2,\ldots,n), \tag{4.1.3}$$

kur $b_i\geqslant 0$; c_j un a_{ij} ir doti reāli skaitļi.

Ievedot papildu nezināmos $y_i\geqslant 0$ $(i=1,2,\ldots,m)$, tāds uzdevums viegli pārveidojams lineāru algebrisku vienādojumu sistēmā

$$\left.\begin{array}{l} z-\ c_1x_1-\ c_2x_2-\ldots-\ c_nx_n=0, \\ y_1+a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n=b_1, \\ y_2+a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n=b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_m+a_{m1}x_1+a_{m2}x_2+\ldots+a_{mn}x_n=b_m. \end{array}\right\} \begin{array}{l} \text{(0-tā rinda)} \\ \text{(1. rinda)} \\ \text{(2. rinda)} \\ \\ (m\text{-tā rinda)} \end{array} \tag{4.1.4}$$

Te y_i $(i=1,2,\ldots,m)$ ir bāzes mainīgie, bet x_j $(j=1,2,\ldots,n)$ — nebāzes mainīgie. Acīm redzams bāzes atrisinājums šādā gadījumā ir atbalsta plāns $y_i=b_i$ $(i=1,2,\ldots,m)$ ar šim plānam atbilstošo mērķa funkcijas vērtību $z=0$.

Ja kaut viens koeficients sistēmas (4.1.4) nulltajā rindā ir negatīvs, bāzes atrisinājumu var uzlabot.

Kā piemēru iztirzāsim uzdevumu 1.2.1°, kura grafiskais atrisinājums dots zīmējumā 2.1. Šis uzdevums ir normālformā, un tas viegli pārveidojams lineāru algebrisku vienādojumu sistēmā

$$\left.\begin{array}{r} z-30x_1-40x_2=\ \ 0, \\ y_1+\ \ x_1+\ 7x_2=\ 84, \\ y_2+\ 2x_1+\ 5x_2=\ 69, \\ y_3+\ 5x_1+\ 5x_2=105, \\ y_4+\ 4x_1+0\cdot x_2=\ 72. \end{array}\right\} \begin{array}{l} \text{(0-tā rinda)} \\ \text{(1. rinda)} \\ \text{(2. rinda)} \\ \text{(3. rinda)} \\ \text{(4. rinda)} \end{array} \tag{4.1.5}$$

Vienādojumu sistēmā (4.1.5) papildu nezināmie $y_i\geqslant 0$ $(i=1,2,3,4)$ ir bāzes mainīgie, bet $x_1\geqslant 0$ un $x_2\geqslant 0$ ir nebāzes mainīgie. Ņemot $x_1=x_2=0$, rodas atbalsta plāns $y_1=84$, $y_2=69$, $y_3=105$, $y_4=72$, kuram atbilst mērķa funkcijas vērtība $z=0$. Tas ir viens no šī uzdevuma bāzes atrisinājumiem. Šim bāzes atrisinājumam ir ļoti vienkārša ekonomiskā interpretācija. Ja ražošana nav sākta, gatavās produkcijas vienību skaits ir nulle, $x_1=x_2=0$, ražošanas resursu atlikumi $y_1=84$, $y_2=69$, $y_3=105$, $y_4=72$ ir vienādi ar to krājumiem un ienākums no gatavās produkcijas realizācijas ir nulle, $z=0$. Dotais bāzes atrisinājums acīm redzami nav vienīgais, jo, izmantojot esošos resursus, var ražot jebkuru no diviem produkcijas veidiem B_1 vai B_2. Tā kā katra produkcijas B_1 vienība dod 30 rbļ., bet katra B_2 vienība 40 rbļ. ienākuma, tad, šķiet, ir dabiski ražot iespējami daudz produkcijas B_2 vienību.

Saskaņā ar uzdevuma nosacījumiem resurss A_1 ļauj ražot produktu B_2 ne vairāk kā $84:7=12$ vienību. Analogi resurss A_2 limitē produkta B_2 ražošanu uz $69:5=13{,}8$ vienībām, bet A_3 — uz $105:5=21$ vienību. Resurss A_4 produkta B_2 ražošanu neierobežo. Tā kā starp skaitļiem 12, 13, 8 un 21 mazākais ir 12, tad produktu B_2 var saražot ne vairāk kā 12 vienību. Resurss A_1 tātad ir «šaurā vieta» produkta B_2 ražošanā. Ražojot $x_2=12$ vienības produktu B_2, nav iespējams ražot nevienu produkta B_1 vienību un $x_1=0$. Ražošanas plānam $x_1=0$ un $x_2=12$ atbilstošais ienākums no gatavās produkcijas realizācijas ir $z=30\cdot 0+40\cdot 12=480$. Izpildot tādu ražošanas plānu, resursu A_1, A_2, A_3 un A_4 patēriņš ir attiecīgi $7\cdot 12=84$, $5\cdot 12=60$, $5\cdot 12=60$ un $0\cdot 12=0$ vienības. Patēriņiem atbilstošie resursu atlikumi tad ir $y_1=84-84=0$, $y_2=69-60=9$, $y_3=105-60=45$ un $y_4=72-0=72$. Tā esam ieguvuši jaunu, uzlabotu bāzes atrisinājumu

$$\begin{aligned} &x_1=0, \quad x_2=12, \quad y_3=45, \quad z=480. \\ &y_1=0, \quad y_2=9, \quad y_4=72, \end{aligned} \tag{4.1.6}$$

Šo jauno, uzlaboto bāzes atrisinājumu iegūst no iepriekšējā bāzes atrisinājuma, apmainot lomām iepriekšējo bāzes mainīgo y_1 ar nebāzes mainīgo x_2. To var izdarīt analītiski ar vienādojumu sistēmas (4.1.5) ekvivalentu pārveidojumu. Izsakot no sistēmas (4.1.5) pirmās rindas vienādojuma $x_2=12-\frac{1}{7}y_1-\frac{1}{7}x_1$ un ievietojot to visos pārējos šīs sistēmas vienādojumos, iegūstam jaunu vienādojumu sistēmu

$$\left.\begin{aligned} &z-30x_1-40\left(12-\frac{1}{7}y_1-\frac{1}{7}x_1\right)=0, \\ &x_2+\frac{1}{7}x_1+\frac{1}{7}y_1=12, \\ &y_2+x_1+5\left(12-\frac{1}{7}y_1-\frac{1}{7}x_1\right)=69, \\ &y_3+5x_1+5\left(12-\frac{1}{7}y_1-\frac{1}{7}x_1\right)=105, \\ &y_4+4x_1=72 \end{aligned}\right\}$$

jeb pēc elementāriem pārveidojumiem

$$
\left.\begin{aligned}
z-\frac{170}{7}x_1+\frac{40}{7}y_1&=480, && \text{(0-tā rinda)}\\
x_2+\frac{1}{7}x_1+\frac{1}{7}y_1&=12, && \text{(1. rinda)}\\
y_2+\frac{9}{7}x_1-\frac{5}{7}y_1&=9, && \text{(2. rinda)}\\
y_3+\frac{30}{7}x_1-\frac{5}{7}y_1&=45, && \text{(3. rinda)}\\
y_4+4x_1&=72. && \text{(4. rinda)}
\end{aligned}\right\} \quad (4.1.7)
$$

Ievietojot vienādojumu sistēmā (4.1.7) $x_1=y_1=0$, iegūstam uzlaboto bāzes atrisinājumu (4.1.6).

Katrs nebāzes mainīgā koeficients mērķa funkcijas vienādojumā rāda, par cik pozitīvām vai negatīvām vienībām izmainās mērķa funkcija, ja attiecīgo nebāzes mainīgo izmaina par vienu tā vienību. Mērķa funkcija palielinās, ja aug tāds nebāzes mainīgais, kam ir negatīvs koeficients. Tā samazinās, ja aug nebāzes mainīgais ar pozitīvu koeficientu.

Vienādojumu sistēma (4.1.7) sākas ar apslēptā veidā dotās mērķa funkcijas vienādojumu. Šajā vienādojumā pie nebāzes mainīgā x_1 ir negatīvs koeficients $(-170/7)$. Tas nozīmē, ka, palielinot x_1 par vienu tā vienību, mērķa funkcija z pieaugs par 170/7 tās vienībām (rubļiem). Tā kā visos pārējos vienādojumu sistēmas (4.1.7) vienādojumos pie x_1 ir pozitīvi koeficienti, tad x_1 palielināšana ir ierobežota. Palielināšanu limitē vienādojumu sistēmas brīvo locekļu attiecība pret mainīgā x_1 pozitīvajiem koeficientiem. Šīs attiecības ir $12:1/7=84$; $9:9/7=7$; $45:30/7=63/6$ un $72:4=18$. Mazākā no tām ir 7. Tas nozīmē to, ka tālākās plāna uzlabošanas «šaurā vieta» ir resursa A_2 atlikums un ka, ražojot 7 vienības produkcijas B_1, resursa A_2 atlikums kļūs vienāds ar 0, attiecīgi samazinoties arī produkcijas B_2 izlaidei un resursu A_3 un A_4 atlikumiem. Izsakot no sistēmas (4.1.7) otrās rindas $x_1=7-\frac{7}{9}y_2+$ $+\frac{5}{9}y_1$ un ievietojot to visos pārējos šīs sistēmas vienādojumos, iegūstam jaunu, ekvivalentu vienādojumu sistēmu

$$
\left.\begin{aligned}
z+\frac{170}{9}y_2-\frac{70}{9}y_1&=650, && \text{(0-tā rinda)}\\
x_2-\frac{1}{9}y_2+\frac{2}{9}y_1&=11, && \text{(1. rinda)}\\
x_1+\frac{7}{9}y_2-\frac{5}{9}y_1&=7, && \text{(2. rinda)}\\
y_3-\frac{10}{3}y_2+\frac{5}{3}y_1&=15, && \text{(3. rinda)}\\
y_4-\frac{28}{9}y_2+\frac{20}{9}y_1&=44. && \text{(4. rinda)}
\end{aligned}\right\} \quad (4.1.8)
$$

Ņemot $y_1=y_2=0$, iegūstam $x_1=7$, $x_2=11$, $y_3=15$, $y_4=44$ un $z=650$. Tas ir jauns, uzlabots bāzes atrisinājums. Ekonomiskā interpretācijā tas nozīmē, ka, ražojot 7 vienības produkta B_1 un 11 vienību produkta B_2, tiks pilnīgi izmantoti resursi A_1 un A_2, resursa A_3 atlikums būs 15 vienību un resursa A_4 atlikums — 44 vienības. Gatavās produkcijas realizācijas ienākums tad būs 650 rbļ.

Arī šis uzlabotais bāzes atrisinājums nav optimāls, jo pēdējās vienādojumu sistēmas nulltajā rindā pie nebāzes mainīgā y_1 ir negatīvs koeficients $(-70/9)$. Tas nozīmē, ka nav lietderīgi izmantot resursu A_1 pilnīgi. Palielinot šī resursa atlikumu y_1 par vienu vienību, par 70/9 rbļ. palielināsies realizācijas ienākums z. Nezināmā y_1 palielināšanu ierobežo attiecības $11:2/9=99/2$; $15:5/3=9$ un $44:20/9=99/5$. Mazākā no tām rāda, ka resursa A_1 atlikumu y_1 var palielināt ne vairāk kā par 9 vienībām, kad par «šauro vietu» kļūst resursa A_3 atlikums y_3.

Izsakot no sistēmas (4.1.8) trešās rindas $y_1=9+2y_2-\frac{3}{5}y_3$ un ievietojot to visos pārējos šīs sistēmas vienādojumos, iegūstam jaunu, ekvivalentu vienādojumu sistēmu

$$\left.\begin{aligned} z+\frac{10}{3}y_2+\frac{14}{3}y_3&=720, && \text{(0-tā rinda)}\\ x_2+\frac{1}{3}y_2-\frac{2}{15}y_3&=9, && \text{(1. rinda)}\\ x_1-\frac{1}{3}y_2+\frac{1}{3}y_3&=12, && \text{(2. rinda)}\\ y_1-2y_2+\frac{3}{5}y_3&=9, && \text{(3. rinda)}\\ y_4+\frac{4}{3}y_2-\frac{4}{3}y_3&=24. && \text{(4. rinda)} \end{aligned}\right\} \qquad (4.1.9)$$

Ņemot $y_2=y_3=0$, iegūstam $x_1=12$, $x_2=9$, $y_1=9$, $y_4=24$ un $z=720$. Tas ir jauns, uzlabots bāzes atrisinājums. Šis bāzes atrisinājums ir arī optimālais, jo vienādojumu sistēmas nulltajā rindā nav neviena negatīva koeficienta un resursu atlikumu y_2 vai y_3 palielināšana samazinās realizācijas ienākumu. Tādēļ $\max z=720$.

Šajā piemērā aprakstīto atbalsta plāna pakāpeniskas uzlabošanas un optimālā plāna atrašanas paņēmienu sauc par *simpleksa metodi* lineārās programmēšanas uzdevumu atrisināšanai.

Salīdzinot uzdevuma 1.2.1° grafisko atrisinājumu (zīm. 2.1) ar rezultātiem, kas iegūti ar simpleksa metodi, redzam, ka atbalsta plānos nezināmo x_1 un x_2 vērtības to pakāpeniskā secībā ir grafiskā risinājuma punktu koordinātes: $O(0;0)$, $A_5(0;12)$, $A_4(7;11)$ un $A_3(12;9)$. Tātad *atbalsta plānu elementi sakrīt ar mērķa funkcijas definīcijas apgabala virsotņu koordinātēm.*

Aplūkotais piemērs rāda, ka simpleksa metode būtībā ir dotās lineāru algebrisku vienādojumu sistēmas pakāpeniska pārveidošana tai ekvivalentā vienādojumu sistēmā, pielietojot soli pa solim modificēto Žordāna izslēgšanas metodi ar attiecīgi izraudzītu galveno elementu katrā izslēgšanas solī. Galvenā elementa noteikšana katram izslēgšanas solim notiek saskaņā ar simpleksa metodes algoritmu, kuru aplūkosim nākošajā paragrāfā.

4.2. §. Simpleksa metodes parastais algoritms. Jebkura lineārās programmēšanas uzdevuma atrisināšanā ar simpleksa metodi ir divi posmi. Pirmajā posmā atrod kaut kādu vienu bāzes atrisinājumu, ja tāds dotajam uzdevumam eksistē. Otrajā posmā atrod optimālo atrisinājumu, ja tāds dotajam uzdevumam eksistē. Otro posmu realizē, pakāpeniski uzlabojot pirmajā posmā atrasto bāzes atrisinājumu, jo ir pareiza šāda teorēma.

T e o r ē m a 1° (p a r b ā z i). *Ja lineārās programmēšanas uzdevumam ir galīgs optimāls atrisinājums, tad šādam uzdevumam eksistē optimāls bāzes atrisinājums.*

Pieņemsim, ka uzdevuma atrisināšanas pirmais posms ir realizēts un ir atrasts kāds atbalsta plāns ar tam atbilstošo mērķa funkcijas vērtību. Šis posms, kā redzējām iepriekšējā paragrāfā, viegli realizējams, ja uzdevums dots normālformā. Katrs normālformā dots uzdevums viegli pārveidojams kanoniskajā formā un izsakāms ar vienādojumu sistēmu (4.1.4). Vienādojumu sistēmu (4.1.4) var pierakstīt tabulas formā:

$+1$	-1	x_1	x_2	$\cdots$	$\boxed{x_k}$	$\cdots$	x_n	Nr.
z	0	$-c_1$	$-c_2$	$\cdots$	$-c_k$	$\cdots$	$-c_n$	0
y_1	b_1	a_{11}	a_{12}	$\cdots$	a_{1k}	$\cdots$	a_{1n}	1
y_2	b_2	a_{21}	a_{22}	$\cdots$	a_{2k}	$\cdots$	a_{2n}	2
.	.	.	.	...	.	...	.	.
.	.	.	.	...	.	...	.	.
.	.	.	.	...	.	...	.	.
* y_r	b_r	a_{r1}	a_{r2}	$\cdots$	$\boxed{a_{rk}}$	$\cdots$	a_{rn}	r
.	.	.	.	...	.	...	.	.
.	.	.	.	...	.	...	.	.
.	.	.	.	...	.	...	.	.
y_m	b_m	a_{m1}	a_{m2}	$\cdots$	a_{mk}	$\cdots$	a_{mn}	m
σ	σ_0-1	σ_1-1	σ_2-1	$\cdots$	σ_k-1	$\cdots$	σ_m-1	$m+1$

(4.2.0)

Tādu tabulu sauksim par kanonisko simpleksa metodes *iterāciju tabulu.* Tabulā sistēmas (4.1.4) vienādojumi lasāmi pa rindām, reizinot katrā rindā ierakstītos lielumus ar tabulas galvas rindā ierakstītajiem lielumiem, visus reizinājumus summējot un pielīdzinot iegūtās summas nullei. Tiešām, pārrakstot mērķa funkciju apslēptā formā, iegūstam izteiksmi

$$z-c_1x_1-c_2x_2-\ldots-c_nx_n=0,$$

kas ir sistēmas (4.1.4) nulltajā rindā ierakstītais vienādojums, bet

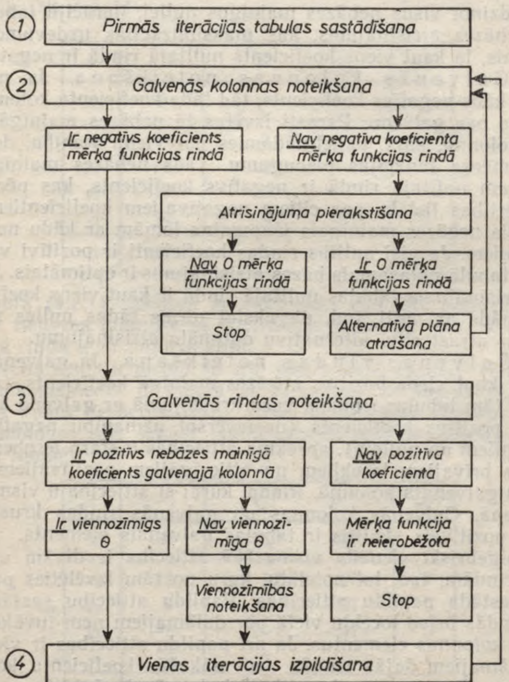
$$y_i - b_i + a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n = 0 \quad (i = 1, 2, \ldots, m)$$

ir sistēmas (4.1.4) i-tajā rindā ierakstītais vienādojums ar to atšķirību, ka brīvais loceklis no labās puses pārnests ar pretēju zīmi kreisajā pusē un ierakstīts kolonnā, kura novietota blakus bāzes mainīgo kolonnai.

Iterāciju tabulā visi bāzes mainīgie atrodas tabulas kreisajā malējā kolonnā, bet visi nebāzes mainīgie — tabulas galvas rindā.

Simpleksa metodes algoritms (maksimizācijas uzdevumam) shematiski attēlots zīmējumā 4.1. Algoritmam ir četri galvenie soļi.

Simpleksa metodes algoritma blokshēma



Zīm. 4.1.

1°. Pirmās iterācijas tabulas sastādīšana. Lai tādu tabulu sastādītu, tad dotais lineārās programmēšanas uzdevums jāpārveido par vienādojumu sistēmu ar nenegatīviem brīvajiem locekļiem (labajām pusēm) un nenegatīviem nezināmajiem, kas sadalīti bāzes un nebāzes mainīgajos. Viens no tādas sistēmas vienādojumiem apslēptā formā definē dotā lineārās programmēšanas uzdevuma mērķa funkciju. To pievieno sistēmai tās sākumā (nulltajā rindā) vai beigās. Pārējie sistēmas vienādojumi ir uzdevuma ierobežojumi, kuri pārveidoti par vienādojumiem ar savu bāzes mainīgo katrā vienādojumā. Katrs nulltās rindas koeficients, kam ir mīnusa zīme, rāda, par cik vienībām palielinās mērķa funkcija, ja attiecīgo nebāzes mainīgo palielina par vienu vienību. Katrs nulltās rindas koeficients, kam ir plusa zīme, rāda, par cik vienībām samazinās mērķa funkcija, ja attiecīgo nebāzes mainīgo palielina par vienu vienību.

Pielīdzinot visus nebāzes mainīgos nullei, iterāciju tabulā nolasāms bāzes atrisinājums, kas maksimizācijas uzdevumos nav optimālais, ja kaut viens koeficients nulltajā rindā ir negatīvs.

2°. Galvenās kolonnas noteikšana. Ja nulltajā rindā ir kāds negatīvs koeficients, tad tāda koeficienta kolonnu var izvēlēties par galveno. Parasti izvēlas tā nebāzes mainīgā koeficienta kolonnu, kurš, palielinādamies par vienu vienību, dod vislielāko mērķa funkcijas pieaugumu. Tāds nebāzes mainīgais ir tas, kuram nulltajā rindā ir negatīvs koeficients, kas pēc absolūtās vērtības lielāks par citiem negatīvajiem koeficientiem šajā rindā. Šis nebāzes mainīgais jāapmaina lomām ar kādu no bāzes mainīgajiem. Ja visi nulltās rindas koeficienti ir pozitīvi vai nulles, tad tabulā nolasāmais bāzes atrisinājums ir optimālais. Ja optimālā atrisinājuma tabulas nulltajā rindā ir kaut viens koeficients, kas vienāds ar nulli, tad, pierakstot pirms tādas nulles mīnusa zīmi, var atrast citu, alternatīvu optimālo atrisinājumu.

3°. Galvenās rindas noteikšana. Ja galvenajā kolonnā ir kaut viens pozitīvs nebāzes mainīgā koeficients, tad tām un tikai tām tabulas rindām, kuru krustojumā ar galveno kolonnu atrodas pozitīvs koeficients (nepievēršot uzmanību negatīvajiem koeficientiem un nullēm), aprēķina attiecības θ starp ierobežojumu sistēmas brīvajiem locekļiem un attiecīgajiem pozitīvajiem koeficientiem galvenajā kolonnā. Rinda, kurai šī attiecība θ vismazākā, ir galvenā. Galvenās kolonnas un galvenās rindas krustojumā esošais pozitīvais skaitlis ir tabulas galvenais elements.

Ja algebriski vienāda vismazākā attiecība ir divām vai vairākām rindām, tad, lai noteiktu, kuru no tām izvēlēties par galveno, sastāda papildu attiecības. Papildu attiecību sastādīšanai šajās rindās brīvo locekļu vietā par dalāmajiem ņem tuvākās koeficientu kolonnas elementus. Ja arī papildu attiecības ir vienādas, par dalāmajiem šajās rindās ņem nākošās koeficientu kolonnas elementus. Procesu turpina tik ilgi, kamēr iegūst vienu vismazāko papildu attiecību. Pēc tās nosaka galveno rindu.

Ja galvenajā kolonnā nav neviena pozitīva koeficienta, galvenā rinda nav jānosaka. Tādā gadījumā mērķa funkcija savā definīcijas apgabalā nav ierobežota.

4°. V i e n a s   i t e r ā c i j a s   i z p i l d ī š a n a. Izpildīt vienu iterāciju nozīmē izpildīt vienu modificētās Žordāna izslēgšanas metodes soli ar iepriekš aprakstītajā kārtībā noteiktu galveno elementu. Pēc vienas iterācijas izpildīšanas iegūst jaunu bāzes atrisinājumu. Ja tas nav optimālais, ar iegūto jauno iterācijas tabulu rīkojas tāpat kā ar iepriekšējo. Lineāru algebrisku vienādojumu sistēmas atrisināšanas gaitas kontrole ar kolonnu summām ir noderīga arī katrā iterācijā izpildīto aritmētisko darbību pareizības pārbaudei. Kā piemēru atkārtosim iepriekšējā paragrāfā iztirzāto uzdevuma 1.2.1°, t. i., sistēmas (4.1.5) risinājumu.

Sistēmu (4.1.5) varam pārrakstīt pirmās iterācijas tabulas formā ar rindu numerācijas kolonnu un kolonnu summu kontroles rindu (sk. 3.3. §):

	1	-1	x_1	$\boxed{x_2}$	(I)
	z	0	-30	-40	0
*	y_1	84	1	$\boxed{7}$	1
	y_2	69	2	5	2
	y_3	105	5	5	3
	y_4	72	4	0	4
	σ	329	-19	-24	5

(4.2.1)

Tabulā (4.2.1) saskaņā ar simpleksa metodes algoritmu galvenā ir nebāzes mainīgā x_2 koeficientu kolonna, jo x_2 koeficients nulltajā rindā ir ar mīnusa zīmi.

Galvenā rinda ir pirmā, jo min (84/7; 69/5; 105/5) = 84/7. Tātad galvenais elements ir tabulā ar taisnstūri iezīmētais skaitlis 7. Izpildot vienu iterāciju, resp., vienu modificētās Žordāna izslēgšanas metodes soli, iegūst jaunu tabulu, kurā mainīgie y_1 un x_2 mainīti lomām:

	1	-1	$\boxed{x_1}$	y_1	(II)
	z	480	$-170/7$	$40/7$	0
	x_2	12	$1/7$	$1/7$	1
*	y_2	9	$\boxed{9/7}$	$-5/7$	2
	y_3	45	$30/7$	$-5/7$	3
	y_4	72	4	0	4
	σ	617	$-109/7$	$24/7$	5

(4.2.2)

Šajā tabulā nolasāma vienādojumu sistēma (4.1.7) un uzlabotais bāzes atrisinājums (4.1.6).

Tabulas (4.2.2) galvenais elements ir 9/7, jo −170/7 ir vienīgais negatīvais šīs tabulas nulltās rindas koeficients un min (12 : 1/7; 9 : 9/7; 45 : 30/7; 72 : 4) = 9 : 9/7. Mainot lomām nebāzes mainīgo x_1 ar bāzes mainīgo y_2, iegūst nākošo iterācijas tabulu

1	−1	y_2	$\boxed{y_1}$	(III)
z	650	170/9	−70/9	0
x_2	11	−1/9	2/9	1
x_1	7	7/9	−5/9	2
* y_3	15	−10/3	$\boxed{5/3}$	3
y_4	44	−28/9	20/9	4
σ	726	109/9	−47/9	5

(4.2.3)

Tabulā (4.2.3) nolasāma vienādojumu sistēma (4.1.8) un bāzes atrisinājums $z=650$; $y_2=y_1=0$; $x_2=11$; $x_1=7$; $y_3=15$ un $y_4=44$. Šis atrisinājums nav optimālais, jo nulltajā rindā pie nebāzes mainīgā ir negatīvs koeficients −70/9. Tā kā min (11 : 2/9; 15 : 5/3; 44 : 20/9) = 15 : 5/3, tad tabulas (4.2.3) galvenais elements 5/3 atrodas trešās rindas un nebāzes mainīgā y_1 kolonnas krustojumā. Nākošajā iterācijā mainām lomām nebāzes mainīgo y_1 ar bāzes mainīgo y_3, iegūstot jaunu iterācijas tabulu

1	−1	y_2	y_3	(IV)
z	720	10/3	14/3	0
x_2	9	1/3	−2/15	1
x_1	12	−1/3	1/3	2
y_1	9	−2	3/5	3
y_4	24	4/3	−4/3	4
σ	773	5/3	47/15	5

(4.2.4)

Tabulā (4.2.4) nolasāma vienādojumu sistēma (4.1.9) un optimālais atrisinājums $z=720$; $y_2=y_3=0$; $x_1=12$; $x_2=9$; $y_1=9$; $y_4=24$.

Atrisinājuma optimalitāte pierādāma ļoti vienkārši. Tabulā (4.2.4) nolasāmā vienādojumu sistēma (4.1.9) ir ekvivalenta ar pirmajā iterācijas tabulā (4.2.1) nolasāmo vienādojumu sistēmu (4.1.5), jo sistēma (4.1.9) iegūta no sistēmas (4.1.5) ar elementāriem algebriskiem pārveidojumiem. Šīs sistēmas ārēji izskatās dažādas, bet tās ir viena un tā paša uzdevuma matemātiskais modelis. Ja dotais uzdevums pašā sākumā būtu formulēts ar vienādojumu sistēmu, kas nolasāma tabulā (4.2.4), tad mērķa funkcija būtu šāda:

$$z=720-\frac{10}{3}y_2-\frac{14}{3}y_3.$$

Tā kā $y_2 \geqslant 0$ un $y_3 \geqslant 0$, tad ar jebkuru tabulas (4.2.4) nebāzes mainīgā y_2 vai y_3 vērtību, kas lielāka par nulli, funkcijas z vērtība ir mazāka par 720. Tādēļ $\max z=720$, ja $y_2=y_3=0$, ko arī vajadzēja pierādīt.

4.3. §. Vispārīgā formā dotā lineārās programmēšanas uzdevuma atrisināšana ar simpleksa metodi. Simpleksa metodes pirmā iterācijas tabula ar dotā uzdevuma bāzes atrisinājumu, kā redzējām iepriekšējā paragrāfā, viegli konstruējama, izmantojot papildu nezināmos, ja uzdevums dots normālformā. Katrā citā formā dotam uzdevumam atbalsta plāna un tam atbilstošās mērķa funkcijas vērtības atrašana nav tik vienkārša. Atbalsta plāna konstruēšanai lieto dažādas metodes, arī simpleksa metodi, risinot ar to t. s. palīguzdevumu vai paplašināto uzdevumu.

Saskaņā ar to, kas teikts 2.2. §, jebkuru lineārās programmēšanas uzdevumu var pārveidot kanoniskajā formā, t. i., uzdevumu var formulēt šādi.

Atrast maksimumu lineārai funkcijai

$$z=\sum_{j=1}^{n} c_j x_j \tag{4.3.1}$$

pie ierobežojumiem

$$\sum_{j=1}^{n} a_{ij}x_j=b_i \quad (i=1, 2, \ldots, m) \tag{4.3.2}$$

un

$$x_j \geqslant 0 \quad (j=1, 2, \ldots, n). \tag{4.3.3}$$

Neko nezaudējot no vispārīguma, varam pieņemt, ka ierobežojumu sistēmā (4.3.2) ir $p \leqslant m$ vienādojumi, kuros atklāti nolasāmi bāzes mainīgie, un ka šie p vienādojumi atrodas sistēmas pirmajās p rindās, bet pēdējās $m-p$ rindās esošajos vienādojumos atklāti nolasāmu bāzes mainīgo nav. To vietā ievedīsim *mākslīgus nezināmos*

$$\overline{y}_i=b_i-\sum_{j=1}^{n} a_{ij}x_j. \tag{4.3.4}$$

Tad pēdējie $m-p$ vienādojumi uzrakstāmi šādi:

$$\overline{y}_i-b_i+\sum_{j=1}^{n} a_{ij}x_j=0 \quad (i=p+1, p+2, \ldots, m). \tag{4.3.5}$$

Mākslīgo nezināmo ievešana ierobežojumu sistēmā ir matemātisks paņēmiens, ar kuru izveido bāzi atbalsta plāna atrašanai pēc simpleksa metodes algoritma.

Doto ierobežojumu sistēmu (4.3.4) acīm redzami apmierina tādi un tikai tādi nezināmie x_j, kuriem atbilstošie visi šajā sistēmā ievestie mākslīgie nezināmie $\overline{y}_i \geqslant 0$ vienādi ar nulli. Visi $\overline{y}_i \geqslant 0$ savukārt ir vienādi ar nulli tad un tikai tad, ja to summa ir nulle. Summējot izteiksmes (4.3.4), iegūstam

$$\sum_{i=p+1}^{m} \overline{y}_i = s_o - \sum_{j=1}^{n} s_j x_j \geqslant 0, \tag{4.3.6}$$

kur

$$s_0 = \sum_{i=p+1}^{m} b_i, \tag{4.3.7}$$

$$s_j = \sum_{i=p+1}^{m} a_{ij} \quad (j=1, 2, \ldots, n). \tag{4.3.8}$$

Nulle tātad ir visu mākslīgo nezināmo summas (4.3.6) apakšējā robeža vai lineāras funkcijas

$$s = \sum_{j=1}^{n} s_j x_j - s_0 \leqslant 0 \tag{4.3.9}$$

augšējā robeža. Lineāro funkciju (4.3.9) sauksim par dotā uzdevuma (4.3.1)—(4.3.3) *sekundāro mērķa funkciju.*

Lai maksimizētu sekundāro mērķa funkciju (4.3.9) pie ierobežojumiem (4.3.2), (4.3.3) un $\overline{y}_i \geqslant 0$ $(i=p+1, p+2, \ldots, m)$, var izmantot simpleksa metodes algoritmu.

Dotajam uzdevumam (4.3.1), (4.3.2) eksistē bāzes atrisinājums tad un tikai tad, ja tā sekundārās mērķa funkcijas maksimums ir nulle. Ja sekundārās mērķa funkcijas maksimums ir mazāks par nulli, dotajam uzdevumam nav bāzes atrisinājuma, jo tā ierobežojumu sistēma ir nesaderīga.

Sekundāro mērķa funkciju ieved simpleksa metodes iterāciju tabulā atsevišķā rindā (M) blakus nulltajai rindai. Maksimizējot sekundāro mērķa funkciju ar simpleksa metodes algoritmu, katrā iterācijā vienlaicīgi pārveido ne vien ierobežojumu sistēmas koeficientus, bet arī nulltās rindas koeficientus.

Tajā iterācijas tabulā, kurā atklājas sekundārās mērķa funkcijas maksimums, kas ir vienāds ar nulli, nolasāms dotā uzdevuma (4.3.1)—(4.3.3) bāzes atrisinājums. Ja tas nav optimāls, iterācijas jāturpina saskaņā ar simpleksa metodes algoritmu, nepievēršot turpmāk uzmanību rindai M.

Tā kā nevienādība (4.3.9) nemainās, ja to reizina ar jebkuru pozitīvu skaitli, tad sekundārās mērķa funkcijas brīvo locekli s_0 un koeficientus s_j $(j=1, 2, \ldots, n)$ var reizināt ar vienu un to pašu pēc patikas izraudzītu lielu pozitīvu skaitli M, piemēram, ar desmitkārtīgu pēc absolūtās vērtības vislielāko primārās mērķa funkcijas koeficientu. Pēc tādas reizināšanas sekundāro un primāro mērķa funkciju var apvienot, tās algebriski summējot. Tā iegūst paplašinātā uzdevuma mērķa funkciju

$$Ms+z=M\sum_{j=1}^{n} s_j x_j - Ms_0 + \sum_{j=1}^{n} c_j x_j$$

jeb

$$Ms+z=\sum_{j=1}^{n}(Ms_j+c_j)x_j-Ms_0. \tag{4.3.10}$$

Ja dotajam uzdevumam ir kaut viens bāzes atrisinājums, tad simpleksa metodes algoritms automātiski noved pie tā, ka visi mākslīgie nezināmie pārvēršas par nullēm. Tie mākslīgie nezināmie, kuri pēc kādas iterācijas izpildīšanas nonāk nebāzes mainīgo lomā, turpmākajās iterācijās nav nepieciešami, un to koeficientu kolonnas turpmākajās iterācijās var atmest.

Piemērs 4.3.1°. Ar simpleksa metodi maksimizēt lineāru funkciju

$$z=x_1-x_2 \tag{4.3.11}$$

pie ierobežojumiem

$$\left.\begin{aligned} x_1+\ x_2&\leqslant 5,\\ 2x_1+\ x_2&\geqslant 6,\\ x_1+2x_2&=6,\\ x_1\geqslant 0;\ x_2&\geqslant 0.\end{aligned}\right\} \tag{4.3.12}$$

Atrisinājums. Ievedot nenegatīvus papildu nezināmos $y_1 \geqslant 0$ un $y_2 \geqslant 0$, ierobežojumu sistēmu (4.3.12) var pārrakstīt vienādojumu sistēmas formā:

$$\left.\begin{aligned} y_1+x_1+\ x_2\qquad &=5,\\ 2x_1+\ x_2-y_2&=6,\\ x_1+2x_2\qquad &=6.\end{aligned}\right\} \tag{4.3.13}$$

Tikai pirmajā sistēmas (4.3.13) vienādojumā ir atklāts bāzes mainīgais y_1. Pārējos divos vienādojumos tādu nav. Papildu nezināmais y_2 gan sastopams tikai otrajā vienādojumā, un mērķa funkcijā (4.3.11) tam ir koeficients 0, bet savā vienādojumā tam ir mīnusa zīme. Tādēļ tas nav bāzes mainīgais. Lai aizpildītu bāzes mainīgo kolonnu, ievedīsim divus mākslīgus nezināmos

$\bar{y}_2 \geqslant 0$ un $\bar{y}_3 \geqslant 0$ attiecīgi otrajā un trešajā sistēmas (4.3.13) vienādojumā. Tad rodas vienādojumu sistēma

$$\left.\begin{aligned} y_1 + x_1 + x_2 \quad &= 5, \\ \bar{y}_2 + 2x_1 + x_2 - y_2 &= 6, \\ \bar{y}_3 + x_1 + 2x_2 \quad &= 6. \end{aligned}\right\} \qquad (4.3.14)$$

Tā kā dotā uzdevuma ierobežojumu sistēmā ievesti mākslīgie nezināmie, tad pirmajā posmā jāatrisina palīguzdevums ar sekundāro mērķa funkciju. Saskaņā ar definīcijas formulu (4.3.9) sekundārā mērķa funkcija šajā gadījumā ir

$$s = 3x_1 + 3x_2 - y_2 - 12. \qquad (4.3.15)$$

Maksimizēsim sekundāro mērķa funkciju (4.3.15) pie ierobežojumiem (4.3.14), $y_1 \geqslant 0$, $y_2 \geqslant 0$, $\bar{y}_2 \geqslant 0$, $\bar{y}_3 \geqslant 0$, $x_1 \geqslant 0$ un $x_2 \geqslant 0$, vienlaicīgi pārveidosim doto mērķa funkciju, izslēdzot no tās katrā iterācijas solī jauno bāzes mainīgo, un atradīsim dotā uzdevuma bāzes atrisinājumu un pēc tam arī optimālo atrisinājumu. Šim nolūkam sastādīsim uzdevuma pirmo simpleksa metodes iterācijas tabulu ar sekundārās mērķa funkcijas rindu un kolonnu summu kontroles rindu:

	1	−1	$\boxed{x_1}$	x_2	y_2	(I)
	s	−12	−3	−3	1	M
	z	0	−1	1	0	0
	y_1	5	1	1	0	1
*	$\bar{y}_2$	6	$\boxed{2}$	1	−1	2
	$\bar{y}_3$	6	1	2	0	3
	σ	4	−1	1	1	4

(4.3.16)

Sekundārās mērķa funkcijas koeficientus pirmajā iterācijas tabulā parasti ieraksta pēc tam, kad aizpildītas visas pārējās šīs tabulas rindas. Sekundārās mērķa funkcijas brīvais loceklis un koeficienti šajā tabulā ir ar pretēju zīmi ņemtas to skaitļu summas, kuri atrodas attiecīgo kolonnu krustojumos ar 2. un 3. rindu, t. i., ar rindām, kurās atrodas mākslīgie nezināmie. Tiešām, $-(6+6) = -12$; $-(1+2) = -3$; $-(2+1) = -3$ un $-(0-1) = 1$.

Tabulas rindā M ir divi negatīvi nebāzes mainīgo koeficienti (-3). To absolūtā vērtība ir vienāda. Tādēļ, izpildot simpleksa metodes algoritma otro soli, jāizvēlas, kuru no divām kolonnām ņemt par galveno. Šādā gadījumā parasti rīkojas saskaņā ar formulu (4.3.10). Tā kā $|-3M-1| > |-3M+1|$, tad galvenā ir nezināmā x_1 koeficientu kolonna, un, tā kā $\min(5:1;\ 6:2;\ 6:1) = 6:2$, tad galvenā rinda ir otrā. Tādēļ galvenais elements tabulā (4.3.16) ir ar taisnstūri apvilktais skaitlis 2.

Izpildot vienu iterāciju, iegūstam nākošo iterācijas tabulu

1	-1	$\bar{y}_2$	$\boxed{x_2}$	y_2	(II)
s	-3	$3/2$	$-3/2$	$-1/2$	M
z	3	$1/2$	$3/2$	$-1/2$	0
y_1	2	$-1/2$	$1/2$	$1/2$	1
x_1	3	$1/2$	$1/2$	$-1/2$	2
* $\bar{y}_3$	3	$-1/2$	$\boxed{3/2}$	$1/2$	3
σ	7	$1/2$	$3/2$	$-3/2$	4

(4.3.17)

Turpmākajās iterācijās atmetam kolonnu, kurā atrodas mākslīgais nebāzes mainīgais $\bar{y}_2$. Nākošā iterācijas tabula ir šāda:

1	-1	$\bar{y}_3$	$\boxed{y_2}$	(III)
s	0	1	0	M
z	0	-1	-1	0
* y_1	1	$-1/3$	$\boxed{1/3}$	1
x_1	2	$-1/3$	$-2/3$	2
x_2	2	$2/3$	$1/3$	3
σ	4	-1	-2	4

(4.3.18)

No tabulas (4.3.18) redzams, ka funkcijas s maksimums ir nulle. Tādēļ dotajam uzdevumam ir bāzes atrisinājums $x_1=2$, $x_2=2$, $y_1=1$, $y_2=0$ un $z=0$. Atrastā funkcijas z vērtība nav maksimālā (bāzes atrisinājums nav optimālais), jo primārajā mērķa funkcijas vienādojumā (tabulas nulltajā rindā) koeficients pie nebāzes mainīgā y_2 ir negatīvs. Nākošā iterācijas tabula ir šāda:

1	-1	y_1	(IV)
s	0	0	M
z	3	3	0
y_2	3	3	1
x_1	4	2	2
x_2	1	-1	3
σ	10	6	4

(4.3.19)

Tabulā (4.3.19) nolasāms uzdevuma atrisinājums: $\max z=3$, kad $y_1=0$, $y_2=3$, $x_1=4$, $x_2=1$.

4.4. §. Simpleksa metodes sašaurināto iterāciju tabulu pieraksts. Pieņemsim, ka lineārās programmēšanas uzdevums vispārīgā veidā formulēts tā, kā uzdevums 2.1.1°. Tādu uzdevumu var vienkārši pierakstīt tabulas (4.4.0) formā. Bāzes mainīgo ailē šajā tabulā ar (+) vai (−) zīmēm ierakstīti papildu nezināmie tiem ierobežojumiem, kas uzdevumā doti nevienādību formā, un

$+1$	-1	x_1	$x_2 \cdots$	$\boxed{x_k}$	$\cdots x_n$	I
s	$-s_0$	$-s_1$	$-s_2 \cdots$	$-s_k$	$\cdots -s_n$	M
z	0	$-c_1$	$-c_2 \cdots$	$-c_k$	$\cdots -c_n$	0
y_1	b_1	a_{11}	$a_{12} \cdots$	a_{1k}	$\cdots a_{1n}$	1
y_2	b_2	a_{21}	$a_{22} \cdots$	a_{2k}	$\cdots a_{2n}$	2
.	.	.	. ...	.	... .	.
.	.	.	. ...	.	... .	.
.	.	.	. ...	.	... .	.
y_p	b_p	a_{p1}	$a_{p2} \cdots$	a_{pk}	$\cdots a_{pn}$	p
$-y_{p+1}$	b_{p+1}	$a_{p+1,1}$	$a_{p+1,2} \cdots$	$a_{p+1,k}$	$\cdots a_{p+1,n}$	$p+1$
$-y_{p+2}$	b_{p+2}	$a_{p+2,1}$	$a_{p+2,2} \cdots$	$a_{p+2,k}$	$\cdots a_{p+2,n}$	$p+2$
.	.	.	. ...	.	... .	.
.	.	.	. ...	.	... .	.
.	.	.	. ...	.	... .	.
$-y_h$	b_h	a_{h1}	$a_{h2} \cdots$	a_{hk}	$\cdots a_{hn}$	h
$\bar{y}_{h+1}$	b_{h+1}	$a_{h+1,1}$	$a_{h+1,2} \cdots$	$a_{h+1,k}$	$\cdots a_{h+1,n}$	$h+1$
$\bar{y}_{h+2}$	b_{h+2}	$a_{h+2,1}$	$a_{h+2,2} \cdots$	$a_{h+2,k}$	$\cdots a_{h+2,n}$	$h+2$
.	.	.	. ...	.	... .	.
.	.	.	. ...	.	... .	.
.	.	.	. ...	.	... .	.
$\bar{y}_{m-1}$	b_{m-1}	$a_{m-1,1}$	$a_{m-1,2} \cdots$	$a_{m-1,k}$	$\cdots a_{m-1,n}$	$m-1$
$\bar{y}_m$	b_m	a_{m1}	$a_{m2} \cdots$	a_{mk}	$\cdots a_{mn}$	m
σ	σ_0-1	σ_1-1	$\sigma_2-1 \cdots$	σ_k-1	$\cdots \sigma_n-1$	$m+1$

(4.4.0)

ar (+) zīmēm mākslīgie nezināmie tiem ierobežojumiem, kas doti vienādojumu formā. Jāievēro, ka sašaurinātajā iterāciju tabulā (4.4.0) nebāzes mainīgo skaitam jābūt lielākam par mākslīgo bāzes mainīgo un ar mīnusa zīmi pierakstīto bāzes mainīgo skaita summu, t. i., jābūt $n > (m-p)$.

Sekundārās mērķa funkcijas brīvo locekli s_0 un koeficientus s_j $(j=1, 2, \ldots, n)$, kas ar pretēju zīmi ierakstāmi tabulas rindā M, iegūst, summējot attiecīgās kolonnas elementus no visām tām rindām, kuru krustojumā ar bāzes mainīgo kolonnu atrodas mākslīgie nezināmie vai papildu nezināmie, kam ir mīnusa zīme. Pēc šādas sašaurinātās iterāciju tabulas sastādīšanas uzdevumu risina ar simpleksa metodes algoritmu iepriekšējā paragrāfā aprakstītajā veidā. Neliela atšķirība pastāv iterācijas soļa izpildē.

Šī atšķirība attiecas uz galvenās kolonnas elementu atvietošanu, sastādot jauno iterācijas tabulu.

Tādā gadījumā, ja galvenās rindas krustojumā ar bāzes mainīgo kolonnu atrodas papildu nezināmais, kam ir mīnusa zīme, galveno elementu jaunajā tabulā atvieto ar $-\gamma=-\frac{1}{a_{rk}}$, bet visus pārējos galvenās kolonnas elementus ar skaitļiem, kurus iegūst, ja galvenās kolonnas elementus reizina ar γ. Sekundārās mērķa funkcijas rindā esošajam elementam jaunajā tabulā bez tam jāpieskaita 1, bet no kolonnu summu kontroles rindā esošā elementa jāatņem 1.

P i e m ē r s 4.4.1°. Minimizēt lineāru funkciju

$$z=100-10x_1+x_2+41{,}5x_3+52x_4 \tag{4.4.1}$$

pie ierobežojumiem

$$\left.\begin{aligned} 3x_1-x_2-4x_3-x_4&\geqslant-1,\\ -3x_1+2x_2-2x_3+2x_4&\geqslant 9,\\ 2x_1-x_2-x_3-3x_4&=-2,\\ -3x_1+2x_2-3x_3&=7, \end{aligned}\right\} \tag{4.4.2}$$

$$x_1\geqslant 0,\ x_2\geqslant 0,\ x_3\geqslant 0,\ x_4\geqslant 0. \tag{4.4.3}$$

Atrisinājums. Ierobežojumu sistēmu (4.4.2) pārveidosim tādā tai ekvivalentā sistēmā, kurā nav neviena negatīva brīvā locekļa:

$$\left.\begin{aligned} -3x_1+x_2+4x_3+x_4&\leqslant 1,\\ -3x_1+2x_2-2x_3+2x_4&\geqslant 9,\\ -2x_1+x_2+x_3+3x_4&=2,\\ -3x_1+2x_2-3x_3&=7. \end{aligned}\right\} \tag{4.4.4}$$

Pēc tāda ierobežojumu sistēmas pārveidojuma sastādīsim simpleksa metodes pirmo iterācijas tabulu

(4.4.5)

	1	−1	x_1	$\boxed{x_2}$	x_3	x_4	(I)
	s	−18	8	−5	4	−5	M
	$\overline{z}$	−100	−10	1	41,5	52	0
*	y_1	1	−3	$\boxed{1}$	4	1	1
	$-y_2$	9	−3	2	−2	2	2
	$\overline{y_3}$	2	−2	1	1	3	3
	$\overline{y_4}$	7	−3	2	−3	0	4
	σ	−100	−14	1	44,5	52	5

Lietojot simpleksa metodes parasto algoritmu, iegūstam nākošās iterāciju tabulas

1	-1	$\boxed{x_1}$	y_1	x_3	x_4	(II)
s	-13	-7	5	24	0	M
$\bar{z}$	-101	-7	-1	$37{,}5$	51	0
x_2	1	-3	1	4	1	1
$-y_2$	7	3	-2	-10	0	2
* y_3	1	$\boxed{1}$	-1	-3	2	3
$\bar{y}_4$	5	3	-2	-11	-2	4
σ	-101	-11	-1	$40{,}5$	51	5

(4.4.6)

1	-1	$\bar{y}_3$	$\boxed{y_1}$	x_3	x_4	(III)
s	-6	7	-2	3	14	M
$\bar{z}$	-94	7	-8	$16{,}5$	65	0
x_2	4	3	-2	-5	7	1
$-y_2$	4	-3	1	-1	-6	2
x_1	1	1	-1	-3	2	3
* $\bar{y}_4$	2	-3	$\boxed{1}$	-2	-8	4
σ	-90	11	-12	$7{,}5$	73	5

(4.4.7)

1	-1	$\bar{y}_4$	x_3	$\boxed{x_4}$	(IV)
s	-2	2	-1	-2	M
$\bar{z}$	-78	8	$0{,}5$	1	0
x_2	8	2	-9	-9	1
* $-y_2$	2	-1	1	$\boxed{2}$	2
x_1	3	1	-5	-6	3
y_1	2	1	-2	-8	4
σ	-66	12	$-16{,}5$	-23	5

(4.4.8)

Tabulā (4.4.9) sekundārā mērķa funkcija sasniedz maksimumu, kas vienāds ar 0. Tādēļ šajā tabulā nolasāms bāzes atrisinājums $\bar{z}=-79$, resp., $z=79$, ja $x_1=9$, $x_2=17$, $x_3=0$, $x_4=1$, $y_1=10$, $y_2=0$. Šis atrisinājums ir arī optimālais, jo nulltajā rindā nav neviena negatīva koeficienta.

+1	−1	$\boxed{x_3}$	y_2	(V)
s	0	0	0	M
$\bar{z}$	−79	0	0,5	0
x_2	17	−4,5	−4,5	1
* x_4	1	$\boxed{0,5}$	−0,5	2
x_1	9	−2,0	−3,0	3
y_1	10	2,0	−4,0	4
σ	−43	−5,0	−12,5	5

(4.4.9)

+1	−1	x_4	y_2	(V′)
$\bar{z}$	−79	0	0,5	0
x_2	26	9	−9	1
x_3	2	2	−1	2
x_1	13	4	−5	3
y_1	6	−4	−2	4
σ	−33	10	−17,5	5

(4.4.10)

Minētais optimālais atrisinājums šajā piemērā nav vienīgais, jo tabulas (4.4.9) nulltajā rindā pie nebāzes mainīgā x_3 ir koeficients 0. Uzskatot šo nulli par negatīvu lielumu, parastajā kārtībā iegūst nākošo iterācijas tabulu (4.4.10), kurā nolasāms cits, alternatīvs optimālais bāzes atrisinājums $z=+79$, ja $x_1=13$, $x_2=26$, $x_3=2$, $x_4=0$, $y_1=6$, $y_2=0$.

Tā kā šim piemēram ir divi alternatīvi optimālie bāzes atrisinājumi, tad optimālo atrisinājumu skaits šajā piemērā ir bezgalīgi liels. Mērķa funkcija sasniedz savu minimumu +79, ja

$$\left.\begin{aligned}
x_1 &= 9\lambda+13(1-\lambda),\\
x_2 &= 17\lambda+26(1-\lambda),\\
x_3 &= 2(1-\lambda),\\
x_4 &= \lambda,\\
y_1 &= 10\lambda+6(1-\lambda),\\
y_2 &= 0,
\end{aligned}\right\}$$

kur λ ir jebkurš skaitlis no intervāla $0\leqslant\lambda\leqslant1$.

Piemērs 4.4.2°. Maksimizēt lineāru funkciju

$$z = 2x_1 - x_2 + 5x_3 \tag{4.4.11}$$

pie ierobežojumiem

$$\left.\begin{aligned} &x_1 + x_2 \leqslant 8, \\ &x_2 + x_3 \leqslant 6, \\ &x_1 + x_2 + x_3 = 15, \\ &x_1 + x_3 \geqslant 10, \end{aligned}\right\} \tag{4.4.12}$$

$$x_1 \geqslant 0,\ x_2 \geqslant 0. \tag{4.4.13}$$

Atrisinājums. Tā kā ierobežojumu sistēmā (4.4.12), (4.4.13) nezināmajam x_3 nav zīmes ierobežojuma, tad izdarīsim mērķa funkcijā un ierobežojumu sistēmā substitūciju $x_3 = x'_3 - x''_3$, kur $x'_3 \geqslant 0$ un $x''_3 \geqslant 0$. Pēc tam sastādīsim pirmo sašaurināto iterācijas tabulu un risināsim doto uzdevumu saskaņā ar simpleksa metodes algoritmu [sk. tabulas (4.4.14)—(4.4.17)].

Saskaņā ar tabulu (4.4.17) sekundārās mērķa funkcijas maksimums ir -1. Tā kā šis maksimums nav nulle, tad dotā uzdevuma ierobežojumu sistēma ir nesaderīga.

	+1	−1	x_1	x_2	$\boxed{x'_3}$	x''_3	(I)	
	s	−25	−2	−1	−2	2	M	
	z	0	−2	1	−5	5	0	
	y_1	8	1	1	0	0	1	
*	y_2	6	0	1	$\boxed{1}$	−1	2	(4.4.14)
	$\bar{y}_3$	15	1	1	1	−1	3	
	$-\bar{y}_4$	10	1	0	1	−1	4	
	σ	13	−2	2	−5	3	5	

	+1	−1	$\boxed{x_1}$	x_2	y_2	x''_3	(II)	
	s	−13	−2	1	2	0	M	
	z	30	−2	6	5	0	0	
	y_1	8	1	1	0	0	1	
	x'_3	6	0	1	1	−1	2	(4.4.15)
	$\bar{y}_3$	9	1	0	−1	0	3	
*	$-y_4$	4	$\boxed{1}$	−1	−1	0	4	
	σ	43	−2	7	5	−2	5	

+1	−1	$\boxed{y_4}$	x_2	y_2	x''_3	(III)
s	−5	−1	−1	0	0	M
z	38	−2	4	3	0	0
* y_1	4	$\boxed{1}$	2	1	0	1
x'_3	6	0	1	1	−1	2
$\overline{y}_3$	5	1	1	0	0	3
x_1	4	−1	−1	−1	0	4
σ	51	−3	5	3	−2	5

(4.4.16)

1	−1	y_1	x_2	y_2	x''_3	(IV)
s	−1	1	1	1	0	M
z	46	2	8	5	0	0
y_4	4	1	2	1	0	1
x'_3	6	0	1	1	−1	2
$\overline{y}_3$	1	−1	−1	−1	0	3
x_1	8	1	1	0	0	4
σ	63	3	11	6	−2	5

(4.4.17)

P i e m ē r s 4.4.3°. Maksimizēt lineāru funkciju

$$z = x_1 + x_2 + x_3 \tag{4.4.18}$$

pie ierobežojumiem

$$\left.\begin{aligned} 2x_1+2x_2+\ x_3 &\geqslant 2,\\ 3x_1-3x_2+2x_3 &\leqslant 6,\\ 3x_1-3x_2-2x_3 &\leqslant 6,\\ 2x_2-2x_3 &\leqslant 2, \end{aligned}\right\} \tag{4.4.19}$$

$$x_1 \geqslant 0,\ x_2 \geqslant 0,\ x_3 \geqslant 0. \tag{4.4.20}$$

Atrisinājums.

+1	−1	$\boxed{x_1}$	x_2	x_3	(I)
s	−2	−2	−2	−1	M
z	0	−1	−1	−1	0
* $-y_1$	2	$\boxed{2}$	2	1	1
y_2	6	3	−3	2	2
y_3	6	3	−3	−2	3
y_4	2	0	2	−2	4
σ	13	4	−6	−4	5

(4.4.21)

+1	−1	y_1	x_2	$\boxed{x_3}$	(II)
s	0	0	0	0	M
z	1	$-1/2$	0	$-1/2$	0
* x_1	1	$-1/2$	1	$\boxed{1/2}$	1
y_2	3	$3/2$	-6	$1/2$	2
y_3	3	$3/2$	-6	$-7/2$	3
y_4	2	0	2	-2	4
σ	9	1	-10	-6	5

(4.4.22)

+1	−1	$\boxed{y_1}$	x_2	x_1	(III)
z	2	-1	1	1	0
x_3	2	-1	2	2	1
* y_2	2	$\boxed{2}$	-7	-1	2
y_3	10	-2	1	7	3
y_4	6	-2	6	4	4
σ	21	-5	2	12	5

(4.4.23)

+1	−1	y_2	x_2	x_1	(IV)
z	3	$1/2$	$-5/2$	$1/2$	0
x_3	3	$1/2$	$-3/2$	$3/2$	1
y_1	1	$1/2$	$-7/2$	$-1/2$	2
y_3	12	1	-6	6	3
y_4	8	1	-1	3	4
σ	26	$5/2$	$-39/2$	$19/2$	5

(4.4.24)

Vienīgais negatīvais koeficients tabulas (4.4.24) nulltajā rindā ir $-5/2$. Tas atrodas nebāzes mainīgā x_2 koeficientu kolonnā, kurā visi skaitļi ir negatīvi. Tas nozīmē, ka mērķa funkcija savā definīcijas apgabalā šajā piemērā nav ierobežota un tā var augt bezgalīgi.

4.5. §. Simpleksa algoritma monotonitāte, singularitāte un galīgums. Pieņemsim, ka jāatrisina lineārās programmēšanas maksimizācijas uzdevums, un apzīmēsim mērķa funkcijas vērtību bāzes atrisinājumā ar A. Nākošajā simpleksa metodes iterācijas tabulā šo skaitli A atvieto ar jaunu skaitli A'. To aprēķina pēc formulas

$$A' = A - \frac{b_r c_k}{a_{rk}}, \qquad (4.5.1)$$

kur $b_r \geqslant 0$ ir galvenās rindas brīvais loceklis, $c_k < 0$ — galvenajā kolonnā esošais mērķa funkcijas koeficients un $a_{rk}>0$ — galvenais elements.

Ja $b_r \neq 0$, tad saskaņā ar formulu (4.5.1) $A'>A$. Tas nozīmē, ka tajā gadījumā, kad $b_r \neq 0$, mērķa funkcijas vērtība monotomi aug, t. i., katrā nākošajā iterācijas tabulā mērķa funkcijas vērtība ir lielāka par iepriekšējo. Ja $b_r=0$, tad $A'=A$. Tādu atbalsta plānu, kurā kāds no bāzes mainīgajiem ir vienāds ar nulli, sauc par *singulāru*. Singularitāte var izpausties jau pirmajā iterācijas tabulā. Tā parādās kārtējā iterācijā visos tādos gadījumos, kad, izpildot simpleksa algoritma 3. soli, jāsastopas ar divām vai vairākām vienādām vismazākajām attiecībām, pēc kurām nosaka galveno rindu. Ja tāda rinda, kuras brīvais loceklis ir 0, kļūst par galveno, tad kārtējā iterācijā mērķa funkcija pieaugumu neiegūst. Singularitātei atkārtojoties, vairākās iterācijās pēc kārtas ir iespējams tāds gadījums, ka notiek atgriešanās pie jau lietotās bāzes. Šādu gadījumu simpleksa algoritmā sauc par *saciklošanos*. Saciklošanās nenotiek, ja pielieto singularitātes gadījumam paredzēto galvenās rindas noteikšanas paņēmienu, kas aprakstīts 4.2. §. Ja saciklošanās ir izslēgta, tad simpleksa algoritmam ir galīgs iterāciju skaits, jo iespējamo atbalsta plānu skaits vienmēr ir galīgs lielums. Tas nepārsniedz skaitli, kuru var aprēķināt ar formulu

$$C_n^m=\frac{n!}{m!(n-m)!},$$

kur n — nezināmo skaits, bet m — ierobežojumu vienādojumu skaits. Ar simboliem $n!$, $m!$ un $(n-m)!$ apzīmēti pirmo n, m un $(n-m)$ dabisko skaitļu reizinājumi. Tā, piemēram, $5!=1\cdot2\cdot3\cdot4\cdot5=120$. Tātad simpleksa algoritms noslēdzas pēc galīga soļu skaita, resp., šis algoritms ir galīgs.

Pieredze liecina, ka praktisku uzdevumu atrisināšanai vajadzīgais iterāciju skaits svārstās no $1{,}5m$ līdz $3m$.

5. nodaļa

DUALITĀTE LINEĀRAJĀ PROGRAMMĒŠANĀ

5.1. §. Duālie uzdevumi. Lineārās programmēšanas teorijā un praktiskajos pielietojumos ievērojama nozīme ir dualitātes principam. Izmantojot vienu un to pašu sākotnējo informāciju, var formulēt divus dažādus lineārās programmēšanas uzdevumus. Vienu no tiem sauc par tiešo, otru par duālo uzdevumu. No matemātiskā viedokļa tiešais un duālais uzdevums ir līdzvērtīgi.

Neatkarīgi no duālā uzdevuma ekonomiskās interpretācijas tā atrisinājums, kuru, kā vēlāk redzēsim, iegūst vienlaicīgi ar tiešā uzdevuma atrisinājumu, satur daudz tādas informācijas, kas nepieciešama tiešā uzdevuma matemātiskā modeļa jutīguma analīzei.

Modeļa jutīguma analīze, kuru aplūkosim 5.3. §, ir saistīta ar atbilžu meklēšanu, piemēram, uz šādiem jautājumiem:

1) vai atrastais optimālais plāns paliek nemainīgs, ja izmaina kādu koeficientu tiešā uzdevuma sākotnējā matemātiskajā modelī;

2) kādas sekas ir ierobežojumu sistēmas vairāku vai pat visu brīvo locekļu izmaiņai tiešā uzdevuma sākotnējā matemātiskajā modelī u. c.

Dualitātes jēdziena noskaidrošanai aplūkosim uzdevumu 1.2.1°, kura tiešais matemātiskais modelis formulēts ar izteiksmēm (1.2.2)—(1.2.4). Neatkārtojot šī tiešā uzdevuma ekonomisko un ģeometrisko interpretāciju, kā arī tā atrisinājuma aprakstu, kas dots 4.1. § un tabulās (4.2.1)—(4.2.4), formulēsim duālā uzdevuma matemātisko modeli.

Pieņemsim, ka u_1 ir ražošanas resursa A_1 vienības cena, bet u_2, u_3 un u_4 — attiecīgi resursu A_2, A_3 un A_4 vienību cenas. Tā kā cena kā resursa vienības vērtības izteiksme naudā nevar būt negatīva, tad

$$u_1 \geqslant 0, \; u_2 \geqslant 0, \; u_3 \geqslant 0, \; u_4 \geqslant 0. \qquad (5.1.1)$$

Ja, nosakot resursu vienības aprēķina cenas, ievēro noteikumu, ka katras produkcijas vienības ražošanai izmantoto resursu vērtība nevar būt mazāka par šīs vienības realizācijas cenu, tad jāizpildās nevienādībām

$$\left.\begin{array}{l} u_1+2u_2+5u_3+4u_4 \geqslant 30, \\ 7u_1+5u_2+5u_3 \qquad\quad\; \geqslant 40. \end{array}\right\} \qquad (5.1.2)$$

Visu ražošanas resursu kopējo vērtību izteic lineāra funkcija

$$w=84u_1+69u_2+105u_3+72u_4. \qquad (5.1.3)$$

Uzdevuma 1.2.1° duālo uzdevumu tagad var formulēt šādi. Noteikt ražošanas resursiem tādas vienību cenas, lai resursu kopvērtība būtu minimālā, bet katras produkcijas vienības ražošanā izmantoto resursu vērtība nebūtu mazāka par attiecīgo produkcijas vienības realizācijas cenu.

Atrisināt tādu uzdevumu matemātiski nozīmē atrast minimumu lineārajai funkcijai (5.1.3) pie ierobežojumiem (5.1.1), (5.1.2).

Šādu uzdevumu, kā katru lineārās programmēšanas uzdevumu, var risināt ar simpleksa metodi pilnīgi neatkarīgi no tiešā uzdevuma 1.2.1° un pilnīgi neatkarīgi no tiešajā uzdevumā lietotajiem nezināmo apzīmējumiem vai to ekonomiskās interpretācijas.

Apzīmējot diferences starp produkcijas vienības vērtību aprēķina cenās un šīs vienības realizācijas cenu ar $v_1 \geqslant 0$ un $v_2 \geqslant 0$, ierobežojumu sistēmu (5.1.2) var pārrakstīt šādi:

$$\left.\begin{array}{l} -v_1-30+\;u_1+2u_2+5u_3+4u_4=0, \\ -v_2-40+7u_1+5u_2+5u_3 \qquad\;\; =0. \end{array}\right\}$$

Minimizācijas uzdevuma vietā var maksimizēt mērķa funkciju

$$\overline{w}=-84u_1-69u_2-105u_3-72u_4.$$

Sastādot pirmo iterācijas tabulu sašaurinātajā pierakstā un risinot doto duālo uzdevumu ar simpleksa metodes algoritmu pazīstamajā veidā, optimālo atrisinājumu iegūst pēc 3. iterāciju soļa. Tas redzams tabulās (5.1.4)—(5.1.7).

	+1	−1	u_1	u_2	$\boxed{u_3}$	u_4	(I)
	s	−70	−8	−7	−10	−4	M
	$\overline{w}$	0	84	69	105	72	0
*	$-v_1$	30	1	2	$\boxed{5}$	4	1
	$-v_2$	40	7	5	5	0	2
	σ	−1	83	68	104	71	3

(5.1.4)

	+1	−1	$\boxed{u_1}$	u_2	v_1	u_4	(II)
	s	−10	−6	−3	−1	4	M
	$\overline{w}$	−630	63	27	21	−12	0
	u_3	6	1/5	2/5	−1/5	4/5	1
*	$-v_2$	10	$\boxed{6}$	3	1	−4	2
	σ	−625	311/5	132/5	99/5	−61/5	3

(5.1.5)

	+1	−1	v_2	$\boxed{u_2}$	v_1	u_4	(III)
	s	0	0	0	0	0	M
	$\overline{w}$	−735	21/2	−9/2	21/2	30	0
	u_3	17/3	1/30	3/10	−7/30	14/15	1
*	u_1	5/3	−1/6	$\boxed{1/2}$	1/6	−2/3	2
	σ	−2186/3	281/30	−47/10	283/30	439/15	3

(5.1.6)

+1	−1	v_2	u_1	v_1	u_4	(IV)
$\overline{w}$	−720	9	9	12	24	0
u_3	14/3	2/15	−3/5	−1/3	4/3	1
u_2	10/3	−1/3	2	1/3	−4/3	2
σ	−713	39/5	47/5	11	23	3

(5.1.7)

Tabulā (5.1.7) nolasāms duālā uzdevuma vienīgais optimālais bāzes atrisinājums $\max \overline{w} = -720$, ja $v_2 = u_1 = v_1 = u_4 = 0$ un $u_3 = 14/3$, $u_2 = 10/3$. Tā kā $\min w = -\max \overline{w}$, tad, apmainot visu tabulas (5.1.7) nulltās rindas elementu zīmes, pārkārtojot rindu un kolonnu numerāciju un atmetot skaitļošanas kontrolei lietoto 3. rindu, duālā uzdevuma optimālā atrisinājuma tabulu (5.1.7) var pārrakstīt šādi:

(5.1.8)

	0	1	2	3	4	
$+1$	-1	v_1	v_2	u_1	u_4	(IV′)
w	720	-12	-9	-9	-24	0
u_2	10/3	1/3	$-1/3$	2	$-4/3$	1
u_3	14/3	$-1/3$	2/15	$-3/5$	4/3	2

Salīdzināšanas nolūkā pārrakstīsim arī tiešā uzdevuma 1.2.1° optimālā atrisinājuma tabulu (4.2.4):

(5.1.9)

	0	1	2	
$+1$	-1	y_2	y_3	(IV)
z	720	10/3	14/3	0
x_1	12	$-1/3$	1/3	1
x_2	9	1/3	$-2/15$	2
y_1	9	-2	3/5	3
y_4	24	4/3	$-4/3$	4

Salīdzinot tabulas (5.1.9) un (5.1.8), redzams, ka tiešā uzdevuma mērķa funkcijas maksimālā vērtība ir vienāda ar duālā uzdevuma mērķa funkcijas minimālo vērtību:

$$\max z = \min w.$$

Duālā uzdevuma optimālā atrisinājuma tabulu iegūst no tiešā uzdevuma atrisinājuma tabulas — un otrādi, ja 1) samaina tabulas rindas ar kolonnām un 2) visus tabulas elementus reizina ar -1. Tas nozīmē, ka nekad nav vajadzības risināt atsevišķi tiešo un duālo uzdevumu. Ja ir atrisināts viens no šiem uzdevumiem, tad no tā atrisinājuma tabulas viegli nolasāms vai iegūstams otra uzdevuma atrisinājums.

Abu savstarpēji duālo lineārās programmēšanas uzdevumu ierobežojumu sistēmas var apvienot vienā simpleksa metodes iterāciju tabulā, kuru sauksim par duālo iterāciju tabulu. Tam nolūkam tiešā uzdevuma bāzes mainīgo kolonnai blakus jāpievieno duālā uzdevuma nebāzes mainīgo kolonna un tiešā uzdevuma nebāzes mainīgo rindai blakus — duālā uzdevuma bāzes mainīgo rinda. Ja vienu no šiem uzdevumiem lasa pa tabulas rindām, tad otrs jālasa pa kolonnām.

Tā, piemēram, uzdevuma 1.2.1° un tā duālā uzdevuma pirmās iterāciju tabulas var apvienot un pierakstīt šādā duālā iterāciju tabulā:

-1		w	v_1	v_2
	$+1$	-1	x_1	x_2
$+1$	z	0	-30	-40
u_1	y_1	84	1	7
u_2	y_2	69	2	5
u_3	y_3	105	5	5
u_4	y_4	72	4	0

Šā uzdevumu pāra pēdējā duālā iterāciju tabula ir šāda:

-1		w	u_2	u_3
	$+1$	-1	y_2	y_3
$+1$	z	720	10/3	14/3
v_1	x_1	12	$-1/3$	1/3
v_2	x_2	9	1/3	$-2/15$
u_1	y_1	9	-2	3/5
u_4	y_4	24	4/3	$-4/3$

(5.1.10)

Tabulā (5.1.10) pa rindām nolasāma tiešajam uzdevumam atbilstošā vienādojumu sistēma

$$\left.\begin{aligned} z-720+\frac{10}{3}y_2+\frac{14}{3}y_3&=0,\\ x_1-\ 12-\frac{1}{3}\,y_2+\frac{1}{3}\,y_3&=0,\\ x_2-\ \ 9+\frac{1}{3}\,y_2-\frac{2}{15}y_3&=0,\\ y_1-\ \ 9-\ 2y_2+\ \frac{3}{5}\,y_3&=0,\\ y_4-\ 24+\frac{4}{3}\,y_2-\frac{4}{3}\,y_3&=0.\end{aligned}\right\}$$

Ievietojot $y_2=y_3=0$, iegūstam $z=720$, $x_1=12$, $x_2=9$, $y_1=9$, $y_4=24$.

Pa kolonnām nolasāma duālajam uzdevumam atbilstošā vienādojumu sistēma

$$\left.\begin{aligned} & -w+720+12v_1+9v_2+9u_1+24u_4=0, \\ & -u_2+\frac{10}{3}-\frac{1}{3}v_1+\frac{1}{3}v_2-2u_1+\frac{4}{3}u_4=0, \\ & -u_3+\frac{14}{3}+\frac{1}{3}v_1-\frac{2}{15}v_2+\frac{3}{5}u_1-\frac{4}{3}u_4=0. \end{aligned}\right\}$$

Ievietojot $v_1=v_2=u_1=u_4=0$, iegūstam $w=720$, $u_2=10/3$, $u_3=14/3$.

Vispār jebkuru tiešo lineārās programmēšanas uzdevumu, kā redzējām, var formulēt tā, kā formulēts uzdevums 2.1.1°, t. i., šādi.

Maksimizēt lineāru funkciju

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

pie ierobežojumiem

$$\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i \quad (i=1,2,\ldots,p),$$

$$\sum_{j=1}^{n} a_{ij}x_j \geqslant b_i \quad (i=p+1,p+2,\ldots,h),$$

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \quad (i=h+1,h+2,\ldots,m),$$

$$x_j \geqslant 0 \quad (j=1,2,\ldots,n),$$

kur visi $b_i \geqslant 0$, c_j un a_{ij} ir doti reāli skaitļi, bet x_j — nezināmi reāli mainīgi lielumi.

Tādu uzdevumu vienmēr var pierakstīt simpleksa metodes iterāciju tabulas (4.4.0) formā. Šādam tiešajam uzdevumam atbilstošais duālais uzdevums viegli nolasāms pēc tabulas (4.4.0) un formulējams šādi.

Minimizēt lineāru funkciju

$$w=\sum_{i=1}^{p} b_i u_i-\sum_{i=p+1}^{h} b_i u_i+\sum_{i=h+1}^{m} b_i u_i$$

pie ierobežojumiem

$$\sum_{i=1}^{p} a_{ij}u_i-\sum_{i=p+1}^{h} a_{ij}u_i+\sum_{i=h+1}^{m} a_{ij}u_i \geqslant c_j$$

$$(j=1,2,\ldots,n),$$

$$u_i \geqslant 0 \quad (i=1,2,\ldots,h),$$

u_i $(i=h+1,h+2,\ldots,m)$ — bez zīmes ierobežojuma.

Jāpiezīmē, ka duālā uzdevuma formulēšanai nav nepieciešams tiešo uzdevumu iepriekš pārveidot tā, lai visi tā nezināmie un ierobežojumu sistēmas brīvie locekļi būtu nenegatīvi, jo starp katru tiešo un tam atbilstošo duālo uzdevumu pastāv savstarpēja atbilstība, kas parādīta shēmā (5.1.11).

(5.1.11)

Tiešais uzdevums	Duālais uzdevums
Maksimizācija Nezināmie x_j Mērķa funkcijas koeficienti c_j Ierobežojumu sistēmas brīvie locekļi b_i Ierobežojumu sistēmas koeficienti a_{ij}, $i=1, 2, \ldots, m$; $j=1, 2, \ldots, n$	Minimizācija Nezināmie u_i Ierobežojumu sistēmas brīvie locekļi c_j Mērķa funkcijas koeficienti b_i Ierobežojumu sistēmas koeficienti a_{ji}, $j=1, 2, \ldots, n$; $i=1, 2, \ldots, m$
Ierobežojums $\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i$	$u_i \geqslant 0$
Ierobežojums $\sum_{j=1}^{n} a_{ij}x_j \geqslant b_i$	$u_i \leqslant 0$
Ierobežojums $\sum_{j=1}^{n} a_{ij}x_j = b_i$	u_i — bez zīmes ierobežojuma
$x_j \geqslant 0$	Ierobežojums $\sum_{i=1}^{m} a_{ji}u_i \geqslant c_j$
$x_j \leqslant 0$	Ierobežojums $\sum_{i=1}^{m} a_{ji}u_i \leqslant c_j$
x_j — bez zīmes ierobežojuma	Ierobežojums $\sum_{i=1}^{m} a_{ji}u_i = c_j$

Izmantojot šo shēmu, aplūkosim dažus piemērus.

Piemērs 5.1.1°. Tiešajam uzdevumam (4.4.18)—(4.4.20) atbilstošo duālo uzdevumu var formulēt šādi.

Minimizēt lineāru funkciju

$$w = 2u_1 + 6u_2 + 6u_3 + 2u_4$$

pie ierobežojumiem

$$\begin{aligned} 2u_1 + 3u_2 + 3u_3 \qquad\quad &\geqslant 1, \\ 2u_1 - 3u_2 - 3u_3 + 2u_4 &\geqslant 1, \\ u_1 + 2u_2 - 2u_3 - 2u_4 &\geqslant 1, \\ u_1 \leqslant 0,\ u_2 \geqslant 0,\ u_3 \geqslant 0,\ u_4 &\geqslant 0. \end{aligned}$$

Piemērs 5.1.2°. Tiešajam uzdevumam (4.4.11)—(4.4.13) atbilstošo duālo uzdevumu var formulēt šādi.

Minimizēt lineāru funkciju

$$w=8u_1+6u_2+15u_3+10u_4$$

pie ierobežojumiem

$$\begin{aligned} u_1 \quad\; +u_3+u_4 &\geqslant 2, \\ u_1+u_2+u_3 \quad\; &\geqslant -1, \\ u_2+u_3+u_4 &= 5, \\ u_1\geqslant 0,\ u_2\geqslant 0,\ u_4&\leqslant 0. \end{aligned}$$

Uzdevums 5.1.1°. Formulēt duālo uzdevumu tiešajam uzdevumam (1.3.2)—(1.3.4) un ar simpleksa metodi atrisināt abus uzdevumus.

Atrisinājums. Tiešajā uzdevumā jāminimizē mērķa funkcija

$$z=5x_1+4x_2$$

pie ierobežojumiem

$$\begin{aligned} x_1+1{,}25x_2 &\geqslant 2{,}5, \\ 400x_1+\ 80x_2 &\geqslant 240, \\ x_1\geqslant 0,\ x_2 &\geqslant 0. \end{aligned}$$

Duālais uzdevums šajā gadījumā formulējams šādi.

Maksimizēt lineāru funkciju

$$w=2{,}5u_1+240u_2$$

pie ierobežojumiem

$$\begin{aligned} u_1+400u_2 &\leqslant 5, \\ 1{,}25u_1\ +80u_2 &\leqslant 4, \\ u_1\geqslant 0,\ u_2 &\geqslant 0. \end{aligned}$$

Ievedot tiešajā uzdevumā nenegatīvus papildu nezināmos y_1 un y_2, šo uzdevumu var izteikt ar vienādojumu sistēmu

$$\left.\begin{aligned} -z+0\quad +5x_1\quad +4x_2&=0, \\ -y_1-2{,}5\ +\ x_1+1{,}25x_2&=0, \\ -y_2-240+400x_1\ +80x_2&=0, \end{aligned}\right\} \tag{5.1.12}$$

kur $x_1\geqslant 0$, $x_2\geqslant 0$, $y_1\geqslant 0$, $y_2\geqslant 0$.

Ievedot duālajā uzdevumā nenegatīvus papildu nezināmos v_1 un v_2, šo uzdevumu var izteikt ar vienādojumu sistēmu

$$\left.\begin{aligned} w-0-\ 2{,}5u_1-\ 240u_2&=0, \\ v_1-5\quad +u_1+\ 400u_2&=0, \\ v_2-4+1{,}25u_1\quad +80u_2&=0, \end{aligned}\right\} \tag{5.1.13}$$

kur $u_1\geqslant 0$, $u_2\geqslant 0$, $v_1\geqslant 0$, $v_2\geqslant 0$.

Vienādojumu sistēmu (5.1.12) un (5.1.13) pierakstu var apvienot duālā iterāciju tabulā un parastajā kārtībā ar simpleksa metodes algoritmu risināt maksimizācijas uzdevumu, kā parādīts tabulās (5.1.14)—(5.1.16).

	-1		z	y_1	y_2	
		$+1$	-1	u_1	$\boxed{u_2}$	(I)
	$+1$	w	0	$-2{,}5$	-240	0
*	x_1	v_1	5	1	$\boxed{400}$	1
	x_2	v_2	4	1,25	80	2
	(I)	σ	8	$-1{,}25$	239	3

(5.1.14)

	-1		z	y_1	x_1	
		$+1$	-1	$\boxed{u_1}$	v_1	(II)
	$+1$	w	3,0000	$-1{,}9000$	0,6000	0
	y_2	u_2	0,0125	0,0025	0,0025	1
*	x_2	v_2	3,0000	$\boxed{1{,}0500}$	$-0{,}2000$	2
	(II)	σ	5,0125	$-1{,}8475$	$-0{,}5975$	3

(5.1.15)

-1		z	x_2	x_1	
	$+1$	-1	v_2	v_1	(III)
$+1$	w	8,4286	1,8095	0,2381	0
y_2	u_2	0,0054	$-0{,}0024$	0,0030	1
y_1	u_1	2,8571	0,9524	$-0{,}1905$	2
(III)	σ	10,2911	1,7595	$-0{,}9494$	3

(5.1.16)

Tabulā (5.1.16) pa rindām nolasāma šāda sistēmai (5.1.13) ekvivalenta vienādojumu sistēma:

$$\left.\begin{aligned} w-8{,}4286+1{,}8095v_2+0{,}2381v_1=0, \\ u_2-0{,}0054-0{,}0024v_2+0{,}0030v_1=0, \\ u_1-2{,}8571+0{,}9524v_2-0{,}1905v_1=0 \end{aligned}\right\}$$

jeb

$$\left.\begin{aligned} w=8{,}4286-0{,}2381v_1-1{,}8095v_2, \\ u_1=2{,}8571+0{,}1905v_1-0{,}9524v_2, \\ u_2=0{,}0054-0{,}0030v_1+0{,}0024v_2. \end{aligned}\right\} \quad (5.1.17)$$

No sistēmas (5.1.17) pirmā vienādojuma redzams, ka $\max w = 8{,}4280$, ja $v_1 = v_2 = 0$, jo koeficienti pie šiem nezināmajiem ir negatīvi un katra šo nezināmo palielināšana pamazina w. Ievietojot $v_1 = v_2 = 0$ pārējos divos sistēmas (5.1.17) vienādojumos, iegūstam $u_1 = 2{,}8571$ un $u_2 = 0{,}0054$. Tādēļ maksimizācijas uzdevuma optimālais atrisinājums, kas nolasāms tabulā (5.1.16), ir $w = 8{,}4286$, ja $u_1 = 2{,}8571$, $u_2 = 0{,}0054$, $v_1 = v_2 = 0$.

Tabulā (5.1.16) pa kolonnām nolasāma šāda sistēmai (5.1.12) ekvivalenta vienādojumu sistēma:

$$\left.\begin{aligned} -z + 8{,}4286 + 0{,}0054y_2 + 2{,}8571y_1 &= 0, \\ -x_2 + 1{,}8095 - 0{,}0024y_2 + 0{,}9524y_1 &= 0, \\ -x_1 + 0{,}2381 + 0{,}0030y_2 - 0{,}1905y_1 &= 0 \end{aligned}\right\}$$

jeb

$$\left.\begin{aligned} z &= 8{,}4286 + 2{,}8571y_1 + 0{,}0054y_2, \\ x_1 &= 0{,}2381 - 0{,}1905y_1 + 0{,}0030y_2, \\ x_2 &= 1{,}8095 + 0{,}9524y_1 - 0{,}0024y_2. \end{aligned}\right\} \qquad (5.1.18)$$

No sistēmas (5.1.18) pirmā vienādojuma redzams, ka $z = 8{,}4286$, ja $y_1 = y_2 = 0$, jo koeficienti pie šiem abiem nezināmajiem ir pozitīvi un katra šo nezināmo palielināšana palielina z. Ievietojot $y_1 = y_2 = 0$ pārējos divos sistēmas (5.1.18) vienādojumos, iegūstam $x_1 = 0{,}2381$ un $x_2 = 1{,}8095$. Tādēļ minimizācijas uzdevuma (1.3.2)—(1.3.4) optimālais atrisinājums ir $z = 8{,}4286$, ja $x_1 = 0{,}2381$, $x_2 = 1{,}8095$, $y_1 = y_2 = 0$. Tas saskan arī ar rezultātu, kuru ieguvām, atrisinot šo uzdevumu grafiski ar zīmējumu 2.2.

5.2. §. Dualitātes teorēmas. Tā kā jebkuru lineārās programmēšanas uzdevumu var pārveidot modificētajā normālformā, tad pieņemsim, ka tiešais uzdevums formulēts šādi.

Maksimizēt mērķa funkciju

$$z = \sum_{j=1}^{n} c_j x_j \qquad (5.2.1)$$

pie ierobežojumiem

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i \quad (i = 1, 2, \ldots, h < m), \qquad (5.2.2)$$

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \quad (i = h+1, h+2, \ldots, m), \qquad (5.2.3)$$

$$x_j \geqslant 0 \quad (j = 1, 2, \ldots, n), \qquad (5.2.4)$$

kur $b_i \geqslant 0$, c_j un a_{ij} ir doti reāli skaitļi.

Tiešajam uzdevumam (5.2.1)—(5.2.4) atbilstošais duālais uzdevums formulējams šādi.

Minimizēt mērķa funkciju

$$w=\sum_{i=1}^{m} b_i u_i \tag{5.2.5}$$

pie ierobežojumiem

$$\sum_{i=1}^{m} a_{ij} u_i \geqslant c_j \quad (j=1, 2, \ldots, n), \tag{5.2.6}$$

$$u_i \geqslant 0 \quad (i=1, 2, \ldots, h), \tag{5.2.7}$$

$$u_i \ (i=h+1,\ h+2, \ldots, m) \text{ — bez zīmes ierobežojuma.} \tag{5.2.8}$$

Ievedot papildu nezināmos $y_i \geqslant 0$ $(i=1, 2, \ldots, h)$ ierobežojumu sistēmā (5.2.2), mākslīgos nezināmos $\overline{y}_i \geqslant 0$ $(i=h+1,\ h+2, \ldots$ $\ldots, m)$ ierobežojumu sistēmā (5.2.3) un papildu nezināmos $v_j \geqslant 0$ $(j=1, 2, \ldots, n)$ ierobežojumu sistēmā (5.2.6), abus savstarpēji duālos uzdevumus var apvienot vienā, duālā iterāciju tabulā (5.2.9).

(5.2.9)

-1		w	v_1	v_2	$\cdots$	v_k	$\cdots$	v_n	
	$+1$	-1	x_1	x_2	$\cdots$	x_k	$\cdots$	x_n	(I)
	s	$-s_0$	$-s_1$	$-s_2$	$\cdots$	$-s_k$	$\cdots$	$-s_n$	M
$+1$	z	0	$-c_1$	$-c_2$	$\cdots$	$-c_k$	$\cdots$	$-c_n$	0
u_1	y_1	b_1	a_{11}	a_{12}	$\cdots$	a_{1k}	$\cdots$	a_{1n}	1
u_2	y_2	b_2	a_{21}	a_{22}	$\cdots$	a_{2k}	$\cdots$	a_{2n}	2
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
u_h	y_h	b_h	a_{h1}	a_{h2}	$\cdots$	a_{hk}	$\cdots$	a_{hn}	h
u_{h+1}	y_{h+1}	b_{h+1}	$a_{h+1,1}$	$a_{h+1,2}$	$\cdots$	$a_{h+1,k}$	$\cdots$	$a_{h+1,n}$	$h+1$
u_{h+2}	y_{h+2}	b_{h+2}	$a_{h+2,1}$	$a_{h+2,2}$	$\cdots$	$a_{h+2,k}$	$\cdots$	$a_{h+2,n}$	$h+2$
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
u_m	y_m	b_m	a_{m1}	a_{m2}	$\cdots$	a_{mk}	$\cdots$	a_{mn}	m
(I)	σ	σ_0-1	σ_1-1	σ_2-1	$\cdots$	σ_k-1	$\cdots$	σ_n-1	$m+1$

Šī tabula sastādīta maksimizācijas uzdevuma risināšanai ar simpleksa metodi. Risinot tiešo uzdevumu, vienlaicīgi tiek risināts arī duālais uzdevums.

Lemma 1°. *Divos savstarpēji duālos uzdevumos nevienam maksimizācijas uzdevuma plānam atbilstošā mērķa funkcijas vērtība nevar pārsniegt jebkuram minimizācijas uzdevuma plānam atbilstošo mērķa funkcijas vērtību,* t. i.,

$$\sum_{j=1}^{n} c_j x_j \leqslant \sum_{i=1}^{m} b_i u_i. \tag{5.2.10}$$

P i e r ā d ī j u m s. Reizinot katru i- to sistēmas (5.2.2) nevienādību ar $u_i \geqslant 0$, bet katru i-to sistēmas (5.2.3) vienādojumu ar u_i (bez zīmes ierobežojuma) un summējot visus šos reizinājumus, iegūst

$$\sum_{i=1}^{m} u_i \left(\sum_{j=1}^{n} a_{ij} x_j \right) \leqslant \sum_{i=1}^{m} b_i u_i. \tag{5.2.11}$$

Analogi, reizinot katru j-to sistēmas (5.2.6) nevienādību ar $x_j \geqslant 0$ un visus reizinājumus summējot, iegūst

$$\sum_{j=1}^{n} x_j \left(\sum_{i=1}^{m} a_{ij} u_i \right) \geqslant \sum_{j=1}^{n} c_j x_j. \tag{5.2.12}$$

Tā kā nevienādību (5.2.11) un (5.2.12) kreisās puses ir vienādas, tad ir pareiza nevienādība (5.2.10), ko arī vajadzēja pierādīt.

L e m m a 2°. *Ja tiešā uzdevuma plānam* x^*_j *(*$j=1, 2, \ldots, n$*) un duālā uzdevuma plānam* u^*_i *(*$i=1, 2, \ldots, m$*) izpildās vienādība*

$$\sum_{j=1}^{n} c_j x^*_j = \sum_{i=1}^{m} b_i u^*_i, \tag{5.2.13}$$

tad x^*_j *(*$j=1, 2, \ldots, n$*) ir tiešā uzdevuma optimālais plāns, bet* u^*_i *(*$i=1, 2, \ldots, m$*) ir duālā uzdevuma optimālais plāns.*

P i e r ā d ī j u m s. Saskaņā ar lemmu 1° izpildās šādas nevienādības:

$$\sum_{j=1}^{n} c_j x_j \leqslant \sum_{i=1}^{m} b_i u^*_i, \tag{5.2.14}$$

$$\sum_{j=1}^{n} c_j x^*_j \leqslant \sum_{i=1}^{m} b_i u_i. \tag{5.2.15}$$

Tā kā ir spēkā vienādība (5.2.13), tad no izteiksmes (5.2.14) izriet, ka

$$\sum_{j=1}^{n} c_j x^*_j \geqslant \sum_{j=1}^{n} c_j x_j,$$

kur x_j ($j=1, 2, \ldots, n$) ir jebkurš tiešā uzdevuma plāns. Tādēļ saskaņā ar optimālā plāna definīciju x^*_j ($j=1, 2, \ldots, n$) ir tiešā uzdevuma optimālais plāns.

Analogi, ievērojot vienādību (5.2.13), no izteiksmes (5.2.15) iegūstam

$$\sum_{i=1}^{m} b_i u^*_i \leqslant \sum_{i=1}^{m} b_i u_i,$$

kur u_i ($i=1, 2, \ldots, m$) ir jebkurš duālā uzdevuma plāns. Tādēļ saskaņā ar optimālā plāna definīciju u^*_i ($i=1, 2, \ldots, m$) ir duālā uzdevuma optimālais plāns, un lemma 2° ir pierādīta.

T e o r ē m a 1° (pirmā dualitātes teorēma). *Ja tiešajam uzdevumam eksistē optimālais plāns* x^*_j $(j=1, 2, \ldots, n)$, *tad arī duālajam uzdevumam eksistē optimālais plāns* u^*_i $(i=1, 2, \ldots, m)$, *un ir spēkā vienādība*

$$\sum_{j=1}^{n} c_j x^*_j = \sum_{i=1}^{m} b_i u^*_i. \tag{5.2.16}$$

Ja turpretim tiešā uzdevuma mērķa funkcija savā definīcijas apgabalā ir neierobežota, tad duālā uzdevuma ierobežojumu sistēma ir nesaderīga. Ja tiešā uzdevuma ierobežojumu sistēma ir nesaderīga, tad vai nu duālā uzdevuma mērķa funkcija ir neierobežota, vai arī ierobežojumu sistēma ir nesaderīga. (Pierādījumu sk. 9.5. §.)

T e o r ē m a 2° (otrā dualitātes teorēma). *Tiešā uzdevuma plāns* x^*_j $(j=1, 2, \ldots, n)$ *un atbilstošā duālā uzdevuma plāns* u^*_i $(i=1, 2, \ldots, m)$ *ir optimālie tad un tikai tad, ja izpildās nosacījumi*

$$1)\ x^*_j\left(\sum_{i=1}^{m} a_{ij} u^*_i - c_j\right) = 0,$$

$$2)\ u^*_i\left(\sum_{j=1}^{n} a_{ij} x^*_j - b_i\right) = 0.$$

P i e r ā d ī j u m s. Pieņemsim, ka x^*_j $(j=1, 2, \ldots, n)$ ir uzdevuma (5.2.1)—(5.2.4) optimālais plāns un u^*_i $(i=1, 2, \ldots, m)$ ir uzdevuma (5.2.5)—(5.2.8) optimālais plāns. Tad nevienādību (5.2.11), (5.2.12) vietā stājas vienādības

$$\sum_{j=1}^{n} c_j x^*_j = \sum_{j=1}^{n} x^*_j \left(\sum_{i=1}^{m} a_{ij} u^*_i\right), \tag{5.2.17}$$

$$\sum_{i=1}^{m} b_i u^*_i = \sum_{i=1}^{m} u^*_i \left(\sum_{j=1}^{n} a_{ij} x^*_j\right). \tag{5.2.18}$$

No vienādības (5.2.17) izriet, ka

$$\sum_{j=1}^{n} x^*_j \left(\sum_{i=1}^{m} a_{ij} u^*_i - c_j\right) = 0,$$

bet, tā kā visi reizinājumi $x^*_j\left(\sum_{i=1}^{m} a_{ij} u^*_i - c_j\right)$ ir nenegatīvi, tad ir izpildīts pirmais nosacījums. Analogi, izmantojot vienādību (5.2.18), var pārliecināties, ka ir izpildīts otrais nosacījums. Līdz ar to nosacījumu nepieciešamība ir pierādīta.

Pieņemsim, ka x^*_j $(j=1, 2, \ldots, n)$ ir uzdevuma (5.2.1)—(5.2.4) atbalsta plāns, u^*_i $(i=1, 2, \ldots, m)$ ir uzdevuma (5.2.5)—(5.2.8) atbalsta plāns un šie atbalsta plāni apmierina pirmo un otro nosacījumu. Summējot pirmos nosacījumus pēc j, bet otros nosacījumus pēc i, iegūstam

$$\sum_{j=1}^{n} \sum_{i=1}^{m} a_{ij} u^*_i x^*_j = \sum_{j=1}^{n} c_j x^*_j,$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x^*_j u^*_i = \sum_{i=1}^{m} b_i u^*_i,$$

no kurienes

$$\sum_{j=1}^{n} c_j x^*_j = \sum_{i=1}^{m} b_i u^*_i.$$

Līdz ar to nosacījumu pietiekamība un visa teorēma 2° ir pierādīta.

5.3. §. Lineārās programmēšanas modeļu jutīguma analīze. Iepriekš (4.5. §) jau minējām, ka duālā uzdevuma atrisinājums ļauj analizēt tiešā uzdevuma matemātiskā modeļa jutīgumu pret izmaiņām, kādas izdara, piemēram, ar tiešā uzdevuma mērķa funkcijas koeficientiem vai ierobežojumu sistēmas brīvajiem locekļiem.

Analīze pamatojas uz šādiem faktiem:

a) optimālā atrisinājuma (simpleksa metodes pēdējās iterācijas tabulas) nulltās rindas koeficienti pie tiešā uzdevuma papildu nezināmajiem ir vienādi ar attiecīgajām duālā uzdevuma nezināmo optimālajām vērtībām;

b) koeficienti pie tiem tiešā uzdevuma primārajiem nezināmajiem x_j, kas pēdējā tabulā palikuši nebāzes mainīgo lomā, ir vienādi ar starpību starp duālā uzdevuma j-tā ierobežojuma kreiso un labo pusi, ja šajā ierobežojumā ievieto duālā uzdevuma optimālā plāna vērtības.

Šo faktu ilustrācijai aplūkosim šādu tiešo uzdevumu.

Maksimizēt

$$z = 4x_1 + 5x_2 + 9x_3 + 11x_4 \tag{5.3.1}$$

pie ierobežojumiem

$$\left.\begin{aligned} x_1 + x_2 + x_3 + x_4 &\leqslant 15, \\ 7x_1 + 5x_2 + 3x_3 + 2x_4 &\leqslant 120, \\ 3x_1 + 5x_2 + 10x_3 + 15x_4 &\leqslant 100, \end{aligned}\right\} \tag{5.3.2}$$

$$x_j \geqslant 0 \quad (j = 1, 2, 3, 4). \tag{5.3.3}$$

Saskaņā ar tiešā un duālā uzdevuma savstarpējās atbilstības shēmu duālais uzdevums formulējams šādi.

Minimizēt

$$w = 15u_1 + 120u_2 + 100u_3 \tag{5.3.4}$$

pie ierobežojumiem

$$\left.\begin{aligned} u_1+7u_2+\ \,3u_3&\geqslant 4,\\ u_1+5u_2+\ \,5u_3&\geqslant 5,\\ u_1+3u_2+10u_3&\geqslant 9,\\ u_1+2u_2+15u_3&\geqslant 11,\end{aligned}\right\} \tag{5.3.5}$$

$$u_i\geqslant 0 \quad (i=1,\ 2,\ 3). \tag{5.3.6}$$

Apzīmējot starpības starp tiešā uzdevuma ierobežojumu sistēmas (5.3.2) labajām un kreisajām pusēm ar $y_i\geqslant 0$ ($i=1$, 2, 3) un starpības starp duālā uzdevuma ierobežojumu sistēmas (5.3.5) kreisajām un labajām pusēm ar $v_j\geqslant 0$ ($j=1$, 2, 3, 4), abus uzdevumus var apvienot šādā simpleksa metodes duālā iterāciju tabulā:

-1		w	v_1	v_2	v_3	v_4		
	$+1$	-1	x_1	x_2	x_3	$\boxed{x_4}$	(I)	
$+1$	z	0	-4	-5	-9	-11	0	
u_1	y_1	15	1	1	1	1	1	(5.3.7)
u_2	y_2	120	7	5	3	2	2	
* u_3	y_3	100	3	5	10	$\boxed{15}$	3	
(I)	σ	234	6	5	4	6	4	

Risinot maksimizācijas uzdevumu ar simpleksa metodi, tabulai (5.3.7) seko šādas duālās iterāciju tabulas:

-1		w	v_1	v_2	v_3	u_3		
	$+1$	-1	$\boxed{x_1}$	x_2	x_3	y_3	(II)	
$+1$	z	220/3	$-9/5$	$-4/3$	$-5/3$	11/15	0	
* u_1	y_1	25/3	$\boxed{4/5}$	2/3	1/3	$-1/15$	1	(5.3.8)
u_2	y_2	320/3	33/5	13/3	5/3	$-2/15$	2	
v_4	x_4	20/3	1/5	1/3	2/3	1/15	3	
(II)	σ	582/3	24/5	3	0	$-6/15$	4	

-1		w	u_1	v_2	v_3	u_3		
	$+1$	-1	y_1	x_2	$\boxed{x_3}$	y_3	(III)	
$+1$	z	1105/12	9/4	1/6	$-11/12$	7/12	0	
v_1	x_1	125/12	5/4	5/6	5/12	$-1/12$	1	(5.3.9)
u_2	y_2	455/12	$-33/4$	$-7/6$	$-13/12$	5/12	2	
* v_4	x_4	55/12	$-1/4$	1/6	$\boxed{7/12}$	1/12	3	
(III)	σ	144	-6	-1	-2	0	4	

-1		w	u_1	v_2	v_4	u_3	
	$+1$	-1	y_1	x_2	x_4	y_3	(IV)
$+1$	z	695/7	13/7	3/7	11/7	5/7	0
v_1	x_1	50/7	10/7	5/7	$-5/7$	$-1/7$	1
u_2	y_2	325/7	$-61/7$	$-6/7$	13/7	4/7	2
v_3	x_3	55/7	$-3/7$	2/7	12/7	1/7	3
(IV)	σ	1118/7	$-48/7$	$-3/7$	24/7	2/7	4

(5.3.10)

Tabulā (5.3.10) nolasāms tiešajam uzdevumam (5.3.1)—(5.3.3) ekvivalents uzdevums.

Maksimizēt

$$z=\frac{695}{7}-\frac{13}{7}y_1-\frac{3}{7}x_2-\frac{11}{7}x_4-\frac{5}{7}y_3 \tag{5.3.11}$$

pie ierobežojumiem

$$\left.\begin{aligned} x_1&=\frac{50}{7}-\frac{10}{7}y_1-\frac{5}{7}x_2+\frac{5}{7}x_4+\frac{1}{7}y_3,\\ y_2&=\frac{325}{7}+\frac{61}{7}y_1+\frac{6}{7}x_2-\frac{13}{7}x_4-\frac{4}{7}y_3,\\ x_3&=\frac{55}{7}+\frac{3}{7}y_1-\frac{2}{7}x_2-\frac{12}{7}x_4-\frac{1}{7}y_3, \end{aligned}\right\} \tag{5.3.12}$$

$$x_j\geqslant 0\ (j=1,\ 2,\ 3,\ 4);\quad y_i\geqslant 0\ (i=1,\ 2,\ 3). \tag{5.3.13}$$

Tabulā (5.3.10) nolasāms arī duālajam uzdevumam (5.3.4)—(5.3.6) ekvivalents uzdevums.

Minimizēt

$$w=\frac{695}{7}+\frac{50}{7}v_1+\frac{325}{7}u_2+\frac{55}{7}v_3 \tag{5.3.14}$$

pie ierobežojumiem

$$\left.\begin{aligned} u_1&=\frac{13}{7}+\frac{10}{7}v_1-\frac{61}{7}u_2-\frac{3}{7}v_3,\\ v_2&=\frac{3}{7}+\frac{5}{7}v_1-\frac{6}{7}u_2+\frac{2}{7}v_3,\\ v_4&=\frac{11}{7}-\frac{5}{7}v_1+\frac{13}{7}u_2+\frac{12}{7}v_3,\\ u_3&=\frac{5}{7}-\frac{1}{7}v_1+\frac{4}{7}u_2+\frac{1}{7}v_3, \end{aligned}\right\} \tag{5.3.15}$$

$$u_i\geqslant 0\ (i=1,\ 2);\quad v_j\geqslant 0\ (j=1,\ 2,\ 3,\ 4). \tag{5.3.16}$$

No izteiksmēm (5.3.11)—(5.3.13) izriet, ka tiešā uzdevuma mērķa funkcijai z ir maksimums 695/7, ja $y_1=x_2=x_4=y_3=0$, $x_1=50/7$, $y_2=325/7$ un $x_3=55/7$. No izteiksmēm (5.3.14)—(5.3.16) savukārt izriet, ka duālā uzdevuma mērķa funkcijai w ir minimums 695/7, ja $v_1=u_2=v_3=0$, $u_1=13/7$, $v_2=3/7$, $v_4=11/7$ un $u_3=5/7$. Šādas duālā uzdevuma nezināmo vērtības apmierina visus ierobežojumus (5.3.5) un minimizē mērķa funkciju w. Tiešām,

$$\left.\begin{aligned} 1\cdot 13/7+\ 3\cdot 5/7&=28/7=\ 4,\\ 1\cdot 13/7+\ 5\cdot 5/7&=38/7>\ 5,\\ 1\cdot 13/7+10\cdot 5/7&=63/7=\ 9,\\ 1\cdot 13/7+15\cdot 5/7&=88/7>11. \end{aligned}\right\} \tag{5.3.17}$$

Starpības starp sakarību (5.3.17) kreisajām un labajām pusēm ir

$$v_1=0,\ v_2=\frac{38}{7}-5=\frac{3}{7},\ v_3=0,\ v_4=\frac{88}{7}-11=\frac{11}{7}.$$

Pieņemsim, ka tiešā uzdevuma mērķa funkcijas (5.3.1) koeficienti iegūst pozitīvus pieaugumus δ_j $(j=1, 2, 3, 4)$. Tad šādus pieaugumus iegūst arī duālā uzdevuma ierobežojumu sistēmas (5.3.5) labās puses un pirmās duālās iterācijas tabulas (5.3.7) vietā stājas šāda tabula:

(5.3.18)

-1		w	v_1	v_2	v_3	v_4	
	$+1$	-1	x_1	x_2	x_3	x_4	(I)
$+1$	z	0	$-4-\delta_1$	$-5-\delta_2$	$-9-\delta_3$	$-11-\delta_4$	0
u_1	y_1	15	1	1	1	1	1
u_2	y_2	120	7	5	3	2	2
u_3	y_3	100	3	5	10	15	3

Duālās iterāciju tabulas (5.3.10) vietā savukārt rodas tabula (5.3.19), kura atšķiras no tabulas (5.3.10) tikai ar tās nulltajā rindā ierakstītajiem lielumiem, kas viegli konstruējami, ja ir jau iegūta tabula (5.3.10).

(5.3.19)

-1		w	u_1	v_2	v_4	u_3	
	1	-1	y_1	x_2	x_4	y_3	(IV)
$+1$	z	$\frac{695}{7}+\frac{50}{7}\delta_1+\frac{55}{7}\delta_3+0\cdot\delta_2$	$\frac{13}{7}+\frac{10}{7}\delta_1-\frac{3}{7}\delta_3+0\cdot\delta_2$	$\frac{3}{7}+\frac{5}{7}\delta_1+\frac{2}{7}\delta_3-\delta_2$	$\frac{11}{7}-\frac{5}{7}\delta_1+\frac{12}{7}\delta_3-\delta_4$	$\frac{5}{7}-\frac{1}{7}\delta_1+\frac{1}{7}\delta_3+0\cdot\delta_2$	0
v_1	x_1	50/7	10/7	5/7	$-5/7$	$-1/7$	1
u_2	y_2	325/7	$-61/7$	$-6/7$	13/7	4/7	2
v_3	x_3	55/7	$-3/7$	2/7	12/7	1/7	3

No tabulas (5.3.19) redzams, ka tabulā (5.3.10) atrastie optimālie plāni nemainās, ja izpildās nosacījumi

$$\left.\begin{aligned} &\frac{13}{7}+\frac{10}{7}\delta_1-\frac{3}{7}\delta_3 \geqslant 0,\\ &\frac{3}{7}+\frac{5}{7}\delta_1+\frac{2}{7}\delta_3-\delta_2\geqslant 0,\\ &\frac{11}{7}-\frac{5}{7}\delta_1+\frac{12}{7}\delta_3-\delta_4\geqslant 0,\\ &\frac{5}{7}-\frac{1}{7}\delta_1+\frac{1}{7}\delta_3 \geqslant 0. \end{aligned}\right\} \qquad (5.3.20)$$

Ievietojot izteiksmēs (5.3.20) $\delta_1=\delta_3=0$, iegūstam $\delta_2\leqslant 3/7$ un $\delta_4\leqslant 11/7$. Tas nozīmē, ka dotā uzdevuma optimālais atrisinājums nemainās, ja tiešā uzdevuma mērķa funkcijas koeficientu pie x_2 palielina par jebkuru pozitīvu skaitli, kas nav lielāks kā 3/7, un koeficientu pie x_4 — par jebkuru pozitīvu skaitli, kas nav lielāks kā 11/7.

Ievietojot izteiksmēs (5.3.20) $\delta_2=\delta_3=\delta_4=0$, iegūstam

$$\left.\begin{aligned} &\frac{13}{7}+\frac{10}{7}\delta_1\geqslant 0 \quad \text{jeb} \quad \delta_1\geqslant-\frac{13}{10},\\ &\frac{3}{7}+\frac{5}{7}\delta_1\geqslant 0 \quad \text{jeb} \quad \delta_1\geqslant-\frac{3}{5},\\ &\frac{11}{7}-\frac{5}{7}\delta_1\geqslant 0 \quad \text{jeb} \quad \delta_1\leqslant\frac{11}{5},\\ &\frac{5}{7}-\frac{1}{7}\delta_1\geqslant 0 \quad \text{jeb} \quad \delta_1\leqslant 5. \end{aligned}\right\}$$

Tas nozīmē, ka optimālais plāns nemainās, ja tiešā uzdevuma mērķa funkcijas koeficientam pie x_1 pieskaita jebkuru skaitli δ_1 no intervāla

$$-3/5\leqslant\delta_1\leqslant 11/5.$$

Analogi, ievietojot izteiksmēs (5.3.20) $\delta_1=\delta_2=\delta_4=0$, atrodam, ka optimālais plāns nemainās, ja tiešā uzdevuma mērķa funkcijas koeficientam pie x_3 pieskaita jebkuru skaitli δ_3 no intervāla

$$-11/12\leqslant\delta_3\leqslant 13/3.$$

Ja pieņemam, ka tiešā uzdevuma ierobežojumu sistēmas (5.3.2) labās puses iegūst pozitīvus pieaugumus β_i $(i=1, 2, 3)$, tad šādus pieaugumus iegūst arī duālā uzdevuma mērķa funkcijas (5.3.4) koeficienti un pirmās duālās iterācijas tabulas (5.3.7) vietā stājas šāda tabula:

-1		w	v_1	v_2	v_3	v_4	
	$+1$	-1	x_1	x_2	x_3	x_4	(I)
$+1$	z	0	-4	-5	-9	-11	0
u_1	y_1	$15+\beta_1$	1	1	1	1	1
u_2	y_2	$120+\beta_2$	7	5	3	2	2
u_3	y_3	$100+\beta_3$	3	5	10	15	3

(5.3.21)

Duālās iterācijas tabulas (5.3.10) vietā savukārt rodas tabula (5.3.22), kura atšķiras no tabulas (5.3.10) tikai ar tās brīvo locekļu kolonnā ierakstītajiem lielumiem, kas viegli konstruējami, ja ir jau iegūta tabula (5.3.10).

(5.3.22)

-1		w	u_1	v_2	v_4	u_3	
	$+1$	-1	y_1	x_2	x_4	y_3	(IV)
$+1$	z	$\frac{695}{7}+\frac{13}{7}\beta_1+\frac{5}{7}\beta_3$	$\frac{13}{7}$	$\frac{3}{7}$	$\frac{11}{7}$	$\frac{5}{7}$	0
v_1	x_1	$\frac{50}{7}+\frac{10}{7}\beta_1-\frac{1}{7}\beta_3$	$\frac{10}{7}$	$\frac{5}{7}$	$-\frac{5}{7}$	$-\frac{1}{7}$	1
u_2	y_2	$\frac{325}{7}-\frac{61}{7}\beta_1+\frac{4}{7}\beta_3+\beta_2$	$-\frac{61}{7}$	$-\frac{6}{7}$	$\frac{13}{7}$	$\frac{4}{7}$	2
v_3	x_3	$\frac{55}{7}-\frac{3}{7}\beta_1+\frac{1}{7}\beta_3$	$-\frac{3}{7}$	$\frac{2}{7}$	$\frac{12}{7}$	$\frac{1}{7}$	3

No tabulas (5.3.22) redzams, ka tabulā (5.3.10) atrastie optimālie plāni nemainās, ja izpildās nosacījumi

$$\left.\begin{aligned} &\frac{50}{7}+\frac{10}{7}\beta_1-\frac{1}{7}\beta_3\geqslant 0,\\ &\frac{325}{7}-\frac{61}{7}\beta_1+\frac{4}{7}\beta_3+\beta_2\geqslant 0,\\ &\frac{55}{7}-\frac{3}{7}\beta_1+\frac{1}{7}\beta_3\geqslant 0. \end{aligned}\right\} \tag{5.3.23}$$

Ievietojot izteiksmēs (5.3.23) $\beta_1=\beta_3=0$, iegūstam, ka $\beta_2\geqslant-325/7$. Tas nozīmē, ka optimālais atrisinājums nemainās, ja tiešā uzdevuma ierobežojumu sistēmas (5.3.2) otrās nevienādības labo pusi samazina par jebkuru pozitīvu skaitli, kas nav lielāks kā 325/7.

Ievietojot izteiksmēs (5.3.23) $\beta_2=\beta_3=0$, iegūstam

$$\left.\begin{aligned} &\frac{50}{7}+\frac{10}{7}\beta_1\geqslant 0 \quad \text{jeb} \quad \beta_1\geqslant -5,\\ &\frac{325}{7}-\frac{61}{7}\beta_1\geqslant 0 \quad \text{jeb} \quad \beta_1\leqslant \frac{325}{61},\\ &\frac{55}{7}-\frac{3}{7}\beta_1\geqslant 0 \quad \text{jeb} \quad \beta_1\leqslant \frac{55}{3}.\end{aligned}\right\}$$

Tas nozīmē, ka optimālais plāns nemainās, ja ierobežojumu sistēmas (5.3.2) pirmās nevienādības labajai pusei pieskaita jebkuru skaitli β_1 no intervāla

$$-5\leqslant\beta_1\leqslant 325/61.$$

Analogi, ievietojot izteiksmēs (5.3.23) $\beta_1=\beta_2=0$, atrodam, ka optimālais plāns nemainās, ja ierobežojumu sistēmas (5.3.2) trešās nevienādības labajai pusei pieskaita jebkuru skaitli β_3 no intervāla

$$-325/4\leqslant\beta_3\leqslant 150/7.$$

5.4. §. Kantoroviča atrisinošo reizinātāju metode. Šī metode, kas publicēta 1939. gadā, vēsturiski ir pirmā vispārīgā lineārās programmēšanas uzdevumu atrisināšanas metode. L. Kantorovičs parādīja, ka ar ražošanas jaudu izmantošanas optimālo plānu cieši saistīta īpašu reizinātāju sistēma *(duālo novērtējumu sistēma)*. Tās atrašana noved pie uzdevuma atrisinājuma, tādēļ tādus reizinātājus sauc par *atrisinošajiem reizinātājiem.*

Risināšanas process sākas ar kādu aptuvenu sākuma novērtējumu sistēmu, kuru pēc tam saskaņā ar uzdevuma nosacījumiem pakāpeniski koriģē un uzlabo. Pēc galīga soļu skaita iegūst optimālo plānu un arī duālo novērtējumu sistēmu.

Piemērs 5.4.1°. Pieņemsim, ka jāizpilda divi dažādi darbi *A* un *B*. To izpildīšanai noteiktā laika vienībā (piemēram, vienā dienā) paredzētas trīs mašīnas — I, II, III. Katrai mašīnai ir sava ražība. Mašīnu ražība darba uzskaites vienībās dota ar šādu tabulu:

Darba veidi	Mašīna		
	I	II	III
A	30	60	30
B	60	90	80

(5.4.1)

Darbu apjomi vispārīgajā gadījumā var būt dažādi. Vienkāršības labad pieņemsim, ka abu darbu kopapjomiem šajā piemērā jābūt vienādiem. Vienošanās starp organizāciju, kura darbus pasūta, un organizāciju, kas darbus izpilda, ir tāda, ka darbu apmaksa izdarāma pēc savstarpēji saskaņotām, abām pusēm pieņemamām cenām, slogojot mašīnas tā, lai tām būtu maksimālā dienas ražība. Uzdevums — izpildīt šo vienošanos.

Atrisinājums. Lai darbu novērtējumi būtu abām pusēm pieņemami, tad vispirms jāņem vērā darbietilpība. Ja visas trīs mašīnas norīkotu darbā A, tad tās dienā ražotu $30+60+30=120$ darba uzskaites vienības, bet darbā B turpretim $60+90+80=$ $=230$ vienības. Acīmredzot darbs A ir sarežģītāks un nocenojams augstāk nekā darbs B. Tādēļ ir dabiski izvēlēties tādus darbu novērtējumus, kas būtu apgriezti proporcionāli mašīnu summārai ražībai, t. i., kā pirmo jeb sākuma novērtējumu sistēmu izvēlēties skaitļus $\lambda_1=c/120$ un $\lambda_2=c/230$, kur c — kaut kāds proporcionalitātes koeficients. Ja proporcionalitātes koeficientu ņem vienādu ar $c=12\cdot 23$, tad novērtējumi ir $\lambda_1=2{,}3$ darbam A un $\lambda_2=1{,}2$ darbam B. Reizinot ar šiem novērtējumiem tabulā (5.4.1) uzrādīto mašīnu ražību, iegūst summu, kas jāmaksā par attiecīgās mašīnas izpildīto darbu. Atbilstošie darba plāna varianta skaitļi apkopoti tabulā (5.4.2).

(5.4.2)

Plāna varianti	Darba veidi	Darbu novērtējumi	Mašīnu dienas darba novērtējumi			Mašīnu dienas sadalījums			Izpildīto darbu apjoms
			I	II	III	I	II	III	
(1)	A	2,3	69	$\boxed{138}$	69		$60\cdot 1$		60
	B	1,2	$\boxed{72}$	108	$\boxed{96}$	$60\cdot 1$		$80\cdot 1$	140
(2)	A	$\frac{2}{3}$	$\boxed{20}$	$\boxed{40}$	20	$30\cdot\frac{8}{9}$	$60\cdot 1$		$86\frac{2}{3}$
	B	$\frac{1}{3}$	$\boxed{20}$	30	$\boxed{26\frac{2}{3}}$	$60\cdot\frac{1}{9}$		$80\cdot 1$	$86\frac{2}{3}$

Pirmajā plāna variantā redzamie mašīnu dienas darba novērtējumi rāda, ka mašīnai II izdevīgāks ir darbs A, jo $\max(138, 108)=138$, kas tabulā iezīmēts ar taisnstūri. Analogi $\max(69,72)=$ $=72$ un $\max(69,96)=96$. Tādēļ mašīnām I un III izdevīgāks ir darbs B. Tad mašīnas dienā paveic darbā A — 60, bet darbā B — 140 vienības. Šis plāns ir pieņemams darba izpildītājam, bet nav pieņemams pasūtītājam, jo $60\neq 140$, un pasūtītājs pieprasa, lai abi darbi būtu paveikti vienādā apjomā. Analizējot pirmo plāna variantu, redzams, ka vismazākā atšķirība ir starp mašīnas I dienas darba novērtējumiem.

Darbu izpildītājam un arī pasūtītājam nevar būt iebildumu, ja daļu no mašīnas I dienas slodzes pārceļ no darba B uz darbu A

ar nosacījumu, ka mašīnas I dienas darba novērtējumi abos darbos ir vienādi. Šo nosacījumu var realizēt ar papildu reizinātāja 72/69 palīdzību. Reizinot pirmā plāna varianta darba *A* novērtējumu ar 72/69, iegūstam jaunus darbu novērtējumus

$$\lambda_1=\frac{72}{69}\cdot 2{,}3=2{,}4;\quad \lambda_2=1{,}2.$$

Ievērojot, ka proporcija $\lambda_1:\lambda_2=2:1$ saglabājas, ja

$$\lambda_1=2/3 \text{ un } \lambda_2=1/3,$$

lietosim šos skaitļus kā jaunus darbu novērtējumus. Ar šādiem novērtējumiem iegūstam jaunu plāna variantu (2), kas arī parādīts tabulā (5.4.2). Kā redzams, mašīnas I dienas darbu novērtējumi otrajā plāna variantā abos darbos ir vienādi. Ja to darba dienas daļu, kurā mašīna I veic darbu *A*, apzīmē ar x_{11}, tad atlikusī dienas daļa, kurā šī mašīna veic darbu *B*, ir $(1-x_{11})$. Darbā *B* visu dienu tiek ieslēgta arī mašīna III, bet darbā *A* arī visu dienu — mašīna II. Pieprasot, lai abu darbu apjoms būtu vienāds, jāizpildās nosacījumam

$$30x_{11}+60=60(1-x_{11})+80. \tag{5.4.3}$$

Atrisinot vienādojumu (5.4.3), iegūstam $x_{11}=8/9$. Tātad $\frac{8}{9}$ no darba laika mašīna I jāizmanto darbā *A*, tai paveicot $30\cdot\frac{8}{9}$ darba vienības. Darbā *B* šai mašīnai jāveic $\left(1-\frac{8}{9}\right)\cdot 60=\frac{1}{9}\cdot 60$ darba vienības. Izpildīto darbu apjomi plāna otrajā variantā ir vienādi, kas atbilst uzdevuma nosacījumiem. Mašīnas šajā plāna variantā izmanto tikai tādos darbos, kuros tām ir visaugstākais dienas darba novērtējums. Iegūtais plāns ir optimālais. Atrisinošie reizinātāji (duālie novērtējumi) šajā piemērā ir $\lambda_1=2/3$ un $\lambda_2=1/3$. Šo skaitļu vietā, protams, var lietot arī jebkurus citus divus skaitļus, kuru attiecība ir 2 : 1.

Lai formulētu šī uzdevuma matemātisko modeli, apzīmēsim ar x_{ij} to darba laika daļu, kurā *i*-tajā darbā norīkota *j*-tā mašīna. Aplūkojamā piemērā ir divi darbi un trīs mašīnas. Tātad $i=1, 2$; $j=1, 2, 3$. Saskaņā ar uzdevuma nosacījumiem summa $30x_{11}+60x_{12}+80x_{23}$ nevar būt mazāka par pirmā darba (*A*) kopējo apjomu, kuru apzīmēsim ar x_0. Analogi arī summa (darba *B* kopējais apjoms) $60x_{21}+90x_{22}+80x_{23}$ nevar būt mazāka par x_0. Bez tam katras mašīnas norīkojumu ilgumu summa nevar pārsniegt pilnu darba laika ilgumu. Ievērojot to, tiešā uzdevuma matemātisko modeli var formulēt šādi.

Maksimizēt $z=x_0$ pie ierobežojumiem

$$\left.\begin{array}{l} 30x_{11}+60x_{12}+30x_{13}\geqslant x_0, \\ 60x_{21}+90x_{22}+80x_{23}\geqslant x_0, \\ x_{11}+x_{21}\leqslant 1, \\ x_{12}+x_{22}\leqslant 1, \\ x_{13}+x_{23}\leqslant 1, \end{array}\right\} \qquad (5.4.4)$$

$$x_{ij}\geqslant 0 \ (i=1,2;\ j=1,2,3).$$

Ievedot nenegatīvus papildu nezināmos $y_i\geqslant 0$ $(i=1,2,\ldots,5)$, šo uzdevumu var izteikt ar vienādojumu sistēmu, kurā maksimizējams z:

$$\left.\begin{array}{l} z-x_0=0, \\ y_1+x_0-30x_{11}-60x_{12}-30x_{13}=0, \\ y_2+x_0-60x_{21}-90x_{22}-80x_{23}=0, \\ y_3+x_{11}+\ x_{21}-1=0, \\ y_4+x_{12}+\ x_{22}-1=0, \\ y_5+x_{13}+\ x_{23}-1=0. \end{array}\right\} \qquad (5.4.5)$$

Atbilstošais duālais uzdevums ir šāds.
Minimizēt $w=u_3+u_4+u_5$ pie ierobežojumiem

$$\left.\begin{array}{ll} u_1+u_2\geqslant 1; & -60u_2+u_3\geqslant 0; \\ -30u_1+u_3\geqslant 0; & -90u_2+u_4\geqslant 0; \\ -60u_1+u_4\geqslant 0; & -80u_2+u_5\geqslant 0; \\ -30u_1+u_5\geqslant 0; & u_i\geqslant 0 \ (i=1,2,\ldots,5). \end{array}\right\} \qquad (5.4.6)$$

Ievedot nenegatīvus papildu nezināmos $v_0\geqslant 0$, $v_{ij}\geqslant 0$ $(i=1,2;$ $j=1,2,3)$, šo uzdevumu var izteikt ar vienādojumu sistēmu, kurā minimizējams w:

$$\left.\begin{array}{l} -w+u_3+u_4+u_5=0, \\ -v_0-1+u_1+u_2=0, \\ -v_{11}-30u_1+u_3=0, \\ -v_{12}-60u_1+u_4=0, \\ -v_{13}-30u_1+u_5=0, \\ -v_{21}-60u_2+u_3=0, \\ -v_{22}-90u_2+u_4=0, \\ -v_{23}-80u_2+u_5=0. \end{array}\right\} \qquad (5.4.7)$$

Vienādojumu sistēmas (5.4.5) un (5.4.7) var apvienot vienā duālā iterāciju tabulā (5.4.8):

-1		w	v_0	v_{11}	v_{12}	v_{13}	v_{21}	v_{22}	v_{23}	
	$+1$	-1	x_0	x_{11}	x_{12}	x_{13}	x_{21}	x_{22}	x_{23}	(I)
$+1$	z	0	-1	0	0	0	0	0	0	0
u_1	y_1	0	1	-30	-60	-30	0	0	0	1
u_2	y_2	0	1	0	0	0	-60	-90	-80	2
u_3	y_3	1	0	1	0	0	1	0	0	3
u_4	y_4	1	0	0	1	0	0	1	0	4
u_5	y_5	1	0	0	0	1	0	0	1	5
(I)	σ	2	0	-30	-60	-30	-60	-90	-80	6

(5.4.8)

Pēc 7 iterāciju soļiem iegūst pēdējo (astoto) duālo iterāciju tabulu (5.4.9):

(5.4.9)

-1		w	u_1	u_5	u_2	v_{13}	v_{22}	u_4	u_3	
	$+1$	-1	y_1	y_5	y_2	x_{13}	x_{22}	y_4	y_3	(VIII)
$+1$	z	$86\frac{2}{3}$	$\frac{2}{3}$	$26\frac{2}{3}$	$\frac{1}{3}$	$6\frac{2}{3}$	10	40	20	0
v_0	x_0	$86\frac{2}{3}$	$\frac{2}{3}$	$26\frac{2}{3}$	$-\frac{1}{3}$	$6\frac{2}{3}$	10	40	20	1
v_{12}	x_{12}	1	0	0	0	0	1	1	0	2
v_{21}	x_{21}	$\frac{1}{9}$	$\frac{1}{90}$	$-\frac{8}{9}$	$-\frac{1}{90}$	$-\frac{11}{9}$	$\frac{5}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	3
v_{23}	x_{23}	1	0	1	0	1	0	0	0	4
v_{11}	x_{11}	$\frac{8}{9}$	$-\frac{1}{90}$	$\frac{8}{9}$	$\frac{1}{90}$	$\frac{11}{9}$	$-\frac{5}{3}$	$-\frac{2}{3}$	$\frac{2}{3}$	5
(VIII)	σ	$175\frac{1}{3}$	$\frac{1}{3}$	$53\frac{1}{3}$	$-\frac{1}{3}$	$13\frac{1}{3}$	20	80	40	6

Tabulā (5.4.9) nolasāms tiešā uzdevuma plāns $x_{12}=1$, $x_{21}=1/9$, $x_{23}=1$, $x_{11}=8/9$, $x_0=86\frac{2}{3}$. Pārējie nezināmie vienādi ar nulli. Šādam plānam atbilstošā mērķa funkcijas $z=x_0$ vērtība $86\frac{2}{3}$ ir maksimālā, jo mērķa funkcijas izteiksme, kas nolasāma tabulā (5.4.9), ir

$$z=86\frac{2}{3}-\frac{2}{3}y_1-26\frac{2}{3}y_5-\frac{1}{3}y_2-6\frac{2}{3}x_{13}-10x_{22}-40y_4-20y_3.$$

Šī izteiksme sasniedz maksimumu, kad visi tās argumenti ir nulle, jo katra nenegatīvā argumenta palielināšana šajā izteiksmē samazina mērķa funkcijas vērtību. Tātad optimālo mašīnu noslogojumu iegūst, ja pirmajā darbā (A) izmanto otro mašīnu visu darba dienu ($x_{12}=1$), bet otro mašīnu 8/9 no darba dienas ($x_{21}= =8/9$). Otrajā darbā (B) pirmā mašīna jāizmanto 1/9 darba dienas ($x_{21}=1/9$), bet trešā mašīna visu darba dienu ($x_{23}=1$).

Tabulā (5.4.9) nolasāms arī duālā uzdevuma atrisinājums $\min w=86\frac{2}{3}$, ja $u_1=2/3$, $u_2=1/3$, $u_3=20$, $u_4=40$, $u_5=26\frac{2}{3}$, $v_{13}= =6\frac{2}{3}$, $v_{22}=10$, $v_0=v_{12}=v_{21}=v_{23}=v_{11}=0$.

Vienkāršs tabulas (5.4.9) salīdzinājums ar tabulas (5.4.2) otro variantu rāda, ka duālā uzdevuma optimālā plāna elementi $u_1=2/3$ un $u_2=1/3$ nav nekas cits kā atrisinošie reizinātāji $\lambda_1=u_1=2/3$ un $\lambda_2=u_2=1/3$. Duālā uzdevuma optimālā plāna elementi $u_3=20$, $u_4=40$ un $u_5=26\frac{2}{3}$ ir attiecīgi mašīnu I, II un III dienas darba duālie novērtējumi. Salīdzinājums parāda, ka konkrētajā piemērā atrisinošo reizinātāju metode daudz ātrāk noved pie uzdevuma atrisinājuma nekā simpleksa metode. Atrisinošo reizinātāju metodes efektivitāte pastāv apstāklī, ka daudzos speciālgadījumos lineārās programmēšanas uzdevuma optimālo atrisinājumu ar šo metodi iegūst ātrāk nekā ar citām šādu uzdevumu atrisināšanas metodēm.

6. nodaļa

TRANSPORTA UZDEVUMA ATRISINĀŠANAS METODES

6.1. §. Transporta uzdevuma ierobežojumu sistēmas un mērķa funkcijas īpašības. Klasisko transporta uzdevumu vispārīgā veidā vienmēr var formulēt šādi.

Minimizēt lineāru funkciju

$$z=\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \qquad (6.1.1)$$

vai *maksimizēt* lineāru funkciju

$$\bar{z}=\sum_{i=1}^{m}\sum_{j=1}^{n} (-c_{ij})x_{ij} \qquad (6.1.2)$$

pie ierobežojumiem

$$\sum_{j=1}^{n} x_{ij}\leqslant a_i \quad (i=1,2,\ldots,m), \qquad (6.1.3)$$

$$\sum_{i=1}^{m} x_{ij}\leqslant b_j \quad (j=1,2,\ldots,n), \qquad (6.1.4)$$

$$x_{ij}\geqslant 0 \quad (i=1,2,\ldots,m;\ j=1,2,\ldots,n), \qquad (6.1.5)$$

kur a_i — noliktavu krājumi, b_j — patērētāju pieprasījumi.

Tādā gadījumā, ja patērētāju pieprasīto kravas vienību kopsumma $\sum_{j=1}^{n} b_j$ pārsniedz noliktavās esošo kravas vienību kopsummu $\sum_{i=1}^{m} a_i$, uzdevuma ierobežojumu sistēma (6.1.3)—(6.1.5) nav saderīga. Lai uzdevumam būtu kaut viens reāls bāzes atrisinājums, tad jāizpildās dabiskam nosacījumam

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

Lietojot šādam gadījumam 1.1. § un 2.2. § aprakstītos formālos paņēmienus, klasisko transporta uzdevumu vienmēr var formulēt kanoniskajā formā, t. i., tā, kā formulēts uzdevums (1.1.7)—(1.1.11), kuru šeit atkārtosim.

Minimizēt lineāru funkciju

$$z=\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{6.1.6}$$

pie ierobežojumiem

$$\sum_{j=1}^{n} x_{ij}=a_i \quad (i=1, 2, \ldots, m), \tag{6.1.7}$$

$$\sum_{i=1}^{m} x_{ij}=b_j \quad (j=1, 2, \ldots, n), \tag{6.1.8}$$

$$x_{ij} \geqslant 0 \quad (i=1, 2, \ldots, m;\ j=1, 2, \ldots, n), \tag{6.1.9}$$

$$\sum_{i=1}^{m} a_i=\sum_{j=1}^{n} b_j, \tag{6.1.10}$$

kur a_i — krājumi, b_j — pieprasījumi.

Kā katru lineārās programmēšanas uzdevumu, arī klasisko transporta uzdevumu var atrisināt ar simpleksa metodes algoritmu. Šāda uzdevuma pirmā reducētā iterācijas tabula attēlota zīmējumā 6.1. Šajā tabulā uzskatāmi izpaužas viena no uzdevuma raksturīgākajām īpatnībām. Starp visiem tāda uzdevuma atrisinājumiem ir vismaz viens optimālais atrisinājums, kurā visi nezināmie ir veseli skaitļi, ja vien a_i $(i=1, 2, \ldots, m)$ un b_j $(j=1, 2, \ldots, n)$ ir veseli skaitļi. Ierobežojumu sistēmas īpatnība ļauj ievērojami vienkāršot transporta uzdevuma iterāciju tabulas un arī šāda uzdevuma risināšanas algoritmu. Atmetot vieniniekus un nulles (tukšās vietas), ar kurām aizpildīta zīmējumā 6.1. attēlotā tabula, transporta uzdevuma mērķa funkcijas koeficientus un nezināmos sakārto m rindās un n kolonnās atbilstoši noliktavu un patērētāju skaitam. Tā izveidoto tarifu matricu un ne-

	+1	−1	x_{11}	x_{12}	...	x_{1n}	x_{21}	x_{22}	...	x_{2n}	...	x_{m1}	x_{m2}	...	x_{mn}	(I)
	s	$-\sum_{j=1}^{n} b_j$	−1	−1	...	−1	−1	−1	...	−1	...	−1	−1	...	−1	M
	$\bar{z}$	0	c_{11}	c_{12}	...	c_{1n}	c_{21}	c_{22}	...	c_{2n}	...	c_{m1}	c_{m2}	...	c_{mn}	0
Noliktavu krājumi	u_1	a_1	1	1	...	1										1
	u_2	a_2					1	1	...	1						2
	.	.									...					.
	u_m	a_m										1	1	...	1	m
Patērētāju pieprasījumi	v_1	b_1	1				1					1				m+1
	v_2	b_2		1				1					1			m+2
	.	.			.				.		...			.		.
	v_n	b_n				1				1					1	m+n
	σ	$\sum_{j=1}^{n} b_j - 1$	c_{11}	c_{12}	...	c_{1n}	c_{21}	c_{22}	...	c_{2n}	...	c_{m1}	c_{m2}	...	c_{mn}	

Zīm. 6.1.

	v_1	v_2	v_3	...	v_n	
u_1	x_{11} c_{11}	x_{12} c_{12}	x_{13} c_{13}	...	x_{1n} c_{1n}	a_1
u_2	x_{21} c_{21}	x_{22} c_{22}	x_{23} c_{23}	...	x_{2n} c_{2n}	a_2
u_3	x_{31} c_{31}	x_{32} c_{32}	x_{33} c_{33}	...	x_{3n} c_{3n}	a_3
.	.	.	.	...	.	.
u_m	x_{m1} c_{m1}	x_{m2} c_{m2}	x_{m3} c_{m3}	...	x_{mn} c_{mn}	a_m
	b_1	b_2	b_3	...	b_n	$\sum_{j=1}^{n} b_j = \sum_{i=1}^{m} a_i$

Zīm. 6.2.

zināmo plāna matricu ieraksta kopīgā tabulā, kas attēlota zīmējumā 6.2. Tabulu noslēdz noliktavās esošo kravas vienību kolonna un patērētāju pieprasīto kravas vienību rinda. Tādas tabulas konkrēts piemērs ir arī uzdevuma (1.1.10) sākotnējās informācijas tabula (1.1.1), kurā gan nav nezināmo apzīmējumu x_{ij}.

Konkrētu transporta uzdevumu iterāciju tabulās nezināmo apzīmējumus parasti neraksta. Tiem paredzētajās vietās raksta tikai bāzes mainīgo skaitliskās vērtības, attiecīgi iezīmējot tās tabulas rūtiņas, kurās atrodas mērķa funkcijas koeficienti pie bāzes mainīgajiem.

T e o r ē m a 1°. *Transporta uzdevumam ir vismaz viens plāns tad un tikai tad, ja izpildās vienādība (6.1.10).*

P i e r ā d ī j u m s. Pieņemsim, ka $x_{ij} \geqslant 0$ $(i=1, 2, \ldots, m;$ $j=1, 2, \ldots, n)$ ir transporta uzdevuma plāns. Tā kā

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

bet šīs izteiksmes kreisā puse ir $\sum_{i=1}^{m} a_i$ un labā puse ir $\sum_{j=1}^{n} b_j$, tad nepieciešami jāizpildās vienādībai (6.1.10).

Pieņemsim, ka izpildās vienādība (6.1.10), un pierādīsim, ka ar to pietiek, lai transporta uzdevumam būtu vismaz viens plāns. Lai to pierādītu, atliek pārliecināties, ka vērtības

$$x_{ij} = \frac{a_i b_j}{Q} \geqslant 0 \quad (i=1, 2, \ldots, m; \; j=1, 2, \ldots, n)$$

apmierina ierobežojumus (6.1.7)—(6.1.8), ja

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

Tiešām,

$$\sum_{j=1}^{n} x_{ij} = \frac{a_i}{Q} \sum_{j=1}^{n} b_j = a_i \quad (i=1, 2, \ldots, m),$$

$$\sum_{i=1}^{m} x_{ij} = \frac{b_j}{Q} \sum_{i=1}^{m} a_i = b_j \quad (j=1, 2, \ldots, n).$$

Līdz ar to teorēma ir pierādīta.

L e m m a 1°. *Ja c_{ij} $(i=1, 2, \ldots, m;$ $j=1, 2, \ldots, n)$ ir dotās, bet c'_{ij} $(i=1, 2, \ldots, m;$ $j=1, 2, \ldots, n)$ — transformētas tarifu matricas elementi un p_i $(i=1, 2, \ldots, m)$, q_j $(j=1, 2, \ldots, n)$ ir kaut kādi reāli skaitļi, tad mērķa funkciju (6.1.6) var izteikt ar formulu*

$$z = \sum_{i=1}^{m} \sum_{j=1}^{n} c'_{ij} x_{ij} + \sum_{i=1}^{m} a_i p_i + \sum_{j=1}^{n} b_j q_j, \tag{6.1.11}$$

kur

$$c'_{ij} = c_{ij} - (p_i + q_j). \tag{6.1.12}$$

P i e r ā d ī j u m s. Izsakot no vienādības (6.1.12) $c_{ij}=c'_{ij}+p_i+q_j$ un ievietojot šo izteiksmi mērķa funkcijas izteiksmē (6.1.6), iegūst formulu

$$z=\sum_{i=1}^{m}\sum_{j=1}^{n}c'_{ij}x_{ij}+\sum_{i=1}^{m}\sum_{j=1}^{n}p_ix_{ij}+\sum_{i=1}^{m}\sum_{j=1}^{n}q_jx_{ij}. \qquad (6.1.13)$$

Bet, ievērojot sakarības (6.1.7)—(6.1.8), pastāv vienādības

$$\sum_{i=1}^{m}\sum_{j=1}^{n}p_ix_{ij}=\sum_{i=1}^{m}p_i\sum_{j=1}^{n}x_{ij}=\sum_{i=1}^{m}p_ia_i$$

un

$$\sum_{i=1}^{m}\sum_{j=1}^{n}q_jx_{ij}=\sum_{j=1}^{n}q_j\sum_{i=1}^{m}x_{ij}=\sum_{j=1}^{n}q_jb_j.$$

Ņemot vērā šīs vienādības, redzams, ka no izteiksmes (6.1.13) izriet formula (6.1.11), ko arī vajadzēja pierādīt.

Lai saīsinātu formulas (6.1.11) rakstību, ievedīsim apzīmējumus

$$z'=\sum_{i=1}^{m}\sum_{j=1}^{n}c'_{ij}x_{ij} \qquad (6.1.14)$$

un

$$K=\sum_{i=1}^{m}a_ip_i+\sum_{j=1}^{n}b_jq_j. \qquad (6.1.15)$$

Tad mērķa funkcija ir divu lielumu summa:

$$z=z'+K. \qquad (6.1.16)$$

Kā redzams, skaitlis K, kas definēts ar formulu (6.1.15) un kas savukārt ir divu saskaitāmo $K_1=\sum_{i=1}^{m}a_ip_i$ un $K_2=\sum_{j=1}^{n}b_jq_j$ summa, nav atkarīgs no nezināmo x_{ij} vērtībām, bet tikai no brīvi izraudzītajiem reālajiem skaitļiem p_i un q_j, kā arī no noliktavu krājumiem a_i un patērētāju pieprasījumiem b_j. Tādēļ, lai atrastu $\min z$, pietiek aprēķināt skaitli K pēc formulas (6.1.15) un pie dotā uzdevuma ierobežojumiem minimizēt lineāro funkciju (transformētā uzdevuma mērķa funkciju) z', kas definēta ar formulu (6.1.14). Acīm redzami

$$\min z=\min z'+K$$

jebkuriem brīvi izraudzītiem reāliem skaitļiem p_i un q_j, kas izmantoti skaitļa K aprēķināšanai.

Ņemot $p_i = \min(c_{i1}, c_{i2}, \ldots, c_{in})$, bet $q_j = 0$ un aprēķinot c'_{ij} ar formulu (6.1.12), iegūst transformēto tarifu matricu, kuras katrā rindā ir vismaz viena nulle. Analogi, ņemot $p_i = 0$, bet $q_j = \min(c_{1j}, c_{2j}, \ldots, c_{mj})$, katrā transformētās tarifu matricas kolonnā atradīsies vismaz viena nulle.

Transformēsim, piemēram, uzdevuma 1.1.1° tarifu matricu (1.1.2) tādā jaunā matricā, kurā nebūtu neviena negatīva elementa, bet katrā rindā un katrā kolonnā būtu vismaz viena nulle. Šim nolūkam katrā matricas rindā jāatrod vismazākais skaitlis. Pirmajā rindā tāds ir $p_1 = \min(8, 7, 6, 9, 5) = 5$, otrajā rindā $p_2 = \min(4, 10, 8, 3, 6) = 3$, trešajā rindā $p_3 = \min(2, 3, 6, 5, 3) = 2$, ceturtajā rindā $p_4 = \min(5, 4, 8, 9, 7) = 4$. Atņemot skaitļus p_1, p_2, p_3 un p_4 no visiem attiecīgās rindas elementiem, iegūst jaunu, pa rindām transformētu tarifu matricu

$$\begin{bmatrix} 3 & 2 & 1 & 4 & 0 \\ 1 & 7 & 5 & 0 & 3 \\ 0 & 1 & 4 & 3 & 1 \\ 1 & 0 & 4 & 5 & 3 \end{bmatrix} \tag{6.1.17}$$

Matricas (6.1.17) katrā kolonnā vismazākie skaitļi ir $q_1 = 0$, $q_2 = 0$, $q_3 = 1$, $q_4 = 0$, $q_5 = 0$. Atņemot tos no visiem attiecīgās kolonnas elementiem, iegūst pa rindām un kolonnām transformēto tarifu matricu

$$C' = \begin{bmatrix} 3 & 2 & 0 & 4 & 0 \\ 1 & 7 & 4 & 0 & 3 \\ 0 & 1 & 3 & 3 & 1 \\ 1 & 0 & 3 & 5 & 3 \end{bmatrix} \tag{6.1.18}$$

Matricas C' katrā rindā un katrā kolonnā ir vismaz viena nulle. Atvietojot tabulā (1.1.1) tarifu matricu (1.1.2) ar transformēto tarifu matricu (6.1.18), iegūst jauna, pārveidota uzdevuma sākotnējo tabulu

3	2	0	4	0	40
1	7	4	0	3	54
0	1	3	3	1	36
1	0	3	5	3	50
32	48	62	11	27	180

(6.1.19)

Šajā tabulā nav atkārtoti noliktavu un patērētāju apzīmējumi, bet ir pierakstīti visi turpmākam uzdevuma risinājumam nepieciešamie skaitļi, t. i., visi tarifu matricas elementi, kā arī noliktavu krājumi $a_1 = 40$, $a_2 = 54$, $a_3 = 36$, $a_4 = 50$ un patērētāju pieprasījumi $b_1 = 32$, $b_2 = 48$, $b_3 = 62$, $b_4 = 11$ un $b_5 = 27$.

Lai pie dotajiem ierobežojumiem (1.1.5)—(1.1.6) minimizētu mērķa funkciju z, kas dota ar izteiksmi (1.1.4), pietiek minimizēt pārveidotā uzdevuma mērķa funkciju

$$\begin{aligned} z' = {} & 3x_{11} + 2x_{12} + 0 \cdot x_{13} + 4x_{14} + 0 \cdot x_{15} + \\ & + 1 \cdot x_{21} + 7x_{22} + 4x_{23} + 0 \cdot x_{24} + 3x_{25} + \\ & + 0 \cdot x_{31} + 1 \cdot x_{32} + 3x_{33} + 3x_{34} + 1 \cdot x_{35} + \\ & + 1 \cdot x_{41} + 0 \cdot x_{42} + 3x_{43} + 5x_{44} + 3x_{45}. \end{aligned} \tag{6.1.20}$$

Mērķa funkcijas (1.1.4) un (6.1.20) šajā gadījumā ir saistītas ar formulu

$$z = z' + 696, \tag{6.1.21}$$

jo, tarifu matricu transformējot, tika pieņemts, ka skaitļi p_i (pa rindām) un q_j (pa kolonnām) attiecīgi ir 5, 3, 2, 4 un 0, 0, 1, 0, 0. Tādēļ saskaņā ar formulu (6.1.15) $K = K_1 + K_2$, kur $K_1 = 5 \cdot 40 + + 3 \cdot 54 + 2 \cdot 36 + 4 \cdot 50 = 634$ un $K_2 = 1 \cdot 62$. Tātad $K = 634 + 62 = 696$.

Raugoties no uzdevuma risināšanas darba kopapjoma viedokļa, tikko aprakstītā tarifu matricas transformēšana ne vienmēr ir visizdevīgākais šī darba vienkāršošanas paņēmiens. Risināšanas darba apjoms ir būtiski atkarīgs no tā, cik izdevīgs ir pirmais atrastais atbalsta plāns. Jo mazāka ir diference starp mērķa funkcijas skaitliskajām vērtībām optimālajā un bāzes atrisinājumā, jo izdevīgāks ir atrastais atbalsta plāns.

Lai atrastu izdevīgu atbalsta plānu, dažkārt (īpaši tad, ja tarifu matricas kolonnas pēc elementu skaita ir ievērojami garākas par rindām) ir lietderīgi transformēt vispirms kolonnas un pēc tam rindas.

Atbalsta plānu jebkuram transporta uzdevumam var atrast ar dažādām metodēm. Dažas no tām aplūkosim nākošajā paragrāfā.

6.2. §. Atbalsta plāns transporta uzdevumā. Šajā paragrāfā aplūkosim dažas vienkāršākās transporta uzdevuma atbalsta plāna atrašanas metodes.

1°. Diagonālā metode. Šo metodi mēdz saukt arī par «ziemeļrietumu» stūra metodi. To var lietot jebkurā gadījumā neatkarīgi no tā, vai tarifu matrica iepriekš ir transformēta vai nav.

Lai noskaidrotu diagonālās metodes algoritmu, atradīsim konkrēta transporta uzdevuma atbalsta plānu, pieņemot, ka šāda uzdevuma sākotnējā informācija dota ar tabulu (6.1.19). Tabulas (6.1.19) «ziemeļrietumu» stūrī atrodas tarifu matricas elements $c'_{11} = 3$. Tā ir maksa par katras kravas vienības nosūtīšanu pa maršrutu (1, 1), t. i., pa maršrutu no pirmās noliktavas uz pirmo patēriņa vietu. Acīm redzami pirmās noliktavas krājums (40 kravas vienības) ir pietiekams, lai pilnīgi apmierinātu pirmā patērētāja pieprasījumu (32 kravas vienības), jo $40 > 32$. Tā kā pa maršrutu (1, 1) nosūtāmo kravas vienību skaits transporta uzdevumā apzīmēts ar x_{11}, tad $x_{11}=\min(32, 40)=32$. Iegūto rezultātu var atzīmēt sākuma informācijas tabulā šādi:

$\boxed{3}^{32}$	2	0	4	0	40
1	7	4	0	3	54
0	1	3	3	1	36
1	0	3	5	3	50
32	48	62	11	27	180

Pēc pirmā patērētāja pieprasījuma apmierināšanas pirmajā noliktavā paliek $40-32=8$ kravas vienības. Izslēdzot no uzdevuma pirmo patērētāju ar minētajām 32 kravas vienībām, jāizslēdz arī maršruti (2, 1), (3, 1) un (4, 1), kas paliek nenoslogoti. Pilnīgi analogu rīcību turpina ar tabulu (6.2.1):

·	$\boxed{2}^{8}$	0	4	0	8
·	7	4	0	3	54
·	1	3	3	1	36
·	0	3	5	3	50
·	48	62	11	27	148

(6.2.1)

Tabulas (6.2.1) «ziemeļrietumu» stūrī atrodas tarifu matricas elements $c'_{12}=2$. Tā ir maksa par katras kravas vienības nosūtīšanu pa maršrutu (1, 2). Pa šo maršrutu var nosūtīt $x_{12}=\min(8, 48)=8$ kravas vienības. Tad pirmās noliktavas krājumi ir vienādi ar nulli. Vēl paliek neapmierināts otrā patērētāja pieprasījums pēc $48-8=40$ kravas vienībām. Izslēdzot no uzdevuma pirmo noliktavu, jāizslēdz arī pārējie maršruti, kuri saistās ar šo noliktavu, t. i., maršruti (1, 3), (1, 4) un (1, 5). Pilnīgi analogu rīcību turpina ar tabulu (6.2.2):

·	·	·	·	·	·
·	$\boxed{7}^{40}$	4	0	3	54
·	1	3	3	1	36
·	0	3	5	3	50
·	40	62	11	27	140

(6.2.2)

Tabulas (6.2.2) «ziemeļrietumu» stūrī atrodas tarifu matricas elements $c'_{22}=7$. Pa maršrutu (2, 2) nosūta $x_{22}=\min(40, 54)=40$ kravas vienības. Analogu rīcību turpina ar tabulu (6.2.3):

·	·	·	·	·	·
·	·	$\boxed{4}^{14}$	0	3	14
·	·	3	3	1	36
·	·	3	5	3	50
·	·	62	11	27	100

(6.2.3)

Aprakstīto rīcību soli pa solim turpinot un iegūtos rezultātus apvienojot, rodas šāda tabula:

	$\boxed{3}^{32}$	$\boxed{2}^{8}$	0	4	0	40
	1	$\boxed{7}^{40}$	$\boxed{4}^{14}$	0	3	54
	0	1	$\boxed{3}^{36}$	3	1	36
	1	0	$\boxed{3}^{12}$	$\boxed{5}^{11}$	$\boxed{3}^{27}$	50
	32	48	62	11	27	180

(6.2.4)

Tabulā (6.2.4) nolasāms viens no iespējamiem dotā uzdevuma atbalsta plāniem:

$$\left.\begin{array}{lll} x_{11}=32, & x_{12}=8, & x_{22}=40, \\ x_{23}=14, & x_{33}=36, & x_{43}=12, \\ x_{44}=11, & x_{45}=27. & \end{array}\right\} \qquad (6.2.5)$$

Visi pārējie 12 nezināmie šajā atbalsta plānā vienādi ar nulli. Šim atbalsta plānam atbilstošā mērķa funkcijas vērtība ir $z'_1=$ $=3\cdot32+2\cdot8+7\cdot40+4\cdot14+3\cdot36+3\cdot12+5\cdot11+3\cdot27=728$.

Lietojot diagonālo metodi, tieši tādu pašu atbalsta plānu (6.2.5) iegūtu uzdevumam 1.1.1°. Tikai mērķa funkcijas vērtība, kas atbilst šim atbalsta plānam, būtu $z_1=8\cdot32+7\cdot8+10\cdot40+8\times$ $\times14+6\cdot36+8\cdot12+9\cdot11+7\cdot27=1424$. Tātad $z_1=728+696=1424$, kas saskan ar formulu (6.1.21).

Tabulā (6.2.4) ir noslogoti 8 maršruti (iezīmētās rūtiņas), bet nenoslogoti $20-8=12$ maršruti. Noslogoto maršrutu skaits tātad ir par 1 mazāks nekā noliktavu un patērētāju skaita summa $4+5=9$.

2°. Minimālā elementa metode. Atšķirībā no diagonālās metodes, kas atbalsta plāna atrašanu saista ar elementu novietojumu tarifu matricā, *minimālā elementa metodē maksimāli noslogo vispirms tos maršrutus, kam ir minimālā tarifa likme (tarifu matricas elements)*. Ļoti bieži tomēr tarifu matricas minimālais elements nav nosakāms viennozīmīgi. Tā, piemēram, tabulā (6.1.19) minimālais elements ir nulle, bet nulles šajā tabulā ir vismaz pa vienai katrā rindā un katrā kolonnā. Šādā gadījumā par minimālo elementu pieņemama tāda nulle, kas savā rindā un kolonnā ir vienīgā. Ja tas nav iespējams, tad par minimālo elementu pieņemama tāda nulle, kuras rindā un kolonnā ir minimālais nuļļu skaits. Tabulā (6.1.19) tādas nulles, kas savā rindā un kolonnā ir vienīgās, ir $c'_{24}=0$, $c'_{31}=0$ un $c'_{42}=0$. Viens no šiem elementiem, piemēram, $c'_{24}=0$ jāpieņem par minimālo. Maršrutu (2, 4) var noslogot ar $x_{24}=\min(11, 54)=11$ kravas vienībām un no tabulas izslēgt ceturto patērētāju ar pārējiem tam atbilstošajiem maršrutiem (1, 4), (3, 4) un (4, 4). Analogu rīcību turpina ar tabulu (6.2.6):

3	2	0	·	0	40
1	7	4	·	3	43
0	1	3	·	1	36
1	0	3	·	3	50
32	48	62	·	27	169

(6.2.6)

Tabulā (6.2.6) tādas nulles, kas savā rindā un kolonnā ir vienīgās, ir $c'_{31}=0$, $c'_{42}=0$. Noslogojot maršrutu (3, 1) ar $x_{31}=$ $=\min(32, 36)=32$ kravas vienībām un maršrutu (4, 2) ar $x_{42}=$ $=\min(48, 50)=48$ kravas vienībām, no tabulas var izslēgt pirmo un otro patērētāju ar tiem atbilstošajiem maršrutiem un analogu rīcību turpināt ar tabulu (6.2.7):

·	·	0	·	0	40
·	·	4	·	3	43
·	·	3	·	1	4
·	·	3	·	3	2
·	·	62	·	27	89

(6.2.7)

Nākošajā solī — pēc maršruta (1, 3) noslogošanas ar $x_{13}=$ $=\min(62, 40)=40$ kravas vienībām paliek tabula (6.2.8):

·	·	·	·	·	·
·	·	4	·	3	43
·	·	3	·	1	4
·	·	3	·	3	2
·	·	22	·	27	49

(6.2.8)

Tabulā (6.2.8) minimālais elements ir $c'_{35}=1$. Pēc maršruta (3, 5) noslogošanas ar $x_{35}=\min(4, 27)=4$ kravas vienībām paliek tabula (6.2.9):

·	·	·	·	·	·
·	·	4	·	3	43
·	·	·	·	·	·
·	·	3	·	3	2
·	·	22	·	23	45

(6.2.9)

Tabulā (6.2.9) atkal ir trīs vienādi minimālie elementi. Tie katrs savā rindā un kolonnā nav arī vienīgie. Noslogojot maršrutu

(2, 5) ar $x_{25}=\min(23, 43)=23$ kravas vienībām, paliek vēl maršruts (2, 3) ar $x_{23}=20$ un maršruts (5, 3) ar $x_{53}=2$ kravas vienībām. Atrastos rezultātus apkopojot, iegūst tabulu (6.2.10):

3	2	$\boxed{0}^{40}$	4	0	40
1	7	$\boxed{4}^{20}$	$\boxed{0}^{11}$	$\boxed{3}^{23}$	54
$\boxed{0}^{32}$	1	3	3	$\boxed{1}^{4}$	36
1	$\boxed{0}^{48}$	$\boxed{3}^{2}$	5	3	50
32	48	62	11	27	180

(6.2.10)

Tabulā (6.2.10) nolasāms šāds atbalsta plāns:

$$\left.\begin{array}{llll} x_{13}=40, & x_{23}=20, & x_{24}=11, & x_{25}=23, \\ x_{31}=32, & x_{35}=4, & x_{42}=48, & x_{43}=2, \end{array}\right\} \qquad (6.2.11)$$

pārējie 12 nezināmie ir vienādi ar nulli. Šim atbalsta plānam atbilstošā mērķa funkcijas vērtība ir $z'_1=4\cdot 20+3\cdot 23+1\cdot 4+3\cdot 2=$ $=159$. Atbalsta plāns (6.2.11) apmierina visus uzdevuma 1.1.1° ierobežojumus, un tam atbilstošā šī uzdevuma mērķa funkcijas vērtība z_1 saskaņā ar formulu (6.1.21) ir $z_1=159+696=855$.

Atbalsta plāns (6.2.11) tātad ir labāks par atbalsta plānu (6.2.5), kas atrasts ar diagonālo metodi, jo $855<1424$.

6.3. §. Atbalsta plāna pakāpeniska uzlabošana ar sadalījumu metodi. Katrā transporta uzdevumā atbalsta plānu skaits ir galīgs lielums, kuru var aprēķināt ar formulu (sk. 9.7. §)

$$C_{mn}^{m+n-1}=\frac{(mn)!}{(m+n-1)!(mn-m-n+1)!}.$$

Te m ir noliktavu skaits, bet n — patērētāju skaits. Iepriekšējā paragrāfā aplūkotajam piemēram atbalsta plānu skaits ir

$$C_{20}^{8}=\frac{20!}{8!12!}\approx 120\,000.$$

Ja no visiem atbalsta plāniem atrasts viens, tad katru reizi rūpīgi jāpārbauda, vai šis plāns nav uzlabojams.

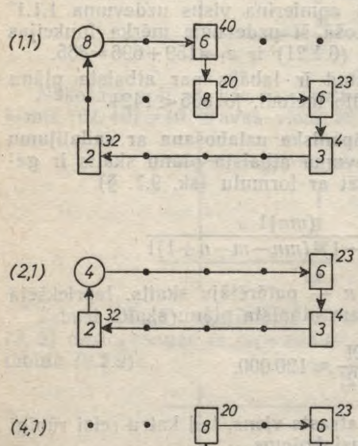
Lai atrastu atbalsta plāna uzlabošanas metodi, aplūkosim uzdevuma 1.1.1° atbalsta plānu (6.2.11), kas atrasts ar minimālā elementa metodi un nolasāms tabulā (6.2.10). Atrastā atbalsta plāna uzlabošanas iespēju pārbaudei var lietot tabulu (6.2.10) vai arī uzdevuma 1.1.1° sākotnējās informācijas tabulu, kurā iezīmēti noslogotajiem maršrutiem atbilstošie netransformētās tarifu matricas elementi, pierakstot atbalsta plāna nezināmo vērtības, kā parādīts tabulā (6.3.1):

8	7	$\boxed{6}^{40}$	9	5	40
4	10	$\boxed{8}^{20}$	$\boxed{3}^{11}$	$\boxed{6}^{23}$	54
$\boxed{2}^{32}$	3	6	5	$\boxed{3}^{4}$	36
5	$\boxed{4}^{48}$	$\boxed{8}^{2}$	9	7	50
32	48	62	11	27	180

(6.3.1)

Tabula (6.3.1) ir noderīga tarifu matricas transformāciju un mērķa funkcijas vērtību aprēķinu pareizības pārbaudei. Tiešām, aprēķinot tabulai (6.3.1) atbilstošo mērķa funkcijas vērtību, iegūst $z_1 = 6\cdot40+8\cdot20+3\cdot11+6\cdot23+2\cdot32+3\cdot4+4\cdot48+8\cdot2=855$. Šis skaitlis ir vienāds ar iepriekšējā paragrāfa beigās aprēķināto vērtību $z_1=855$, kas iegūta ar formulu (6.1.21). Gadījumā, ja abi rezultāti nav vienādi, tad aprēķinos pieļauta kļūda, kas jāatrod un jāizlabo.

Tabulā (6.3.1), tāpat kā tabulā (6.2.10) un katrā citā atbalsta plānu saturošā tabulā, kurā nezināmo skaits ir $4\cdot5=20$, jābūt $4+5-1=8$ iezīmētiem maršrutiem. Pa noslogotajiem maršrutiem transportējamo kravas vienību daudzumi ir uzdevuma bāzes mainīgo skaitliskās vērtības. Pārējie $20-8=12$ nezināmie ir nebāzes mainīgie, kuru skaitliskās vērtības atbalsta plānā vienādas ar nulli. Tos tabulā neiezīmē.



Zīm. 6.3.

Katra kravas vienība, kuru no atbalsta plānā iezīmēta maršruta pārliek uz kādu neiezīmētu maršrutu, rada izmaiņas vismaz 3 iezīmēto maršrutu noslogojumos. Tā sekas ir arī pozitīvs vai negatīvs mērķa funkcijas pieaugums, t. i., kopējo transporta izmaksu palielināšanās vai samazināšanās. Tā, piemēram, ja tabulā (6.3.1) neiezīmēto maršrutu (1,1) noslogo ar vienu kravas vienību, tad šī vienība jāatņem no maršrutu (1, 3) un (3, 1) noslogojumiem, bet, ja to dara, tad savukārt par vienu kravas vienību jāpalielina maršrutu (2, 3) un (3, 5) noslogojumi. Beidzot, ja palielina maršrutu (2, 3) un (3, 5) noslogojumu, tad jāsamazina maršruta (2, 5) noslogojums. Tā rodas *noslēgta ķēde (cikls)* (1, 1), kas attēlota zīmējumā 6.3. Šajā ciklā ietilpst maršruti (1, 1), (1, 3), (2, 3), (2, 5), (3, 5) un (3, 1) ar tarifiem $c_{11}=8$, $c_{13}=6$, $c_{23}=8$, $c_{25}=6$, $c_{35}=3$ un $c_{31}=2$.

Definīcija 1°. *Par ciklu matricā sauc tādu tās elementu virkni, kurā katri divi (bet ne trīs vai vairāk) blakus esošie virknes locekļi atrodas vienā šīs matricas rindā vai kolonnā. Virknes pirmais un pēdējais loceklis atrodas vienā rindā vai vienā kolonnā. Šādas virknes locekļus sauc par cikla virsotnēm.*

Cikla grafiskā attēla līnijas posmi pie katras cikla virsotnes veido taisnu leņķi.

Zīmējumā 6.3 ar (2, 1) apzīmēts tabulas (6.3.1) elementam $c_{21}=4$ *atbilstošais cikls* (2, 1), (2, 5), (3, 5), (3, 1), un tajā pašā zīmējumā ar atzīmi (4, 1) attēlots elementam $c_{41}=5$ atbilstošais cikls (4, 1), (4, 3), (2, 3), (2, 5), (3, 5), (3, 1).

Neiezīmēto tarifu matricas elementu sauc par šim matricas elementam atbilstošā cikla *pirmo virsotni*. Otrā, trešā un visas turpmākās cikla virsotnes ir iezīmētie matricas elementi. Cikla virsotņu kopskaits vienmēr ir pāru skaitlis, kas nav mazāks par 4.

Tās cikla virsotnes, kam kārtas numuri ir nepāru skaitļi 1, 3, 5, . . . , veido tarifu matricas elementu virkni, kuru sauc par *pozitīvo pusķēdi* pretstatā *negatīvajai pusķēdei*. Negatīvo pusķēdi veido cikla virsotnes, kuru kārtas numuri ir pāru skaitļi 2, 4, 6, ...

Ja cikla pirmajai virsotnei atbilstošajam maršruta noslogojumam (kas vienāds ar nulli) pieskaita vienu kravas vienību, tad tā jānoņem no visiem negatīvajai pusķēdei atbilstošajiem maršrutiem un jāpieskaita ne vien cikla pirmajai virsotnei, bet arī visām pārējām pozitīvās pusķēdes virsotnēm atbilstošajiem maršrutu noslogojumiem. Tas nozīmē, ka *katra kravas vienība, kuru noņem negatīvajai pusķēdei un pieskaita pozitīvajai pusķēdei atbilstošajiem maršrutu noslogojumiem, samazina mērķa funkcijas vērtību par visu negatīvās pusķēdes virsotnēs esošo tarifu matricas elementu kopsummu, bet, no otras puses, palielina mērķa funkcijas vērtību par visu pozitīvās pusķēdes virsotnēs esošo tarifu matricas elementu kopsummu.*

Tā, piemēram, tabulas (6.3.1) neiezīmētam elementam $c_{11}=8$ atbilstošā cikla negatīvās pusķēdes virsotnēs atrodas tarifu matricas elementi $c_{13}=6$, $c_{25}=6$ un $c_{31}=2$. To summa ir $6+6+2=14$. Pozitīvās pusķēdes virsotnēs atrodas tarifu matricas elementi $c_{11}=8$, $c_{23}=8$ un $c_{35}=3$. To summa ir $8+8+3=19$. Tātad vienas kravas vienības pieskaitīšana maršruta (1, 1) noslogojumam rada mērķa funkcijas vērtības pieaugumu par $19-14=5$ vienībām. Tādēļ maršrutam (1, 1) atbilstošā nezināmā x_{11} izslēgšana no tabulas (6.3.1) nebāzes mainīgajiem, kuru vērtība atbalsta plānā vienāda ar nulli, apmaiņai pret kādu no bāzes mainīgajiem radītu mērķa funkcijas vērtības pieaugumu par $\Delta_{11}x_{11}$ vienībām. Te $\Delta_{11}=$ $=8-6+8-6+3-2=19-14=5$. Skaitli $\Delta_{11}=5$ sauc par *maršruta (1, 1) novērtējumu.*

Jebkura maršruta (i, j) novērtējumu Δ_{ij} var iegūt, izveidojot šim maršrutam, resp., tarifu matricas neiezīmētam elementam c_{ij} atbilstošo ciklu. Izveidotajā ciklā pēc kārtas ar alternējošām zīmēm algebriski summē visus šī cikla virsotnēs esošos tarifu matricas elementus. Cikla pirmajai virsotnei liek zīmi «+», otrajai «−», trešajai «+» utt.

Aplūkojamā piemērā $\Delta_{21}=4-6+3-2=-1$; $\Delta_{41}=5-8+8-6+$ $+3-2=0$. Analogi atrodami visi pārējie tabulas (6.3.1) nebāzes maršrutu novērtējumi. Tie sakopoti tabulā (6.3.2):

5	5	□	8	1
−1	6	□	□	□
□	2	1	5	□
0	□	□	6	1

(6.3.2)

Pozitīvie nebāzes maršrutu novērtējumi norāda, ka šiem maršrutiem atbilstošie nezināmie nav apmaināmi lomām ar kādu no bāzes nezināmajiem. Tāda apmaiņa atbalsta plānu nevis uzlabotu, bet gan pasliktinātu, jo palielinātos šim plānam atbilstošā mērķa funkcijas vērtība. Tas nozīmē, ka rakstības saīsināšanas nolūkā pietiek, ja pozitīvo novērtējumu vietā tabulā (6.3.1) liek punktu. Tāds rakstības saīsinājums ļauj apvienot tabulas (6.3.1) un (6.3.2) vienā tabulā:

8.	7.	$\boxed{6}^{40}$	9.	5.	40
4_{-1}	10.	$\boxed{8}^{20}$	$\boxed{3}^{11}$	$\boxed{6}^{23}$	54
$\boxed{2}^{32}$	3.	6.	5.	$\boxed{3}^{4}$	36
5_0	$\boxed{4}^{48}$	$\boxed{8}^{2}$	9.	7.	50
32	48	62	11	27	180

(6.3.3)

Tabulā (6.3.2), tātad arī apvienotajā tabulā (6.3.3), ir tikai viens nebāzes maršruts ar negatīvu novērtējumu $\Delta_{21}=-1$. Tas nozīmē, ka katra kravas vienība, kuru no šim maršrutam atbilstošā cikla negatīvās puskēdes virsotnēm pārvieto uz pozitīvās puskēdes virsotnēm, samazina mērķa funkcijas vērtību par vienu tarifa vienību. Šim maršrutam atbilstošais cikls (2, 1) ar 4 virsotnēm attēlots zīmējumā 6.3. Ja mazāko no negatīvās puskēdes maršrutu noslogojumiem, t. i., min $(23, 32)=23$ pieskaita abām pozitīvās puskēdes virsotnēm un atņem no abām negatīvās puskēdes virsotnēm, tad iegūst jaunu, uzlabotu atbalsta plānu, kuram atbilst samazināta mērķa funkcijas vērtība $z_2=832$. Šis atbalsta plāns nolasāms tabulā (6.3.4):

8.	7.	$\boxed{6}^{40}$	9.	5.	40
$\boxed{4}^{23}$	10.	$\boxed{8}^{20}$	$\boxed{3}^{11}$	6.	54
$\boxed{2}^{9}$	3.	6_0	5.	$\boxed{3}^{27}$	36
5.	$\boxed{4}^{48}$	$\boxed{8}^{2}$	9.	7.	50
32	48	62	11	27	180

(6.3.4)

Sastādot visiem tabulas (6.3.4) nebāzes maršrutiem atbilstošos ciklus un aprēķinot katram maršrutam tā novērtējumu, iegūtie rezultāti atzīmēti tajā pašā tabulā (6.3.4).

Kā redzams, tabulā (6.3.4) nav neviena tāda nebāzes maršruta, kura novērtējums būtu negatīvs. Tādēļ tālāka atbalsta plāna uzlabošana nav iespējama un tabulā (6.3.4) nolasāmais atbalsta plāns ir optimālais. Tātad uzdevuma 1.1.1° optimālais atrisinājums ir

$$x_{13}=40,\quad x_{21}=23,\quad x_{23}=20,\quad x_{24}=11,$$
$$x_{31}=9,\quad x_{35}=27,\quad x_{42}=48,\quad x_{43}=2,$$

visi pārējie nezināmie šajā plānā vienādi ar nulli un $\min z=832$.

Aprakstīto transporta uzdevuma atbalsta plāna uzlabošanas un optimālā plāna atrašanas paņēmienu sauc par *sadalījumu metodi* lineārajā programmēšanā. Sadalījumu metode ir transporta uzdevuma ierobežojumu sistēmas īpatnībām piemērots simpleksa metodes paveids.

Viena atbalsta plāna nomaiņu ar tam sekojošu nepasliktinātu atbalsta plānu sauc par iterācijas soli. Sākot uzdevuma 1.1.1° atrisināšanai vajadzīgo iterācijas procesu ar atbalsta plānu (6.2.11), optimālo plānu, kā redzējām, iegūst pēc viena iterācijas soļa izpildes. Ja šo procesu sāktu no atbalsta plāna (6.2.5), tad būtu vajadzīgi vismaz 8 iterācijas soļi, lai iegūtu optimālo plānu.

Transporta uzdevumam ir viens vienīgs optimālais plāns, ja visi optimālā plāna nebāzes maršrutu novērtējumi ir pozitīvi. Ja kaut viens no šiem novērtējumiem ir nulle, tad uzdevumam ir vismaz vēl viens optimālais atbalsta plāns un bezgalīgi daudz optimālo plānu, kas nav atbalsta plāni. Tādi alternatīvi optimālie plāni ir arī uzdevumam 1.1.1°, jo optimālā plāna tabulā (6.3.4) nebāzes maršruta (3, 3) novērtējums ir nulle. Sastādot šim maršrutam atbilstošo ciklu (3, 3), (3, 1), (2, 1), (2, 3), no tā negatīvajām virsotnēm (3, 1) un (2, 3) atņemot $\min(9, 20)=9$

kravas vienības un kieskaitot tās pozitīvajām virsotnēm, iegūst jaunu atbalsta plānu, kurš nolasāms tabulā (6.3.5):

8	7	$\boxed{6}^{40}$	9	5	40
$\boxed{4}^{32}$	10	$\boxed{8}^{11}$	$\boxed{3}^{11}$	6	54
2	3	$\boxed{6}^{9}$	5	$\boxed{3}^{27}$	36
5	$\boxed{4}^{48}$	$\boxed{8}^{2}$	9	7	50
32	48	62	11	27	180

(6.3.5)

Tabulā (6.3.5) nolasāmais atbalsta plāns arī ir optimālais. Tam atbilstošā mērķa funkcijas vērtība ir vienāda ar iepriekšējo, min $z=$ $=832$. Šī uzdevuma optimālā mērķa funkcijas vērtība nemainās arī tad, ja no maršrutam (3, 3) atbilstošā cikla negatīvajām virsotnēm atņem un pozitīvajām virsotnēm pieskaita tikai daļu no negatīvo virsotņu minimālā noslogojuma, t. i., ja negatīvo virsotņu noslogojumu samazina par 9λ vienībām un par tikpat daudz vienībām palielina pozitīvo virsotņu noslogojumu. Te λ ir jebkurš no daļskaitļiem 1/9, 2/9, ..., 8/9. Tā, piemēram, ja $\lambda=2/9$, tad iegūst plānu, kas nolasāms tabulā (6.3.6):

8	7	$\boxed{6}^{40}$	9	5	40
$\boxed{4}^{25}$	10	$\boxed{8}^{18}$	$\boxed{3}^{11}$	6	54
$\boxed{2}^{7}$	3	$\boxed{6}^{2}$	5	$\boxed{3}^{27}$	36
5	$\boxed{4}^{48}$	$\boxed{8}^{2}$	9	7	50
32	48	62	11	27	180

(6.3.6)

Tabulā (6.3.6) nolasāmais plāns nav atbalsta plāns, jo iezīmēto maršrutu skaits ir $9>8$, bet tam atbilstošā mērķa funkcijas vērtība ir vienāda ar min $z=832$.

Aprakstot sadalījuma metodi lineārajā programmēšanā, tika pieņemts, ka atbalsta plāns nav *singulārs,* t. i., ka neviens no bāzes mainīgajiem nav nulle. Singularitāte transporta uzdevumam tomēr ir bieža parādība. Tā rodas katrreiz, ja, pirmo atbalsta plānu meklējot, kādas noliktavas A_i krājumi a_i ir vienādi ar patēriņa vietas B_j pieprasījumiem b_j un nezināmo x_{ij} izvēlas par bāzes mainīgo. Tādā gadījumā maršruta (i, j) rindā vai kolonnā tūdaļ jāiezīmē vēl viens bāzes maršruts ar noslogojumu 0. Pretējā gadījumā samazinās bāzes mainīgo skaits, kuram katrā atbalsta plānā jābūt vienādam ar noliktavu un patēriņa vietu skaita summu mīnus 1.

Singularitāte parādās arī tad, ja bāzē ieslēdzamam maršrutam atbilstoša cikla negatīvajās virsotnēs ir divi vai vairāki vienādi mazākie maršrutu noslogojumi. Tādā gadījumā no bāzes mainīgajiem izslēdz tikai vienu negatīvo virsotni, bet pārējās atstāj bāzē ar noslogojumu 0.

Teorēma 1° (par ciklu). *Ja matricā, kurai ir m rindas un n kolonnas ($m \neq 1$, $n \neq 1$), iezīmē brīvi izraudzītus $m+n$ elementus, tad vienmēr var konstruēt ciklu, kura virsotnes ir vismaz daļa no iezīmētajiem elementiem.*

Pierādījums. Minimālais rindu un kolonnu skaits pēc teorēmas nosacījuma ir $m=2$, $n=2$. Šādam gadījumam, kurā iezīmēto elementu skaits nav mazāks par 4, teorēma ir pierādīta, jo visi 4 elementi ir cikla virsotnes.

Pieņemsim, ka $m>2$ un $n>2$, ka teorēma ir pareiza $m+n-1$ brīvi izraudzītiem iezīmētiem elementiem matricā, kuras rindu un kolonnu skaita summa ir $m+n-1$, un pierādīsim, ka tā ir pareiza arī $m+n$ brīvi izraudzītiem iezīmētiem elementiem matricā, kurai ir m rindas un n kolonnas.

Pieļausim, ka kādā ($m \times n$) tipa matricas rindā (kolonnā) ir viens vienīgs iezīmēts elements. Ja izsvītro šo rindu (kolonnu), matricas rindu (kolonnu) skaits par vienu samazinās, bet par vienu samazinās arī iezīmēto elementu skaits. Tādēļ šādā gadījumā saskaņā ar pieņēmumu var konstruēt ciklu, kura virsotnes ir vismaz daļa no iezīmētajiem elementiem.

Ja ($m \times n$) tipa matricā nav tādas rindas vai kolonnas, kurā būtu tikai viens no iezīmētajiem $m+n$ elementiem, tad katrā rindā (kolonnā) ir vismaz divi iezīmēti elementi vai arī nav neviena tāda elementa. Atzīmējot vienu no iezīmētajiem elementiem ar «+» un nosaucot to par cikla 1. virsotni, savienosim šo virsotni ar tajā pašā rindā (kolonnā) esošo otro elementu, nosaucot to par 2. virsotni. 2. virsotni savienosim ar tajā pašā kolonnā (rindā) esošo otro elementu, nosaucot to par 3. virsotni, utt. Tādu savienošanu nevar turpināt neierobežoti, jo iezīmēto elementu skaits ir galīgs. Pēc galīga pagriezienu skaita jāatgriežas kolonnā (rindā), kurā atrodas ar zīmi «+» atzīmētais elements, un cikls noslēdzas.

Tā kā teorēma ir pareiza gadījumam $m+n=4$, tad, pēc pierādītā, tā ir pareiza arī gadījumiem $m+n=5$, $m+n=6$ utt. Tātad tā ir pareiza jebkurai matricai, kuras rindu un kolonnu skaits ir lielāks par 1.

6.4. §. Potenciālu metode un dualitāte. Maršrutu novērtējumu atrašana ar iepriekšējā paragrāfā aprakstīto sadalījumu metodi liela nezināmo skaita gadījumā ir ļoti apgrūtinoša. Vienkāršāka ir potenciālu metode, kura pamatojas uz šādu teorēmu.

Teorēma 1°. *Transporta uzdevuma plāns x_{ij} ($i=1, 2, \ldots, m$; $j=1, 2, \ldots, n$) ir optimāls, ja eksistē tādi skaitļi u_i ($i=1, 2, \ldots, m$) un v_j ($j=1, 2, \ldots, n$), ka visiem maršrutiem izpildās nosacījums*

$$u_i+v_j \leqslant c_{ij}, \tag{6.4.1}$$

bet visiem slogotajiem maršrutiem — nosacījums

$$u_i+v_j=c_{ij}, \tag{6.4.2}$$

kur c_{ij} ir maršruta (i, j) tarifs.

Pierādījums. Pieņemsim, ka ir atrasti tādi skaitļi $u_i=u^*_i$ un $v_j=v^*_j$, kas apmierina teorēmas nosacījumus, un pierādīsim, ka tad transporta uzdevuma plāns $x_{ij}=x^*_{ij}$ $(i=1, 2, \ldots, m;$ $j=1, 2, \ldots, n)$ ir optimālais.

Tā kā skaitļi p_i un q_j formulās (6.1.11)—(6.1.12) ir jebkuri reāli skaitļi, tad to vietā šajās formulās var ievietot dotos skaitļus u^*_i un v^*_j un formulas pārrakstīt šādi:

$$z=z'+K^*, \tag{6.4.3}$$

kur

$$z'=\sum_{i=1}^{m}\sum_{j=1}^{n} c^*_{ij}x_{ij},$$

$$K^*=\sum_{i=1}^{m} a_iu^*_i+\sum_{j=1}^{n} b_jv^*_j,$$

$$c^*_{ij}=c_{ij}-(u^*_i+v^*_j)\geqslant 0.$$

Ja maršruts (i, j) ir slogots, t. i., ja $x_{ij}=x^*_{ij}>0$, tad $u^*_i+v^*_j=c_{ij}$ un $c^*_{ij}=0$, bet, ja maršruts (i, j) nav slogots, tad $x_{ij}=x^*_{ij}=0$. Tas nozīmē, ka pie dotajiem u^*_i un v^*_j funkcijas z' vērtība formulā (6.4.3) ir nulle un mērķa funkcija z sasniedz savu apakšējo robežu K^*. Tātad $x_{ij}=x^*_{ij}$ $(i=1, 2, \ldots, m;$ $j=1, 2, \ldots, n)$ ir optimālais plāns, ko arī vajadzēja pierādīt.

Teorēma 2° (apgrieztā teorēma). *Ja x^*_{ij} $(i=1, 2, \ldots, m;$ $j=1, 2, \ldots, n)$ ir klasiskā transporta uzdevuma (6.1.6)—(6.1.10) optimālais plāns, tad vienmēr eksistē skaitļi u^*_i $(i=1, 2, \ldots, m)$ un v^*_j $(j=1, 2, \ldots, n)$, kas apmierina nosacījumus (6.4.1)—(6.4.2).*

Pierādījums. Klasisko transporta uzdevumu var formulēt kā duālo uzdevumu šādam tiešajam lineārās programmēšanas uzdevumam.

Maksimizēt

$$w=\sum_{i=1}^{m} a_iu_i+\sum_{j=1}^{n} b_jv_j \tag{6.4.4}$$

pie ierobežojumiem

$$u_i+v_j\leqslant c_{ij} \tag{6.4.5}$$

$$(i=1, 2, \ldots, m;\ j=1, 2, \ldots, n),$$

$$u_i \text{ un } v_j \text{ — bez zīmes ierobežojuma} \tag{6.4.6}$$

vai

$$\Delta_{ij}=c_{ij}-(u_i+v_j)\geqslant 0. \tag{6.4.7}$$

Ja x^*_{ij} $(i=1, 2, \ldots, m;\ j=1, 2, \ldots, n)$ ir tiešajam uzdevumam (6.4.4)—(6.4.6) atbilstošā duālā uzdevuma (6.1.6)—(6.1.10) optimālais plāns, tad saskaņā ar pirmo dualitātes teorēmu 5.2.1° arī tiešajam uzdevumam eksistē optimālais plāns $u_i=u^*_i$ $(i=1, 2, \ldots, m)$ un $v_j=v^*_j$ $(j=1, 2, \ldots, n)$, kurš apmierina nosacījumus (6.4.5). Šim plānam atbilst lineārās funkcijas w maksimālā vērtība, kas ir vienāda ar funkcijas z minimālo vērtību. Tā kā uzdevumam (6.1.6)—(6.1.10) vienmēr eksistē optimālais atrisinājums, tad vienmēr eksistē tādi skaitļi u^*_i $(i=1, 2, \ldots, m)$ un v^*_j $(j=1, 2, \ldots, n)$, kuri apmierina nosacījumus (6.4.1)—(6.4.2), ko arī vajadzēja pierādīt.

Tiešo teorēmu 1° un apgriezto teorēmu 2° var apvienot vienā *pamatteorēmā par potenciāliem*. Formulēsim to.

Teorēma 3°. *Transporta uzdevuma plāns* x_{ij} $(i=1, 2, \ldots, m;$ $j=1, 2, \ldots, n)$ *ir optimālais tad un tikai tad, ja eksistē tādi skaitļi* u_i $(i=1, 2, \ldots, m)$ *un* v_j $(j=1, 2, \ldots, n)$, *kuri apmierina nosacījumus*

$$u_i+v_j=c_{ij} \text{ visiem } x_{ij}>0,$$

$$u_i+v_j\leqslant c_{ij} \text{ visiem } x_{ij}=0.$$

Skaitļus $u_1, u_2, \ldots, u_m$ sauc par *krājumu potenciāliem*, bet $v_1, v_2, \ldots, v_n$ — par *pieprasījumu potenciāliem*.

Lai pārbaudītu, vai kādu iegūtu atbalsta plānu var uzlabot, tad vispirms jāatrisina potenciālu noteikšanas vienādojumu sistēma (6.4.2), t. i., jāatrod tādi skaitļi $u_1, u_2, \ldots, u_m$ un $v_1, v_2, \ldots, v_n$, kas apmierina visus vienādojumus (6.4.2). Šajā ļoti vienkāršajā lineārajā vienādojumu sistēmā nav vairāk kā $m+n-1$ lineāri neatkarīgu vienādojumu ar $m+n$ nezināmajiem. Tādai vienādojumu sistēmai ir bezgalīgi daudz atrisinājumu, jo vienmēr vienu nezināmo var izvēlēties brīvi un pēc tam pakāpeniski citu pēc cita atrisināt lineārus vienādojumus $x+a=b$, kur a un b ir jau atrasti, zināmi skaitļi. Paskaidrosim to ar piemēru, atrodot tabulai (6.2.10) vienu no bezgalīgi daudzajām potenciālu kopām, kas apmierina nosacījumus (6.4.2). Atrisināmā potenciālu noteikšanas vienādojumu sistēma ir šāda:

$$\left.\begin{array}{llllllll}
u_1 & & & & +v_3 & & & =0,\\
u_2 & & & & +v_3 & & & =4,\\
u_2 & & & & & +v_4 & & =0,\\
u_2 & & & & & & +v_5 & =3,\\
u_3 & +v_1 & & & & & & =0,\\
u_3 & & & & & & +v_5 & =1,\\
u_4 & & +v_2 & & & & & =0,\\
u_4 & & & & +v_3 & & & =3.
\end{array}\right\} \qquad (6.4.8)$$

Lai atrisinātu šādu vienādojumu sistēmu, kurā ir 8 vienādojumi ar 9 nezināmajiem, vienu nezināmo var izvēlēties brīvi.

Parasti visbiežāk sastopamo nezināmo izvēlas vienādu ar nulli. Ņemot, piemēram, $v_3=0$, sistēmā (6.4.8) nolasām, ka $u_1=0$, $u_2=4$ un $u_4=3$. Ievietojot tā atrastās potenciālu vērtības pārējos šīs sistēmas vienādojumos un tos pēc kārtas atrisinot, iegūstam vienu no potenciālu kopām, kas apmierina vienādojumu sistēmu (6.4.2). Tāda kopa ir $u_1=0$, $u_2=4$, $u_3=2$, $u_4=3$, $v_1=-2$, $v_2=-3$, $v_3=0$, $v_4=-4$ un $v_5=-1$. Atrastos potenciālus pieraksta tajā pašā tabulā (6.2.10), kurā iezīmēti bāzes maršruti ar tiem atbilstošajiem tarifiem un bāzes nezināmo (maršrutu slogojuma) vērtībām. Tā rodas tabula (6.4.9):

$u_i \backslash v_j$	-2	-3	0	-4	-1	
0	3	2	$\boxed{0}^{40}$	4	0	40
4	1	7	$\boxed{4}^{20}$	$\boxed{0}^{11}$	$\boxed{3}^{23}$	54
2	$\boxed{0}^{32}$	1	3	3	$\boxed{1}^{4}$	36
3	1	$\boxed{0}^{48}$	$\boxed{3}^{2}$	5	3	50
	32	48	62	11	27	180

(6.4.9)

Lai atrastu potenciālu kopu, kas apmierina visus sistēmas (6.4.2) vienādojumus, nav nepieciešams šo sistēmu atsevišķi izrakstīt. Tā bez grūtībām nolasāma tabulā (6.2.10), jo katrs šai tabulā iezīmētais tarifu matricas elements ir tā vienādojuma labā puse, kura kreisajā pusē atrodas iezīmētajam maršrutam atbilstošā krājumu potenciālu un pieprasījumu potenciālu summa. Ja brīvi pieņemts viens no potenciāliem, piemēram, $v_3=0$, un pierakstīts tabulā (6.4.9), tad šajā tabulā tūdaļ var ierakstīt arī $u_1=0$, $u_2=4$ un $u_4=3$. Ja tabulā ierakstīts $u_2=4$, tad jābūt $v_4=-4$ un $v_5=-1$, bet, ja $v_5=-1$, tad $u_3=2$ un tālāk $v_1=-2$. Tā kā $u_4=3$ jau tabulā ierakstīts, tad $v_2=-3$ un visi vajadzīgie 9 potenciāli ir atrasti.

Pēc tam ar formulu

$$\Delta_{ij}=c_{ij}-(u_i+v_j) \tag{6.4.10}$$

aprēķina visu nenoslogoto maršrutu novērtējumus, kas saskan ar tabulā (6.3.2) ierakstītajiem novērtējumiem, kurus ieguvām, izmantojot sadalījumu metodi. Tiešām,

$$\Delta_{11}=3-(-2+0)=5,$$

$$\Delta_{21}=1-(4-2)=-1,$$

$$\cdots\cdots\cdots\cdots$$

$$\Delta_{44}=5-(3-4)=6,$$

$$\Delta_{45}=3-(3-1)=1.$$

Atbalsta plāna uzlabošana pēc novērtējumu atrašanas neatšķiras no tā paņēmiena, kuru lieto sadalījumu metodē.

Apvienojot tabulu (6.3.2) ar tabulu (6.4.9) un neizrakstot pozitīvo novērtējumu skaitliskās vērtības, iegūstam šādu dotā transporta uzdevuma iterāciju tabulu:

	−2	−3	0	−4	−1	
0	3.	2.	$\boxed{0}^{40}$	4.	0.	40
4	1^*_{-1}	7.	$\boxed{4}^{20}$	$\boxed{0}^{11}$	$\boxed{3}^{23}$	54
2	$\boxed{0}^{32}$	1.	3.	3.	$\boxed{1}^{4}$	36
3	1_0	$\boxed{0}^{48}$	$\boxed{3}^{2}$	5.	3.	50
	32	48	62	11	27	180

(6.4.11)

Nākošais, uzlabotais atbalsta plāns un tam atbilstošā potenciālu sistēma parādīta tabulā (6.4.12):

	−3	−3	0	−4	−2	
0	3.	2.	$\boxed{0}^{40}$	4	0	40
4	$\boxed{1}^{23}$	7	$\boxed{4}^{20}$	$\boxed{0}^{11}$	3	54
3	$\boxed{0}^{9}$	1	3_0	3	$\boxed{1}^{27}$	36
3	1	$\boxed{0}^{48}$	$\boxed{3}^{2}$	5	3	50
	32	48	62	11	27	180

(6.4.12)

Tabulā (6.4.12) nolasāmais plāns ir optimālais, jo visu nenoslogoto maršrutu novērtējumi ir nenegatīvi. Šim plānam atbilstošā mērķa funkcijas vērtība ir $z' = 1\cdot 23 + 4\cdot 20 + 3\cdot 2 + 1\cdot 27 = 136$.

Tabulā (6.4.12) nolasāmais optimālais plāns saskan ar tabulā (6.3.4) nolasāmo optimālo plānu. Risinot uzdevumu ar netransformētu tarifu matricu, iegūtais $\min z = 832$ atšķiras no $\min z' = 136$ par skaitli $K = 696$, kas aprēķināts 6.1. §; $\min z = 136 + 696 = 832$.

6.5. §. Aproksimācijas metode. Noslēdzot 6.2. §, atzīmējām, ka parasti transporta uzdevuma atbalsta plāns, kuru atrod ar minimālā elementa metodi, ir labāks par atbalsta plānu, kas atrasts ar diagonālo metodi. Vēl efektīvāka ir *aproksimācijas metode* atbalsta plāna atrašanai, jo ar šo metodi atrastais atbalsta plāns bieži vien sakrīt ar optimālo plānu vai ir ļoti tuvs optimālajam plānam. Līdz ar to ievērojami saīsinās atbalsta plāna uzlabošanas process, kurš jāveic ar potenciālu metodi vai arī ar sadalījumu metodi. Neiztirzājot aproksimācijas metodes vispārīgo pamatojumu, aplūkosim šīs metodes pielietojumu konkrētā uzdevumā. Tam nolūkam izmantosim uzdevuma 1.1.1° sākotnējās informācijas tabulu (1.1.1), transformējot tarifu matricu pa rindām un kolonnām, kā aprakstīts 6.1. §, un atvietojot doto tabulu (1.1.1) ar tabulu (6.1.19), kuru šeit atkārtosim kā tabulu (6.5.1):

	I	II	III	IV	V	
1.	3	2	0	4	0	40
2.	1	7	4	0	3	54
3.	0	1	3	3	1	36
4.	1	0	3	5	3	50
	32	48	62	11	27	180

(6.5.1)

Aproksimācijas metodes pirmajā solī aprēķina *divu vismazāko elementu diferences* katrā rindā un katrā kolonnā. Mūsu piemērā tās ir šādas:

1\. $0-0=0$, I $1-0=1$,
2\. $1-0=1$, II $1-0=1$,
3\. $1-0=1$, III $3-0=3$,
4\. $1-0=1$, IV $3-0=3$,
V $1-0=1$.

Ar šīm diferencēm paplašina tabulu (6.5.1):

	I	II	III	IV	V		
1.	3	2	0	4	0	40	0
2.	1	7	4	0	3	54	1
3.	0	1	3	3	1	36	1
4.	1	0	3	5	3	50	1
	32	48	62	11	27	180	
	1	1	3	3	1		

(6.5.2)

Vislielākā no atrastajām diferencēm norāda rindu vai kolonnu, kuras vismazākais elements ir tā maršruta tarifs, kurš meklējamā atbalsta plānā ir maksimāli noslogojams.

Atrastās tarifu diferences ekonomiski nozīmē transporta izdevumu samazinājumu par tik tarifa vienībām uz katru kravas vienību, cik liela ir dotā diference kādai noliktavai (tarifu matricas rindai) vai patērētājam (tarifu matricas kolonnai).

Aplūkojamā piemērā maksimāli slogojamais maršruts meklējams tabulas (6.5.2) III vai IV kolonnā, jo šīm kolonnām atbilst vienādas (vislielākās) diferences 3. Izvēloties no tām III kolonnu, maksimāli jāslogo maršruts (1, 3), jo tam atbilst III kolonnas vismazākais tarifs 0. Noslogojot maršrutu (1, 3) ar $x_{13}=$ $=\min(40, 62)=40$ kravas vienībām, no tabulas var izslēgt 1. noliktavu ar visiem tai atbilstošajiem maršrutiem. Analogu rīcību turpina nākošajā solī ar tabulu (6.5.3). Turpmākajā diferenču aprēķināšanā nav ņemami vērā jau noslogoto maršrutu tarifi.

	I	II	III	IV	V		
1.	·	·	$\boxed{0}^{40}$	·	·	·	·
2.	1	7	4	0	3	54	1
3.	0	1	3	3	1	36	1
4.	1	0	3	5	3	50	1
	32	48	22	11	27	140	
	1	1	0	3	2		

(6.5.3)

Tabulā (6.5.3) maksimālā diference atbilst IV kolonnai, kuras mazākais elements ir maksimāli slogojamā maršruta (2, 4) tarifs. Slogojot šo maršrutu ar $x_{24} = \min(11, 54) = 11$ kravas vienībām, no tabulas var izslēgt IV kolonnu ar visiem tai atbilstošajiem maršrutiem. Analogu rīcību turpina ar tabulu (6.5.4):

	I	II	III	IV	V		
1.	·	·	$\boxed{0}^{40}$	·	·	·	·
2.	1	7	4	$\boxed{0}^{11}$	3	43	2
3.	0	1	3	·	1	36	1
4.	1	0	3	·	3	50	1
	32	48	22	·	27	129	
	1	1	0	·	2		

(6.5.4)

Nākošajā solī iegūst tabulu (6.5.5):

	I	II	III	IV	V		
1.	·	·	$\boxed{0}^{40}$	·	·	·	·
2.	1	7	4	$\boxed{0}^{11}$	·	43	3
3.	0	1	3	·	$\boxed{1}^{27}$	9	1
4.	1	0	3	·	·	50	1
	32	48	22	·	·	102	
	1	1	0	·	·		

(6.5.5)

Aprakstītais paņēmiens soli pa solim jāturpina tik ilgi, kamēr visas 180 kravas vienības ir sadalītas pa maršrutiem. Gala rezultātā rodas tabula (6.5.6):

3	2	$\boxed{0}^{40}$	4	0	40
$\boxed{1}^{32}$	7	$\boxed{4}^{11}$	$\boxed{0}^{11}$	3	54
0	1	$\boxed{3}^{9}$	3	$\boxed{1}^{27}$	36
1	$\boxed{0}^{48}$	$\boxed{3}^{2}$	5	3	50
32	48	62	11	27	180

(6.5.6)

Salīdzinot tabulu (6.5.6) ar tabulu (6.3.5), redzams, ka ar aproksimācijas metodi atrastais atbalsta plāns šajā piemērā ir arī optimālais.

6.6. §. Diferenciālo renšu metode. Diferenciālo renšu metodi transporta uzdevuma atrisināšanai proponējis A. Lurjē un teorētiski pamatojis A. Brudno*. Šai metodei ir ciešs sakars ar

* Брудно А. Л. Метод дифференциальных рент Лурье для определения плана оптимальных перевозок. — «Докл. АН СССР», 1960, т. 131, № 6, с. 1238—1241.

Kantoroviča atrisinošo reizinātāju metodi. Paskaidrosim diferenciālo renšu metodi ar to pašu piemēru, kuru izmantojām, izklāstot sadalījuma metodi un potenciālu metodi. Sākotnējo informāciju, kas dota ar tabulu (1.1.1), šajā gadījumā ir lietderīgi pierakstīt tā, kā parādīts tabulā (6.6.1), atstājot pagaidām neaizņemtu kolonnu D un rindu Δ:

Noliktavas	Patērētāji					D
	32	48	62	11	27	
40	8	7	6	9	5	
54	4	10	8	3	6	
36	2	3	6	5	3	
50	5	4	8	9	7	
Δ						

(6.6.1)

Apzīmējumi. Pieņemsim, ka vispārīgā gadījumā ir dota klasiskā transporta uzdevuma tarifu matrica $(c_{ij})_{(m\times n)}$ un šajā matricā dažiem elementiem ir apvilkts taisnstūrītis. Katram tā apzīmētam elementam, kuru turpmāk sauksim par *rūtiņu*, atbilst noteikts *iezīmēts maršruts.*

Definīcija 1° (izsvītrojamība). *Iezīmēto maršrutu vai rūtiņu sistēmu sauc par izsvītrojamu, ja rūtiņas var sanumurēt tā, lai katra no tām atrastos tādā matricas rindā vai kolonnā, kurā nav rūtiņu ar lielākiem numuriem. Tādu numerāciju sauc par izsvītrojošu.*

Izsvītrošanas algoritms A1. *Izvēlas rūtiņu, kas savā rindā vai kolonnā ir vienīgā* (ja tāda ir), *apzīmē to ar numuru 1 un izsvītro. Pēc tam no atlikušajām rūtiņām atkal izvēlas rūtiņu, kas savā rindā vai kolonnā ir vienīgā* (ja tāda ir), *apzīmē to ar numuru 2 un izsvītro. Analogi izsvītro rūtiņas ar numuriem 3, 4 utt.*

Ja rūtiņu sistēma ir izsvītrojama, tad ar algoritmu A1 iegūst rūtiņu izsvītrojošo numerāciju. Neizsvītrojamas rūtiņu sistēmas gadījumā algoritms A1 noved pie tādas matricas, kurai katrā rindā un katrā kolonnā paliek iezīmētas vismaz divas rūtiņas.

Rūtiņu sistēma ir izsvītrojama tad un tikai tad, ja iezīmēto rūtiņu skaits matricā nepārsniedz skaitli $m+n-1$, kur $m\times n$ ir dotās matricas rindu un kolonnu skaita reizinājums.

Definīcija 2°. *Skaitļus* μ_i $(i=1,2,\ldots,m)$ *sauc par noliktavu jaudu (vai krājumu* a_i*) atlikumiem, bet* α_j $(j=1,2,\ldots,n)$ — *par patērētāju kapacitāšu (vai pieprasījumu* b_j*) atlikumiem, ja* $x_{ij}\geqslant 0$ *ir maršrutu (i, j) noslogojumi un*

$$\left.\begin{aligned}\mu_i&=a_i-\sum_{j=1}^{n}x_{ij}\geqslant 0,\\ \alpha_j&=b_j-\sum_{i=1}^{m}x_{ij}\geqslant 0.\end{aligned}\right\}\qquad(6.6.2)$$

Algoritms A2. *Pieņemsim, ka ir dota pēc algoritma A1 sanumurēta rūtiņu sistēma, ka* $x_{ij}=0$ *visiem maršrutiem, kas nav iezīmēti, ka* $\mu^0_i \equiv a_i$ un $\alpha^0_j \equiv b_j$ *un ka* $x^k \equiv x_{i(k),\ j(k)}$, *kur ar* $i(k)$ un $j(k)$ *apzīmēta tā rinda un kolonna, kurā atrodas rūtiņa ar numuru* k. *Tad visu iezīmēto maršrutu noslogošanu izdara rūtiņu numuru kārtībā pēc formulas*

$$x^k = \min(\mu^{k-1}_{i(k)},\ \alpha^{k-1}_{j(k)}) \quad (k=1, 2, \ldots). \tag{6.6.3}$$

Pēc tam kad visi sanumurētajām rūtiņām atbilstošie maršruti saskaņā ar algoritmu A2 ir noslogoti, katrai noliktavai var dot pozitīvu vai negatīvu *raksturojumu* atkarībā no tā, kāda ir dotās noliktavas jauda salīdzinājumā ar tai piesaistīto patērētāju kapacitāti.

Definīcija 3°. *Doto noliktavu un tai atbilstošo tarifu matricas rindu sauc par pozitīvu, ja, palielinot šīs vienas noliktavas jaudu par* $\Delta > 0$, *nevar palielināt dotajai rūtiņu sistēmai atbilstošo iezīmēto maršrutu summāro noslogojumu. Pretējā gadījumā noliktavu un tai atbilstošo tarifu matricas rindu sauc par negatīvu. Rūtiņas, kuras atrodas pozitīvā (negatīvā) rindā, sauc par pozitīvām (negatīvām).*

Pieņemsim, ka 1) S^p ir izsvītrojamo rūtiņu sistēma, 2) σ ir pozitīva, bet ξ — negatīva šīs sistēmas rūtiņa, kas abas atrodas vienā tarifu matricas kolonnā. Tādā gadījumā rūtiņu sistēmām S^p un $S^p-\xi$ atbilstošo iezīmēto maršrutu summārie noslogojumi un noliktavu raksturojumi ir vienādi.

Pieņemsim, ka 1) S^p un $S^p+\sigma$ ir izsvītrojamas rūtiņu sistēmas, 2) rūtiņa σ atrodas pozitīvā sistēmas S^p rindā un tādā tās kolonnā, kurā nav nevienas pozitīvas rūtiņas, 3) abām rūtiņu sistēmām atbilstošo iezīmēto maršrutu summārie noslogojumi ir vienādi. Tādā gadījumā ir spēkā šādas statiskas pazīmes noliktavu raksturošanai: 1) tā sistēmas $S^p+\sigma$ rinda, kurā atrodas rūtiņa σ, ir negatīva, 2) negatīvas ir visas tās sistēmas $S^p+\sigma$ rindas, kuras ir negatīvas sistēmā S^p.

Pieņemsim, ka noliktavu jaudu kopsumma ir vienāda ar patērētāju kapacitāšu kopsummu un pēc algoritma A1 iezīmēto maršrutu noslogošana izdarīta saskaņā ar algoritmu A2. Tādā gadījumā pastāv šādas statiskas pazīmes noliktavu raksturošanai: 1) ja $\mu_i > 0$, tad i-tā rinda ir pozitīva; 2) ja $\alpha_j > 0$, tad visas k-tajā kolonnā iezīmētās rūtiņas ir negatīvas; 3) ja kolonnā ir viena pozitīva rūtiņa, tad visas rūtiņas, kurām atbilstošie iezīmētie maršruti ir ar pozitīviem noslogojumiem, ir pozitīvas.

Pieņemot, ka klasiskais transporta uzdevums formulēts tā, ka noliktavu jaudu kopsumma ir vienāda ar patērētāju kapacitāšu kopsummu, aplūkosim soli pa solim vienu diferenciālo renšu metodes ciklu šāda uzdevuma atrisināšanai.

Sākotnējo rūtiņu sistēmu, kurā ietilpst tik rūtiņu, cik kolonnu ir tarifu matricai, veido pirmajā ciklā. Šajā ciklā lieto doto tarifu matricu $(c_{ij})_{(m\times n)}=(c^1{}_{ij})_{(m\times n)}$ un katrā tās stabiņā iezīmē vienu minimālo elementu. Iegūto n rūtiņu sistēmu apzīmēsim ar S^1. Aplūkojamam piemēram (6.6.1) atbilstošajā rūtiņu sistēmā S^1 ietilpst 5 rūtiņas, kas iezīmētas ar taisnstūriem tabulā (6.6.4):

I	32	48	62	11	27	D
40	8	7	$\boxed{6}^{40}$	9	5	-22
54	4	10	8	$\boxed{3}^{11}$	6	43
36	$\boxed{2}^{32}$	$\boxed{3}^{4}$	6	5	$\boxed{3}^{0}$	-71
50	5	4	8	9	7	50
Δ	2	①	2	3		93

(6.6.4)

Tabulā (6.6.4) izdarīta rūtiņu numerācija saskaņā ar algoritmu A1 un šīm rūtiņām atbilstošo iezīmēto maršrutu noslogošana saskaņā ar algoritmu A2: $x_{31}=32$, $x_{32}=4$, $x_{13}=40$, $x_{24}=11$ un $x_{35}=0$. Atbilstoši definīcijai 3° kolonnā D uzrādīti noliktavu raksturojumi šajā ciklā. Pozitīvas ir otrā un ceturtā noliktava ar uzviju $+43$ un $+50$, bet negatīvas — pirmā un trešā noliktava ar deficītiem -22 un -71. Pirmajā ciklā šajā piemērā tātad nesadalītas paliek 93 kravas vienības. Iezīmēto maršrutu summārais noslogojums ir $32+4+40+11+0=87$ kravas vienības.

Pirms p-tā cikla sākuma $(p>1)$ ir zināma iepriekšējā, $p-1$ cikla rūtiņu sistēma S^{p-1} un sistēmai S^{p-1} atbilstošie noliktavu raksturojumi, kā arī $p-1$-ajā ciklā lietotā tarifu matrica $(c^p{}_{ij})_{(m\times n)}$. Kārtējais, p-tais cikls sākas ar soli 1°.

Solis 1°. *Šajā solī veido p-tā cikla rūtiņu sistēmu S^p:* 1) kolonnā, kurā nav nevienas pozitīvas rūtiņas, atrod vienu vismazāko pozitīvas rindas elementu $a^p{}_{i'j'}$. Rūtiņa (i', j') pieder pie p-tā cikla rūtiņu sistēmas S^p; 2) rūtiņu sistēmā S^{p-1} atmet tās negatīvās rūtiņas, kuras atrodas vienā kolonnā ar kādu pozitīvu rūtiņu. Visas pārējās sistēmas S^{p-1} rūtiņas, kas papildinātas ar rūtiņu (i', j'), sastāda rūtiņu sistēmu S^p.

Solis 2°. Saskaņā ar algoritmu A1 *izdara izsvītrojošo numerāciju* rūtiņu sistēmā.

Solis 3°. Saskaņā ar algoritmu A2 pēc solī 2° izdarītās numerācijas *izdara iezīmēto maršrutu noslogošanu.* Ja visas kravas vienības sadalītas, tad process ir noslēdzies. Pretējā gadījumā jāizpilda solis 4°.

Solis 4°. Saskaņā ar definīciju 3°, izpildot soli 3°, kā arī izmantojot statiskās pazīmes, *atrod visu noliktavu raksturojumus.* Var izmantot arī šādu pazīmi noliktavu raksturošanai: ja tā rūtiņa sistēmā S^p, kura nebija sistēmā S^{p-1}, ir pozitīva (negatīva), tad visas tās rindas, kuras sistēmā S^{p-1} bija pozitīvas (negatīvas), ir pozitīvas (negatīvas) arī sistēmā S^p.

S o l i s 5°. Šajā solī *veido nākošā, $p+1$-mā cikla tarifu matricu* $(c_{ij}^{p+1})_{(m\times n)}$. Tam nolūkam 1) aprēķina lielumus

$$\Delta^p = \min_{j'} (\min_{i'} c^p_{ij} - \min_{i} c^p_{ij}), \tag{6.6.5}$$

kur j' ir tādu kolonnu numuri (sistēmā S^p), kurās nav nevienas pozitīvas rūtiņas; i' ir pozitīvo rindu numuri (sistēmā S^p), bet i — jebkuras rindas numurs p-tā cikla tarifu matricā. Lielumu Δ^p sauc par *starprenti*; 2) visiem p-tā cikla tarifu matricas $(c^p_{ij})_{(m\times n)}$ negatīvo rindu elementiem pieskaita starprenti Δ^p, atstājot nemainītus visu pozitīvo rindu elementus.

Tā iegūst jaunu, $p+1$-mā cikla tarifu matricu $(c_{ij}^{p+1})_{(m\times n)}$ un izpilda soli 1° ciklā $p+1$.

Saskaņā ar formulu (6.6.5) tabulas (6.6.4) rindā Δ atzīmētas diferences starp pozitīvo rindu minimālajiem elementiem un to kolonnās esošajiem vismazākajiem elementiem. Pirmajā kolonnā šī diference ir $4-2=2$, otrajā $4-3=1$, trešajā $8-6=2$ un piektajā $6-3=3$. Ceturtajā kolonnā atrodas pozitīva rūtiņa, tādēļ tai diference nav aprēķināma. Tā kā $\min\{2, 1, 2, 3\}=1$, tad saskaņā ar formulu (6.6.5) tabulai (6.6.4) atbilstošā starprente ir $\Delta^1=1$.

Pieskaitot atrasto starprenti $\Delta^1=1$ visiem pirmā cikla tarifu matricas negatīvo rindu elementiem, bet nemainot visu pozitīvo rindu elementus, iegūst otrā cikla tarifu matricu, kas redzama tabulā (6.6.6):

II	32	48	62	11	27	D
40	9	8	$\boxed{7}^{40}$	10	6	−22
54	$\boxed{4}^{32}$	10	8	$\boxed{3}^{11}$	6	11
36	$\boxed{3}^{0}$	$\boxed{4}^{9}$	7	6	$\boxed{4}^{27}$	−39
50	5	4	8	9	7	50
Δ		0	1		2	61

(6.6.6)

Otrā cikla pirmajā solī iepriekšējā cikla rūtiņām pievienota rūtiņa (2, 1). Maršrutu noslogojums izsvītrojošās rūtiņu numerācijas kārtībā ir $x_{13}=40$, $x_{24}=11$, $x_{35}=27$, $x_{32}=9$, $x_{31}=0$, $x_{21}=32$. Noliktavu novērtējumi sakrīt ar pirmā cikla novērtējumiem. Nesadalīto kravas vienību kopskaits ir 61. Otrā cikla starprente ir 0. Tādēļ trešā cikla tarifu matrica, kas redzama tabulā (6.6.7), neatšķiras no otrā cikla tarifu matricas:

III	32	48	62	11	27	D
40	9	8	$\boxed{7}^{40}$	10	6	−22
54	$\boxed{4}^{32}$	10	8	$\boxed{3}^{11}$	6	11
36	3	$\boxed{4}^{9}$	7	6	$\boxed{4}^{27}$	0
50	5	$\boxed{4}^{39}$	8	9	7	11
Δ			0			22

(6.6.7)

Trešā cikla pirmajā solī no otrā cikla jāatmet negatīvā rūtiņa (3, 1), jo tā atrodas vienā kolonnā ar pozitīvo rūtiņu (2, 1). No jauna trešajā ciklā figurē rūtiņa (4, 2). Nesadalītas paliek 22 kravas vienības. Arī šī cikla starprente ir 0.

Ceturtais cikls attēlots tabulā (6.6.8):

IV	32	48	62	11	27	D
40	9	8	$\boxed{7}^{40}$	10	6	−0
54	$\boxed{4}^{32}$	10	8	$\boxed{3}^{11}$	6	11
36	3	$\boxed{4}^{0}$	$\boxed{7}^{9}$	6	$\boxed{4}^{27}$	−13
50	5	$\boxed{4}^{48}$	8	9	7	2
Δ			①		2	13

(6.6.8)

Ceturtajā ciklā ir 7 rūtiņas. No jauna iesaistīta rūtiņa (3, 3). Maršrutu noslogojumi izsvītrojošās numerācijas kārtībā ir $x_{13}=40$, $x_{21}=32$, $x_{24}=11$, $x_{35}=27$, $x_{33}=9$, $x_{32}=0$ un $x_{42}=48$. Šim ciklam ir divas pozitīvas un divas negatīvas rindas, diferenciālā starprente ir 1. Nesadalītais kravas vienību skaits ir 13.

Piektajā ciklā, kā redzams tabulā (6.6.9), ir jauna tarifu matrica, bet rūtiņu skaits nepalielinās. No jauna iesaistīta rūtiņa (4, 3), bet atmesta negatīvā rūtiņa (3, 2), kas atrodas vienā kolonnā ar pozitīvo rūtiņu (4, 2).

V	32	48	62	11	27	D
40	10	9	$\boxed{8}^{40}$	11	7	−0
54	$\boxed{4}^{32}$	10	8	$\boxed{3}^{11}$	6	11
36	4	5	$\boxed{8}^{20}$	7	$\boxed{5}^{16}$	−11
50	5	$\boxed{4}^{48}$	$\boxed{8}^{2}$	9	7	−0
Δ		6	⓪		1	11

(6.6.9)

Process šajā piemērā noslēdzas ar sesto ciklu, kas attēlots tabulā (6.6.10):

VI	32	48	62	11	27	D
40	10	9	$\boxed{8}^{40}$	11	7	0
54	$\boxed{4}^{32}$	10	$\boxed{8}^{11}$	$\boxed{3}^{11}$	6	0
36	4	5	$\boxed{8}^{9}$	7	$\boxed{5}^{27}$	0
50	5	$\boxed{4}^{48}$	$\boxed{8}^{2}$	9	7	0
Δ						0

(6.6.10)

Tabulā (6.6.10) izsvītrojošās numerācijas kārtībā nolasāms optimālais transporta plāns: $x_{13}=40$, $x_{21}=32$, $x_{42}=48$, $x_{43}=2$, $x_{24}=11$, $x_{34}=27$, $x_{35}=27$, $x_{33}=9$, $x_{23}=11$. Ja dotā transporta uzdevuma noliktavu jaudu kopsumma ir vienāda ar patērētāju kapacitāšu kopsummu un, risinot šo uzdevumu ar diferenciālo renšu metodi, līdz p-tajam ciklam nav sadalītas visas kravas vienības, tad var pierādīt, ka 1) tarifu matrica $(c^p_{ij})_{(m\times n)}$ un rūtiņu sistēma S^p ir pilnīgi noteiktas; sistēma S^p ir izsvītrojama, un, ja $p>1$, tajā ir rūtiņa, kuras nav sistēmā S^{p-1}; 2) pārejot no $p-1$-mā cikla uz p-to ciklu, nesadalītais kravas vienību skaits nepalielinās; ja tas nesamazinās, tad pieaug negatīvo noliktavu skaits; 3) visas sistēmas S^p rūtiņas iezīmē savu kolonnu minimālos tarifu matricas $(c^p_{ij})_{(m\times n)}$ elementus; 4) diferenciālo renšu metode pēc galīga ciklu skaita noved pie optimālā transporta plāna.

7. nodaļa

TĪKLVEIDA PLĀNOŠANAS ELEMENTI

7.1. §. Vispārīgie jēdzieni par grafu un tīklu. Pieņemsim, ka ir dota punktu kopa X un orientētu loku kopa U, piemēram, 6 punktu kopa

$$X=\{P_1, P_2, P_3, P_4, P_5, P_6\} \tag{7.1.1}$$

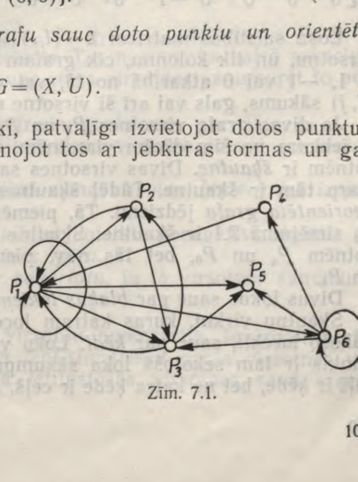
un 15 orientētu loku kopa, kuri savieno dotos punktus:

$$\begin{aligned} U=\{&(1,1), (1,2), (1,3), (1,5), (1,6), (2,1),\\ &(2,3), (3,1), (3,2), (3,5), (4,1), (6,3),\\ &(6,4), (6,5), (6,6)\}. \end{aligned} \tag{7.1.2}$$

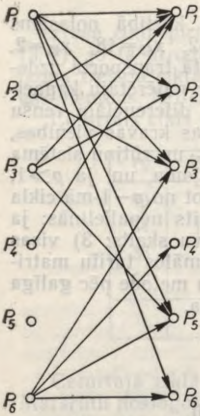
D e f i n ī c i j a 1°. *Par grafu sauc doto punktu un orientētu loku kopu*

$$G=(X, U).$$

Grafu var attēlot grafiski, patvaļīgi izvietojot dotos punktus plaknē (vai telpā) un savienojot tos ar jebkuras formas un garuma lokiem. Tā, piemēram, grafu, kuru veido punktu kopa (7.1.1) un loku kopa (7.1.2), var attēlot plaknē tā, kā parādīts zīmējumā 7.1, vai arī tā, kā parādīts zīmējumā 7.2. Punktu kopas X elementus P_i $(i=1, 2, \ldots)$ sauc par grafa *virsotnēm* (vai *mezgliem*). Ja vir-



Zīm. 7.1.



Zīm. 7.2.

sotņu skaits ir galīgs naturāls skaitlis, tad grafu sauc par *galīgu*. Grafu var analītiski definēt ar tā *virsotņu saistības (incidences) matricu*. Aplūkojamā piemērā dotā grafa virsotņu saistības matrica ir

$$V=\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Virsotņu saistības matrica ir kvadrātiska matrica, kuras rindu un kolonnu skaits (matricas kārta) ir vienāds ar virsotņu skaitu. Tās elements $a_{ij}=0$, ja starp virsotnēm P_i un P_j nav orientēta loka, un $a_{ij}=1$, ja virsotni P_i un virsotni P_j savieno orientēts loks (i, j).

Grafu var definēt arī ar *loku saistības (incidences) matricu*. Aplūkojamā piemērā dotā grafa *loku saistības matrica* ir šāda:

$$L=$$
$$=\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & -1 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & -1 & 1 & 1 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Loku saistības matricai $\boldsymbol{L}=(l_{rk})_{(m\times n)}$ ir tik rindu, cik grafam virsotņu, un tik kolonnu, cik grafam loku. Matricas elements l_{rk} ir 1, -1 vai 0 atkarībā no tā, vai virsotne P_i ir attiecīgi loka (i, j) sākums, gals vai arī šī virsotne nepieder pie loka.

Ja divas grafa virsotnes P_i un P_j savieno loks, kura sākums ir jebkura no šīm divām virsotnēm, tad saka, ka starp šīm virsotnēm ir *šķautne*. Divas virsotnes sauc par *blakus virsotnēm*, ja starp tām ir šķautne. Tādēļ šķautnes jēdziens ir ekvivalents ar *neorientēta grafa* jēdzienu. Tā, piemēram, starp virsotnēm P_1 un P_2 zīmējumā 7.1 ir šķautne. Šķautne ir arī, piemēram, starp virsotnēm P_4 un P_6, bet tās nav, piemēram, starp virsotnēm P_3 un P_4.

Divus lokus sauc par *blakus lokiem*, ja tiem ir kopīga virsotne.

Šķautņu virkni, kuras katram loceklim ir kopīga virsotne ar nākošo locekli, sauc par *ķēdi*. Loku virkni, kurā katra loka galapunkts ir tam sekojošā loka sākumpunkts, sauc par *ceļu*. Katrs ceļš ir ķēde, bet ne katra ķēde ir ceļš. Aplūkojamā piemērā $\{(1,2),$

(2, 3), (3, 5), (6, 5)} ir ķēde, bet nav ceļš. Ceļš ir, piemēram, {(1, 2), (2, 3), (3, 5)}, un tā ir arī ķēde.

Noslēgtu ķēdi sauc par *ciklu,* bet noslēgtu ceļu — par *kontūru.* Katra kontūra ir cikls, bet ne katrs cikls ir kontūra. Jēdzieni *škautne, ķēde* un *cikls* tātad atšķiras no jēdzieniem *loks, ceļš* un *kontūra* tikai ar to, ka pēdējiem ņem vērā orientāciju.

Par *ceļa (vai kontūras) garumu* sauc tajā iesaistīto loku skaitu. *Cilpa* ir kontūra, kuras garums ir 1. Cilpa tātad ir loks, kura gals sakrīt ar sākumu. Tā, piemēram, zīmējumā 7.1 un 7.2 cilpas ir loki (1, 1) un (6, 6).

Grafu sauc par sakarīgu, ja starp jebkurām divām tā virsotnēm eksistē vismaz viena ķēde. Tāds, piemēram, ir zīmējumā 7.1 attēlotais grafs. Grafu sauc par *cieši sakarīgu,* ja starp katrām divām tā virsotnēm eksistē vismaz viens ceļš. Zīmējumā 7.1 attēlotais grafs nav cieši sakarīgs, jo nav ceļa, kas savienotu virsotni P_5 ar kādu citu šī grafa virsotni.

Simetriska grafa gadījumā katras divas blakus virsotnes P_i un P_j savieno divi savstarpēji pretimvērsti loki (i, j) un (j, i). Ja katras divas blakus virsotnes P_i un P_j savieno tikai viens no diviem iespējamiem, savstarpēji pretimvērstiem lokiem (i, j) vai (j, i), tad grafu sauc par *antisimetrisku.* Acīm redzami katrs grafs, kurā nav nevienas kontūras, ir antisimetrisks.

Galīgu, sakarīgu grafu, kuram ir vismaz divas virsotnes un nav ciklu, sauc par *koku.* Tas ir grafs, kuram ir m virsotnes un $m-1$ loks.

Pieņemsim, ka sakarīgā grafā, kurā nav kontūru, starp virsotnēm P_i un P_j ir ceļš, kura garums nav nulle. Tad virsotni P_i sauc par *priekšteci* un P_j — par *pēcteci.* Pēctecībai ir divas raksturīgas īpašības:

1) pēctecība ir antisimetriska — ja P_i ir P_j priekštecis, tad P_j nevar būt P_i priekštecis;

2) pēctecība ir transitīva — ja P_j ir P_i pēctecis un P_k ir P_j pēctecis, tad P_k ir arī P_i pēctecis. Tas nozīmē, ka katrā sakarīgā, galīgā antisimetriskā grafā var visas virsotnes sanumurēt to pēctecības kārtībā.

Galīgu, sakarīgu antisimetrisku grafu sauc par *tīklu,* ja

1) grafam ir viena un tikai viena virsotne (*avots jeb ieeja*), kurai nav priekšteču;

2) grafam ir viena un tikai viena virsotne (*satece jeb izeja*), kurai nav pēcteču;

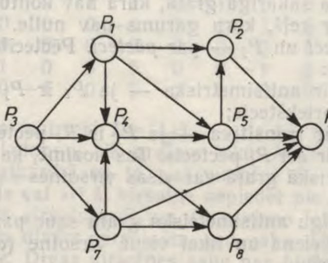
3) katram grafa lokam atbilst noteikts nenegatīvs skaitlis t_{ij}, kuru sauc par šī loka *caurlaides spēju.*

Grafu vai tīklu sauc par *sakārtotu,* ja tā virsotnes sanumurētas to pēctecības kārtībā.

7.2. §. Sakārtota grafa virsotņu numerācija. Ievērojot pēctecību, visas galīga, sakarīga, antisimetriska grafa virsotnes var sadalīt *kārtās.* Vienā kārtā ietilpstošās virsotnes savā starpā nesaista neviens šāda grafa loks. Nulltajā kārtā ietilpst tās virsotnes, kurām nav pēcteču. Tīkla gadījumā tāda virsotne (izeja) ir viena vienīga. Ja nosvītro visus tos lokus, kuru galapunkti ir nulltās kārtas virsotnes, tad izveidojas tādas virsotnes, kam nav pēcteču. Tās ir 1. kārtas virsotnes. 1. kārtas virsotnes tātad ir nulltās kārtas virsotnēm blakus esošie priekšteči. Nosvītrojot visus lokus, kuru galapunkti ir 1. kārtas virsotnes, atklājas 2. kārtas virsotnes utt. Šādu nosvītrošanu turpina tik ilgi, kamēr atklājas virsotnes, kurām nav neviena priekšteča. Tās ir grafa visvecākās kārtas virsotnes, kuras mēdz saukt arī par nulltā *ranga* virsotnēm. Tīklam ir viena un tikai viena pēdējās kārtas vai nulltā ranga virsotne.

Visas galīga, sakarīga, antisimetriska grafa virsotnes var sakārtot arī pēc to rangiem. Nulltā ranga virsotnes ir tās, kurām nav priekšteču. Nosvītrojot lokus, kuru sākumpunkti ir nulltā ranga virsotnes, izveidojas tādas virsotnes, kam nav priekšteču. Tās ir 1. ranga virsotnes. 1. ranga virsotnes tātad ir nulltā ranga virsotnēm blakus esošie pēcteči. Nosvītrojot visus lokus, kuru sākumpunkti ir 1. ranga virsotnes, atklājas 2. ranga virsotnes utt. Šādu nosvītrošanas procesu turpina tik ilgi, kamēr atklājas virsotnes, kurām nav vairs neviena pēcteča. Tās ir visaugstākā ranga, resp., nulltās kārtas virsotnes.

Zīmējumā 7.3 attēlotā grafa virsotnes pēc to rangiem sadalās šādi: nulltais rangs — P_3; 1. rangs — P_1, P_7; 2. rangs — P_4; 3. rangs — P_5, P_8; 4. rangs — P_2; 5. rangs — P_6.



Zīm. 7.3.

Grafu teorijas praktiskie pielietojumi plānošanā daudzos gadījumos vienkāršojas, ja dotā grafa virsotnes numurē, tās iepriekš sakārtojot pēc to rangiem vai kārtām. Tādēļ aplūkosim vienu no iespējamiem analītiskajiem algoritmiem virsotņu sakārtošanai pēc to kārtas vai ranga, pieņemot, ka antisimetrisks grafs ir dots ar tā virsotņu saistības matricu (kas atbilst zīmējumā 7.3 attēlotajam grafam).

Tam nolūkam izmanto matricu $\boldsymbol{V}$, kā parādīts tabulā (7.2.1), pierakstot pie katras šīs matricas rindas $\boldsymbol{v}^i$ un katras kolonnas $\boldsymbol{v}_j$ attiecīgo virsotņu P_i resp. P_j apzīmējumus $P_1, P_2, \ldots, P_8$:

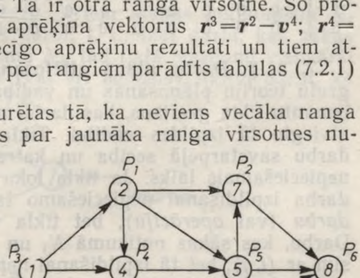
	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	Rangs	Virsotnes
P_1	0	1	0	1	1	0	0	0		
P_2	0	0	0	0	0	1	0	0		
P_3	1	0	0	1	0	0	1	0		
P_4	0	0	0	0	1	0	0	1		
P_5	0	1	0	0	0	1	0	0		
P_6	0	0	0	0	0	0	0	0		
P_7	0	0	0	1	0	1	0	1		
P_8	0	0	0	0	0	1	0	0		
$\boldsymbol{r}^0$	1	2	0	3	2	4	1	2	0	P_3
$\boldsymbol{r}^1$	0	2	*	2	2	4	0	2	1	P_1, P_7
$\boldsymbol{r}^2$	*	1	*	0	1	3	*	1	2	P_4
$\boldsymbol{r}^3$	*	1	*	*	0	3	*	0	3	P_5, P_8
$\boldsymbol{r}^4$	*	0	*	*	*	1	*	*	4	P_2
$\boldsymbol{r}^5$	*	*	*	*	*	0	*	*	5	P_6

(7.2.1)

Sastādot visu matricas $\boldsymbol{V}$ rindu — vektoru $\boldsymbol{v}^i$ (sk. 9. 1. §) summu $\boldsymbol{r}^0 = \sum_i \boldsymbol{v}^i$, iegūst vektoru $\boldsymbol{r}^0 = (1, 2, 0, 3, 2, 4, 1, 2)$, kura trešā koordināte ir 0. Tas nozīmē, ka P_3 ir nulltā ranga virsotne. Pēc tam aprēķina vektora $\boldsymbol{r}^1 = \boldsymbol{r}^0 - \boldsymbol{v}^3$ nenegatīvās koordinātes, starp kurām parādās divas jaunas nulles, kas atbilst virsotnēm P_1 un P_7. Tās ir pirmā ranga virsotnes. Nākošajā solī aprēķina vektora $\boldsymbol{r}^2 = \boldsymbol{r}^1 - \boldsymbol{v}^1 - \boldsymbol{v}^7$ nenegatīvās koordinātes, starp kurām ir viena jauna nulle, kas atbilst virsotnei P_4. Tā ir otrā ranga virsotne. Šo procesu soli pa solim turpinot, aprēķina vektorus $\boldsymbol{r}^3 = \boldsymbol{r}^2 - \boldsymbol{v}^4$; $\boldsymbol{r}^4 = \boldsymbol{r}^3 - \boldsymbol{v}^5 - \boldsymbol{v}^8$; $\boldsymbol{r}^5 = \boldsymbol{r}^4 - \boldsymbol{v}^2$. Attiecīgo aprēķinu rezultāti un tiem atbilstošais virsotņu sadalījums pēc rangiem parādīts tabulas (7.2.1) pagarinājumā.

Ja grafa virsotnes sanumurētas tā, ka neviens vecāka ranga virsotnes numurs nav mazāks par jaunāka ranga virsotnes numuru, tad grafu var saukt par sakārtotu. Viena un tā paša ranga ietvaros virsotnes var sanumurēt patvaļīgi.

Sanumurējot zīmējumā 7.3 attēlotā grafa virsotnes to rangu kārtībā, iegūst jaunu, sakārtotu grafu, kas attēlots zīmējumā 7.4. Tā virsotnes apzīmētas ar kārtas skaitļiem 1, 2, . . . , 8, kas ierakstīti aplīšos. Salīdzināša-



Zīm. 7.4.

nai zīmējumā 7.4 blakus jaunajiem virsotņu apzīmējumiem attēloti arī sākotnējie nesakārtotā grafa virsotņu apzīmējumi $P_1, P_2, \ldots, P_8$.

Sakārtota antisimetriska grafa virsotņu saistības matrica ir augšējā trīsstūrmatrica, t. i., tāda kvadrātiska matrica, kuras visi elementi, kas atrodas zem galvenās diagonāles, ir vienādi ar nulli. Zīmējumā 7.4 attēlotā grafa virsotņu saistības matrica pierakstīta tabulā (7.2.2):

	1	2	3	4	5	6	7	8
1	0	1	1	1	0	0	0	0
2	0	0	0	1	1	0	1	0
3	0	0	0	1	0	1	0	1
4	0	0	0	0	1	1	0	0
5	0	0	0	0	0	0	1	1
6	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0

(7.2.2)

Attiecīgo loku saistības matricu var pierakstīt tā, kā parādīts tabulā (7.2.3):

	1 2	1 3	1 4	2 4	2 5	2 7	3 4	3 6	3 8	4 5	4 6	5 7	5 8	6 8	7 8
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
2	−1	0	0	1	1	1	0	0	0	0	0	0	0	0	0
3	0	−1	0	0	0	0	1	1	1	0	0	0	0	0	0
4	0	0	−1	−1	0	0	−1	0	0	1	1	0	0	0	0
5	0	0	0	0	−1	0	0	0	0	−1	0	1	1	0	0
6	0	0	0	0	0	0	0	−1	0	0	−1	0	0	1	0
7	0	0	0	0	0	−1	0	0	0	0	0	−1	0	0	1
8	0	0	0	0	0	0	0	0	−1	0	0	0	−1	−1	−1

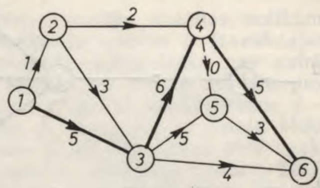
(7.2.3)

7. 3. §. Darbu plāna attēlošana ar tīkla grafiku. Pielietojot grafu teoriju plānošanas un vadības jautājumu risināšanā, bieži izmanto tīkla grafikus, kas darbu plānu padara pārskatāmāku un atvieglo tā izpildes vadību. Tīkla sastādīšanai jāzina veicamo darbu savstarpējā secība un katra atsevišķa darba izpildīšanai nepieciešamais laiks. Ja tīkla loku caurlaides spēju identificē ar darba izpildīšanai nepieciešamo laiku, tad tīkla loku sauc par *darbu* (vai *operāciju*), bet tīkla virsotnes — par *notikumiem*. Darbu, kas sākas notikumā N_i un beidzas notikumā N_j, apzīmēsim ar (i, j), bet tā izpildīšanai nepieciešamo laiku — ar t_{ij}.

Ja darba (i, j) visgarākais jeb t. s. *pesimistiskais laiks* ir a_{ij}, visīsākais jeb *optimistiskais laiks* ir b_{ij} un visvarbūtīgākais laiks ir m_{ij}, tad, veidojot tīkla grafiku, laikus t_{ij} visiem i un j aprēķina pēc formulas

$$t_{ij}=\frac{1}{6}\,(a_{ij}+4m_{ij}+b_{ij}). \qquad (7.3.1)$$

Pieņemsim, ka 10 dažādu darbu komplekss attēlots ar sakārtotu tīkla grafiku, kas redzams zīmējumā 7.5. Tīkla loki šajā gadījumā nozīmē atsevišķus darbus, bet loku caurlaides spējas — darba laikus, kas zīmējumā 7.5 atzīmēti pie katra loka (darba).



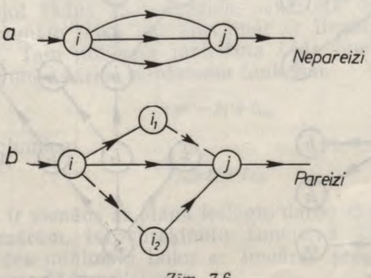
Zīm. 7.5.

Jēdzienā *darbs* ietilpst katrs process, kam vajadzīgs zināms laiks un resursi. Šajā jēdzienā ietilpst arī *gaidīšanas laiki,* kas neprasa materiālus resursus. Var būt arī *fiktīvie darbi,* kas neprasa nedz materiālus resursus, nedz laiku, bet norāda secību, kurā kāds darbs var sekot aiz cita darba. Fiktīvs, piemēram, ir darbs (4, 5) zīmējumā 7.5. Tas nozīmē, ka darbu (5, 6) nevar sākt, iekāms nav nobeigti visi darbi, kuru noslēgumā realizējas notikums N_4. Fiktīvos darbus, kuru laiks vienāds ar 0, tīkla grafikā parasti attēlo ar svītrlīnijām.

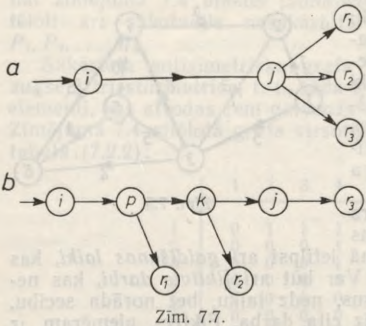
Notikums atšķirībā no darba nav process, kam vajadzīgs laiks un materiāli resursi. Tas ir moments, kurā sākas vai beidzas kāds atsevišķs darbs vai darbu grupa.

Izšķir *vienkāršus* un *mezgla notikumus.* Vienkāršs ir tāds notikums, kurā nebeidzas neviens darbs vai arī beidzas tikai viens darbs un tūdaļ pēc tam sākas arī tikai viens darbs. Mezgla notikumi ir visi tie notikumi, kas nav vienkārši.

1°. Paralēli darbi. Pareizi veidotā tīklā nevar būt kontūru. Katrs atsevišķs darbs (loks) var savienot divus un tikai divus blakus notikumus. Ja starp diviem blakus notikumiem N_i un N_j veicami vairāki paralēli darbi, t. i., tādi darbi, kas sākas notikumā N_i un beidzas notikumā N_j, tad tīklā ievedami vienkārši papildu notikumi, kuros sākas (vai arī beidzas) fiktīvi darbi, kā parādīts, piemēram, zīmējumā 7.6.



Zīm. 7.6.

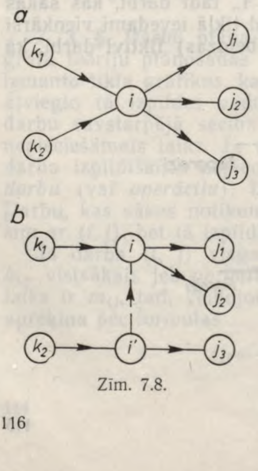


Zīm. 7.7.

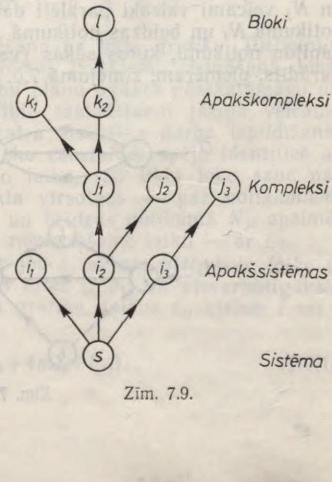
2°. Viena darba vairāki posmi. Ja kādam darbam (i, j) ir vairāki cits citam sekojoši posmi (i, p), (p, k), (k, j) un darba (j, r_1) uzsākšanu netraucē iepriekšējā darba (i, j) posmi (p, k) un (k, j), bet darba (j, r_2) uzsākšanu netraucē posms (k, j), tad darbs (i, j) sadalāms ar papildu notikumiem, kā parādīts zīmējumā 7.7.

3°. Neatkarīgi darbi. Ja N_i ir mezgla notikums, kurā beidzas vairāk nekā viens darbs un daži darbi pēc šī notikuma nav atkarīgi no notikuma N_i iestāšanās, tas attēlojams ar papildu notikuma un fiktīva darba palīdzību, kā parādīts, piemēram, zīmējumā 7.8.

Acīmredzot ievērojamas grūtības tīkla sastādīšanā rada atsevišķu darbu savstarpējās secības un sakarības izveidošana, t. i., notikumu saistības matricas sastādīšana. Šī uzdevuma veikšana var saistīties ar īpašiem pētījumiem, jo praksē vienai un tai pašai darbu sistēmai var būt daudz tīklveida plānošanas modeļu un tie var būt dažādi veidoti. Lielas darbu sistēmas tīklveida plānošanas modeļa sastādīšanu parasti sāk ar uzdevuma formulēšanu un plāna struktūras koka sastādīšanu. Plāna struktūras koka principiālās shēmas piemērs parādīts zīmējumā 7.9.



Zīm. 7.8.



Zīm. 7.9.

Sākotnējai notikumu numerācijai pastāv dažādas notikumu kodēšanas sistēmas. To izvēlē izšķiroša nozīme ir izveidotajam plāna struktūras kokam. Pieņemot, ka katrā blokā nav vairāk kā 99 notikumi, zīmējumā 7.9 attēlotajam kokam atbilstu pieczīmīgi notikumu numuri *ijkll*, kur *i* apzīmē apakšsistēmu, *j* — kompleksu, *k* — apakškompleksu, *ll* — notikumu dotajā blokā. Tā, piemēram, otrās apakšsistēmas pirmā kompleksa otrā apakškompleksa 76. notikums apzīmējams ar sākotnējo numuru 21276. Analogi numurs 10032 nozīmē pirmās apakšsistēmas 32. notikumu utt.

7.4. §. Tīklveida plāna izpildes dekretētais laiks, minimālais laiks un kritiskais ceļš. Katra sastādīta darbu plāna izpildīšanai nepieciešams noteikts laiks. Izšķir plāna izpildes *dekretēto laiku* un *minimālo laiku*. Par dekretēto laiku sauc to plāna izpildes laiku, kura sākuma un beigu momenti noteikti ar augstāku vadības orgānu rīkojumiem. Reālos apstākļos dekretētais laiks nevar būt mazāks par minimālo laiku. Par minimālo laiku sauc to darbu laiku summu, kas nepieciešama, lai paveiktu visus darbus, kuri veido *kritisko ceļu*. Par kritisko sauc tādu ceļu, kas sākas pirmajā notikumā (tīkla ieejā) N_1 un beidzas noslēguma notikumā (tīkla izejā) N_m, ja visu uz šī ceļa esošo darbu laiku summa ir vislielākā salīdzinājumā ar katru citu ceļu starp notikumiem N_1 un N_m. Visus darbus, kuri atrodas uz kritiskā ceļa, sauc par *kritiskajiem darbiem*.

Tātad kritisko ceļu veido tikai kritiskie darbi. Ja uz kāda ceļa atrodas kaut viens *nekritisks darbs*, tad ceļš nav kritisks.

Katra sakārtota darbu plāna izpildīšanas minimālo laiku un kritisko ceļu var noteikt, atrisinot relatīvi vienkāršu uzdevumu. Lai formulētu šo uzdevumu, apzīmēsim ar t_i laika momentu, kurā nav palicis neizpildīts neviens darbs, kas beidzas notikumā N_i $(i=1, 2, \ldots, m)$. Šajā momentā var uzsākt jebkuru darbu (i, j), kurš sākas notikumā N_i, beidzas notikumā N_j un kura ilgums ir t_{ij}. Laika momentu t_i sauksim par notikuma N_i *visagrāko iestāšanās laiku*.

Izmantojot šādus apzīmējumus, sakārtota darbu plāna izpildīšanas minimālo laiku var aprēķināt ar lineārās programmēšanas metodi. Tam nolūkam jāatrisina šāds lineārās programmēšanas uzdevums. Atrast minimumu funkcijai.

$$\tau=-t_1+t_m \tag{7.4.1}$$

pie ierobežojumiem

$$t_j \geqslant t_i+t_{ij}, \tag{7.4.2}$$

kuru skaits ir vienāds ar plānā ieslēgto darbu skaitu.

Tā, piemēram, lai aprēķinātu zīmējumā 7.5 attēlotā darbu plāna izpildes minimālo laiku ar lineārās programmēšanas metodi, jāatrisina šāds uzdevums.

Atrast minimumu lineārai funkcijai.

$$\tau = -t_1 + t_6 \tag{7.4.3}$$

pie ierobežojumiem

$$\left.\begin{array}{ll} t_2 \geqslant t_1 + 1, & t_5 \geqslant t_3 + 5, \\ t_3 \geqslant t_1 + 5, & t_5 \geqslant t_4 + 0, \\ t_3 \geqslant t_2 + 3, & t_6 \geqslant t_3 + 4, \\ t_4 \geqslant t_2 + 2, & t_6 \geqslant t_4 + 5, \\ t_4 \geqslant t_3 + 6, & t_6 \geqslant t_5 + 3. \end{array}\right\} \tag{7.4.4}$$

Tā kā uzdevuma (7.4.3), (7.4.4) mainīgajiem lielumiem t_i nav zīmes ierobežojumu, tad visi duālā uzdevuma ierobežojumi ir vienādojumi un attiecīgo duālo uzdevumu var formulēt šādi.

Atrast maksimumu lineārai funkcijai

$$\begin{aligned} z = x_{12} + 5x_{13} + 3x_{23} + 2x_{24} + 6x_{34} + 5x_{35} + \\ + 0 \cdot x_{45} + 4x_{36} + 5x_{46} + 3x_{56} \end{aligned} \tag{7.4.5}$$

pie ierobežojumiem

$$\left.\begin{array}{l} x_{12} + x_{13} = 1, \\ -x_{12} + x_{23} + x_{24} = 0, \\ -x_{13} - x_{23} + x_{34} + x_{35} + x_{36} = 0, \\ -x_{24} - x_{34} + x_{45} + x_{46} = 0, \\ -x_{35} - x_{45} + x_{56} = 0, \\ -x_{36} - x_{46} - x_{56} = -1, \end{array}\right\} \tag{7.4.6}$$

$$x_{ij} \geqslant 0 \tag{7.4.7}$$

$$(i = 1, 2, \ldots, 5;\ j = 2, 3, \ldots, 6).$$

Ierobežojumu sistēmā ir tik vienādojumu, cik virsotņu (notikumu) ir tīklā, un tik nezināmo, cik loku (darbu) ir šajā tīklā. Ierobežojumu sistēmas (7.4.6) koeficientu matrica ir sakārtotā tīkla loku (darbu) saistības matrica. Tas uzskatāmi redzams uzdevuma (7.4.5) – (7.4.7) pirmajā iterācijas tabulā (7.4.8):

+1	−1	$\begin{smallmatrix}1\\2\end{smallmatrix}$	$\begin{smallmatrix}1\\3\end{smallmatrix}$	$\begin{smallmatrix}2\\3\end{smallmatrix}$	$\begin{smallmatrix}2\\4\end{smallmatrix}$	$\begin{smallmatrix}3\\4\end{smallmatrix}$	$\begin{smallmatrix}3\\5\end{smallmatrix}$	$\begin{smallmatrix}4\\5\end{smallmatrix}$	$\begin{smallmatrix}3\\6\end{smallmatrix}$	$\begin{smallmatrix}4\\6\end{smallmatrix}$	$\begin{smallmatrix}5\\6\end{smallmatrix}$	(I)
z	0	−1	−5	−3	−2	−6	−5	0	−4	−5	−3	0
v_1	1	1	1									1
v_2	0	−1		1	1							2
v_3	0		−1	−1		1	1		1			3
v_4	0				−1	−1		1		1		4
v_5	0						−1	−1			1	5
v_6	−1								−1	−1	−1	6

(7.4.8)

Tabulā (7.4.8) nezināmo x_{ij} vietā divās rindās ierakstīti šo nezināmo indeksi. Pirmais indekss ir darba (i, j) sākuma notikuma, bet otrais — beigu notikuma numurs. Ar v_i $(i=1, 2, \ldots, 6)$ tabulā (7.4.8) apzīmēti mākslīgie bāzes mainīgie. Izpildot piecus simpleksa metodes iterāciju soļus, iegūst tabulā (7.4.8) dotā lineārās programmēšanas uzdevuma optimālo atrisinājumu, kas nolasāms tabulā (7.4.9):

1	-1	x_{12}	x_{24}	x_{35}	x_{45}	x_{36}	(VI)
z	16	1	7	3	2	7	0
x_{13}	1	1					1
x_{23}	0	-1	1				2
x_{34}	1		1	1		1	3
x_{46}	1			1	1	1	4
x_{56}	0			-1	-1		5
v_6	0						6

(7.4.9)

Tā kā mērķa funkcijas z maksimālā vērtība $z=16$ ir vienāda ar duālā uzdevuma (7.4.3), (7.4.4) mērķa funkcijas τ minimālo vērtību, tad dotā darbu plāna izpildes minimālais laiks ir $\tau=16$. Tāds laiks ir nepieciešams, lai paveiktu darbus (1, 3), (3, 4) un (4, 6), kuri veido šī darbu plāna kritisko ceļu.

7.5. §. Tīklveida plāna parametri un to atrašanas algoritmi. Iepriekšējā paragrāfā redzējām, ka tīklveida plāna izpildes minimālo laiku un kritisko ceļu var atrast ar simpleksa metodes algoritmu. Šis algoritms tomēr nav praktiski izdevīgs tīklveida plāna parametru noteikšanai. Tas kļūst acīm redzams, ja aplūkojam uzdevumu (7.4.1), (7.4.2) un tā konkrēto piemēru (7.4.3), (7.4.4).

Darba plāna pirmā notikuma *visagrākās iestāšanās laiku* t_1 var vienmēr pieņemt par laika skaitīšanas sākuma momentu, t. i.,

$$t_1=0. \tag{7.5.1}$$

Tad vispārīgā gadījumā $\tau=t_m$, t. i., *pēdējā notikuma visagrākās iestāšanās laiks ir vienāds ar visa darbu plāna izpildes minimālo laiku* un

$$t_j=\max_{(i,\, j)\in U^-_j}(t_i+t_{ij}), \tag{7.5.2}$$

kur (i, j) ir loks (darbs), kas sākas virsotnē (notikumā) N_i un beidzas virsotnē (notikumā) N_j, bet U_j^- ir to loku (darbu) apakškopa, kuri beidzas notikumā N_j.

Pielietojot šādu algoritmu (7.5.1)—(7.5.2) uzdevumam (7.4.3), (7.4.4), iegūstam

$$\left.\begin{aligned} t_1&=0,\\ t_2&=\max(0+1)=1,\\ t_3&=\max(0+5;\ 1+3)=5,\\ t_4&=\max(1+2;\ 5+6)=11,\\ t_5&=\max(5+5;\ 11+0)=11,\\ t_6&=\max(5+4;\ 11+5;\ 11+3)=16. \end{aligned}\right\} \quad (7.5.3)$$

Tātad konkrētajā piemērā pēdējā notikuma N_6 visagrākās iestāšanās laiks ir $t_6=16$. Tas arī ir šī darbu plāna izpildīšanas minimālais laiks $\tau=16$.

Lai noskaidrotu, kuri no darbu plānā iesaistītajiem darbiem ir kritiski, jāņem vērā, ka kritisko darbu laiku summa ir vienāda ar visa plāna minimālo izpildīšanas laiku τ un katra tāda notikuma aizkavēšanās, kurš atrodas uz kritiskā ceļa, aizkavē plāna izpildīšanu, attiecīgi palielinot τ. Notikumus, kuri atrodas uz kritiskā ceļa, sauc par *kritiskajiem notikumiem*. Kritiskā notikuma N_i visagrākās iestāšanās laiks t_i ir vienāds ar šī notikuma *visvēlākās iestāšanās laiku* t^*_i.

Nekritiskā notikuma visagrākās iestāšanās laiks nesakrīt ar tā visvēlākās iestāšanās laiku. Par nekritiska notikuma N_j *visvēlākās iestāšanās laiku* t^*_j sauc to laika momentu, kurā obligāti jāuzsāk visi tie darbi, kas sākas notikumā N_j. Katra notikuma N_j aizkavēšanās aiz tā visvēlākās iestāšanās laika t^*_j aizkavē visa plāna izpildīšanu un attiecīgi pagarina minimālo plāna izpildīšanas laiku τ.

Laika intervālu

$$R_j=t^*_j-t_j, \quad (7.5.4)$$

t. i., starpību starp nekritiskā notikuma N_j visvēlākās un visagrākās iestāšanās laiku, sauc par notikuma N_j *iestāšanās laika rezervi*. Kritiskā ceļa notikumiem iestāšanās laika rezerve ir vienāda ar nulli.

Darbu plāna pirmais notikums N_1 un pēdējais notikums N_m vienmēr ir kritiski notikumi. Tādēļ

$$t_1=t^*_1=0, \quad (7.5.5)$$

$$t_m=t^*_m=\tau \quad (7.5.6)$$

un

$$t^*_i=\min_{(i,\,j)\in U^+_i}(t^*_j-t_{ij}), \quad (7.5.7)$$

kur (i, j) ir darbs, kas sākas notikumā N_i un beidzas notikumā N_j, bet U_i^+ ir to darbu apakškopa, kuri sākas notikumā N_i.

Pielietojot algoritmu (7.5.5)—(7.5.7) uzdevumam (7.4.3), (7.4.4), iegūstam

$$\left.\begin{aligned}
t^*_6 &= 16,\\
t^*_5 &= \min\,(16-3) = 13,\\
t^*_4 &= \min\,(16-5;\ 13-0) = 11,\\
t^*_3 &= \min\,(16-4;\ 13-5;\ 11-6) = 5,\\
t^*_2 &= \min\,(11-2;\ 5-3) = 2,\\
t^*_1 &= \min\,(\ 5-5;\ 2-1) = 0.
\end{aligned}\right\} \tag{7.5.8}$$

Ievietojot skaitļus (7.5.8) un (7.5.3) formulās (7.5.4), atrodam visu aplūkojamā piemēra notikumu iestāšanās laika rezerves: $R_1 = R_3 = R_4 = R_6 = 0$, bet $R_2 = 2-1 = 1$ un $R_5 = 13-11 = 2$. Tas nozīmē, ka zīmējumā 7.5 attēlotā darbu plāna kritiskie notikumi ir N_1, N_3, N_4 un N_6, kritiskais ceļš iet caur šiem notikumiem un kritiskie darbi ir (1, 3), (3, 4) un (4, 6). Visi pārējie 7 darbi ir nekritiski. Nekritiski ir arī notikumi N_2 un N_5.

Bez minimālā laika τ un notikuma N_i iestāšanās laika rezerves par tīkla parametriem sauc

1) *darba* (i, j) *agrāko sākumu*

$$t_{ij}^{as} = t_i, \tag{7.5.9}$$

2) *darba* (i, j) *agrākās beigas*

$$t_{ij}^{ab} = t_i + t_{ij}, \tag{7.5.10}$$

3) *darba* (i, j) *vēlākās beigas*

$$t_{ij}^{vb} = t^*_j, \tag{7.5.11}$$

4) *darba* (i, j) *vēlāko sākumu*

$$t_{ij}^{vs} = t^*_j - t_{ij}, \tag{7.5.12}$$

kur t_i — notikuma N_i visagrākās iestāšanās laiks, t^*_j — notikuma N_j visvēlākās iestāšanās laiks un t_{ij} — darba (i, j) izpildīšanas laiks.

Svarīgi tīklveida plāna parametri ir *darbu laiku rezerves*, kuras definēsim ar šādām formulām:

1) *darba* (i, j) *pilnā laika rezerve*

$$R_{ij}^{p} = t^*_j - t_i - t_{ij} \tag{7.5.13}$$

jeb

$$R_{ij}^{p} = t_{ij}^{vb} - t_{ij}^{ab}, \tag{7.5.14}$$

jeb

$$R_{ij}^{p} = t_{ij}^{vs} - t_{ij}^{as}; \tag{7.5.15}$$

2) *darba* (i, j) *brīvā laika rezerve*

$$R_{ij}{}^{\mathrm{b}} = t_j - t_i - t_{ij} \tag{7.5.16}$$

jeb

$$R_{ij}{}^{\mathrm{b}} = t_j - t_{ij}{}^{\mathrm{ab}}; \tag{7.5.17}$$

3) *darba* (i, j) *neatkarīgā laika rezerve*

$$R_{ij}{}^{\mathrm{n}} = \max\,[0;\ t_j - t^*_i - t_{ij}]. \tag{7.5.18}$$

Izmantojot formulas (7.5.9)—(7.5.18), tabulā (7.5.19) apkopoti zīmējumā 7.5 attēlotā tīklveida darbu plāna parametri:

Notikumi N_i	Notikumi N_j	t_{ij}	t_i	t^*_i	t_j	t^*_j	$t_{ij}{}^{\mathrm{ab}}$	$t_{ij}{}^{\mathrm{vs}}$	$R_{ij}{}^{\mathrm{p}}$	$R_{ij}{}^{\mathrm{b}}$	$R_{ij}{}^{\mathrm{s}}$
1	2	1	0	0	1	2	1	1	1	0	0
1	3	5	0	0	5	5	5	0	0	0	0
2	3	3	1	2	5	5	4	2	1	1	0
2	4	2	1	2	11	11	3	9	8	8	7
3	4	6	5	5	11	11	11	5	0	0	0
3	5	5	5	5	11	13	10	8	3	1	1
3	6	4	5	5	16	16	9	12	7	7	7
4	5	0	11	11	11	13	11	13	2	0	0
4	6	5	11	11	16	16	16	11	0	0	0
5	6	3	11	13	16	16	14	13	2	2	0

(7.5.19)

Kritiskajiem darbiem visas laiku rezerves ir vienādas ar nulli.

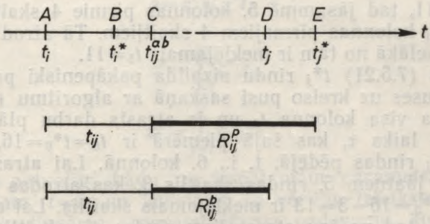
Nekritiska darba (i, j) *pilnā laika rezerve* $R_{ij}{}^{\mathrm{p}}$ ir tas laiks, par kuru var atlikt šī darba uzsākšanu, nepagarinot tīklveida darbu plāna izpildīšanas minimālo laiku τ. Ja izsmelta darba (i, j) visa pilnā laika rezerve, tad šis darbs kļūst kritisks.

Nekritiska darba (i, j) *brīvā laika rezerve* $R_{ij}{}^{\mathrm{b}}$ ir tas laiks, par kuru var atlikt šī darba uzsākšanu, nemainot šim darbam sekojošo darbu agrāko sākumu t_j. Pilnās un brīvās laika rezerves ir vienādas visiem tiem nekritiskajiem darbiem, kuri beidzas kritiskajos notikumos.

Nekritiska darba (i, j) *neatkarīgā laika rezerve* $R_{ij}{}^{\mathrm{n}}$ ir tas laiks, par kuru var atlikt šī darba uzsākšanu, nemainot to darbu (h, i) vēlākās beigas $t^*_i = t_{hi}{}^{\mathrm{vb}}$, kuri beidzas darba (i, j) sākuma notikumā N_i, kā arī nemainot darbam (i, j) sekojošo darbu (j, k) agrāko sākumu $t_j = t_{jk}{}^{\mathrm{as}}$.

Ja ar formulām (7.5.9)—(7.5.12) definētos laika momentus atzīmē kā citu citam sekojošus punktus uz laika ass t (zīm. 7.10), kur nekritiska darba (i, j) izpildīšanai nepieciešamo laiku t_{ij} attēlo orientēts nogrieznis AC, tad darba (i, j) pilno laika rezervi

$R_{ij}{}^p$ attēlo nogrieznis CE, bet brīvo laika rezervi — nogrieznis $CD=CE-DE$. Tā ir diference starp darba (i, j) beigu notikuma N_j visvēlākās iestāšanās laiku t^*_j un šī notikuma visagrākās iestāšanās laiku t_j, jo no definīcijas formulām (7.5.13) un (7.5.16) izriet, ka $R_{ij}{}^p-R_{ij}{}^b=t^*_j-t_j$. Kritiskajiem darbiem šī diference ir vienāda ar nulli.



Zīm. 7.10.

Sakārtota tīkla parametru aprēķināšanai ērti izmantojama darbu laiku saistības matrica. Par *darbu laiku saistības matricu* sauksim notikumu (tīkla virsotņu) saistības matricu, kurā visi tie elementi a_{ij}, kas vienādi ar 1, atvietoti ar attiecīgajiem darbu laikiem t_{ij}. Tos darbu laikus, kas vienādi ar 0, darbu laiku saistības matricā apzīmēsim ar $\varnothing$. Tā, piemēram, zīmējumā 7.5. attēlotā tīkla darbu laiku saistības matrica ir

$$T=\begin{bmatrix} 0 & 1 & 5 & 0 & 0 & 0 \\ 0 & 0 & 3 & 2 & 0 & 0 \\ 0 & 0 & 0 & 6 & 5 & 4 \\ 0 & 0 & 0 & 0 & \varnothing & 5 \\ 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (7.5.20)$$

Lai aprēķinātu tīkla parametrus, darbu laiku saistības matrica pierakstāma tabulas formā:

	1	2	3	4	5	6	t_i
1	0	1	5	0	0	0	0
2	0	0	3	2	0	0	1
3	0	0	0	6	5	4	5
4	0	0	0	0	$\varnothing$	5	11
5	0	0	0	0	0	3	11
6	0	0	0	0	0	0	16
t^*_j	0	2	5	11	13	16	

(7.5.21)

Tabulas (7.5.21) kolonnu t_i aizpilda pakāpeniski pa rindām no augšas uz leju saskaņā ar algoritmu (7.5.2). Aplūkojamā piemērā šis algoritms realizējas ar formulām (7.5.3). Tā, piemēram, lai atrastu $t_4=11$, kad jau aizpildītas pirmās 3 rindas, tad katrā no šīm rindām jāsummē 4. kolonnā ierakstītais skaitlis ar t_i kolonnā ierakstīto skaitli. Tā atrod summas 0, 3 un 11. Lielākā summa ir meklējamais skaitlis $t_4=11$. Lai atrastu nākošo parametru $t_5=11$, tad jāsummē 5. kolonnas pirmie 4 skaitļi ar atbilstošajiem t_i kolonnas pirmajiem 4 skaitļiem. Tā atrod summas 0, 1, 10, 11. Lielākā no tām ir meklējamais $t_5=11$.

Tabulas (7.5.21) t^*_j rindu aizpilda pakāpeniski pa kolonnām no labās puses uz kreiso pusi saskaņā ar algoritmu (7.5.7). Kad ir aizpildīta visa kolonna t_i un ir atrasts darbu plāna izpildes minimālais laiks τ, kas šajā piemērā ir $t_6=t^*_6=16$, šo skaitli ieraksta t^*_j rindas pēdējā, t. i., 6. kolonnā. Lai atrastu t^*_5, tad no $t^*_6=16$ jāatņem 5. rindas skaitlis 3, kas atrodas 6. kolonnā. Diference $t^*_5=16-3=13$ ir meklējamais skaitlis. Lai atrastu, piemēram, t^*_2, tad no jau atrastajiem rindas t^*_j skaitļiem jāatņem 2. rindas attiecīgajās kolonnās esošie skaitļi. Mazākā no atrastajām diferencēm $16-0$, $13-0$, $11-2$, $5-3$ ir skaitlis $5-3=2$. Šis skaitlis ir meklējamais $t^*_2=2$.

8. nodaļa

PARAMETRISKĀ UN DAĻVEIDA LINEĀRĀ PROGRAMMĒŠANA

8.1. §. Matemātiskās programmēšanas uzdevums, kura lineārās mērķa funkcijas koeficienti ir viena parametra lineāras funkcijas. Pieņemsim, ka vispārīgā formā ir dots šāds matemātiskās programmēšanas uzdevums.

Atrast maksimumu funkcijai

$$z=\sum_{j=1}^{n}(c'_j+c''_j t)x_j \tag{8.1.1}$$

pie ierobežojumiem

$$\sum_{j=1}^{n} a_{ij}x_j\leqslant b_i \quad (i=1,2,\ldots,p<h), \tag{8.1.2}$$

$$\sum_{j=1}^{n} a_{ij}x_j\geqslant b_i \quad (i=p+1,\ldots,h<m), \tag{8.1.3}$$

$$\sum_{j=1}^{n} a_{ij}x_j=b_i \quad (i=h+1,\ldots,m), \tag{8.1.4}$$

$$x_j\geqslant 0 \quad (j=1,2,\ldots,n), \tag{8.1.5}$$

kur a_{ij}, $b_i\geqslant 0$, c'_j un c''_j ir doti reāli skaitļi, bet x_j — nezināmi reāli lielumi un t ir reāls lielums, kas mainās noteiktā intervālā.

Tāds matemātiskās programmēšanas uzdevums ir viens no visvienkāršākajiem nelineārās programmēšanas uzdevumiem. Šo uzdevumu sauksim īsi par *parametriskās programmēšanas uzdevumu.*

Parametriskās programmēšanas uzdevuma mērķa funkciju (8.1.1) vienmēr var izteikt šādi:

$$z = z' + z''t, \tag{8.1.6}$$

kur

$$z' = \sum_{j=1}^{n} c'_j x_j, \tag{8.1.7}$$

$$z'' = \sum_{j=1}^{n} c''_j x_j. \tag{8.1.8}$$

Izvēloties patvaļīgi kādu no iespējamām parametra t vērtībām, piemēram, $t=k$, un ievietojot šo vērtību izteiksmē (8.1.1), iegūst parasto lineārās programmēšanas uzdevumu, kuru formulēsim šādi.

Atrast maksimumu lineārai funkcijai

$$z_k = \sum_{j=1}^{n} q_j x_j \tag{8.1.9}$$

pie ierobežojumiem (8.1.2)—(8.1.5), kur

$$q_j = c'_j + c''_j k \quad (j = 1, 2, \ldots, n). \tag{8.1.10}$$

Tādu uzdevumu var atrisināt ar parasto simpleksa metodi. Šo uzdevumu risinot, ir lietderīgi visās iterāciju tabulās nulltās rindas vietā ievest 3 rindas, kuras apzīmēsim ar 0, 0′ un 0″.

Pirmās iterācijas tabulas rindā 0 raksta funkcijas (8.1.9) koeficientus q_j, rindā 0′ — funkcijas (8.1.7) koeficientus c'_j un rindā 0″ — funkcijas (8.1.8) koeficientus c''_j. Meklējot maksimumu funkcijai (8.1.9), rindās 0, 0′ un 0″ ierakstītos koeficientus pārveido saskaņā ar parasto simpleksa metodes algoritmu. Ja dotā uzdevuma ierobežojumu sistēma nav saderīga, tad parametriskās programmēšanas uzdevumam nav neviena atrisinājuma.

Ja ierobežojumu sistēma ir saderīga un pēdējās iterācijas tabulas rindā 0 visi nebāzes mainīgo koeficienti ir nenegatīvi, tad šīs tabulas brīvo locekļu rindā 0 nolasāms funkcijas (8.1.9), t. i., z_k, maksimums. Šis maksimums ir vienāds ar skaitli $\alpha_0 + \beta_0 k$, kur α_0 nolasāms pēdējās iterācijas tabulas brīvo locekļu rindā 0′, bet β_0 — rindā 0″.

Reizinot visus pēdējās iterācijas tabulas rindas 0″ elementus β_j ar t un pieskaitot šos reizinājumus rindas 0′ elementiem α_j, iegūst pārveidotu funkcijas (8.1.1) izteiksmi

$$z = \alpha_0 + \beta_0 t - \sum_{j=1}^{n} (\alpha_j + \beta_j t) x_j. \tag{8.1.11}$$

No šīs izteiksmes izriet, ka $\alpha_0+\beta_0 t$ ir funkcijas (8.1.1) maksimums pie visām tām parametra t vērtībām, kuras apmierina nosacījumus

$$\alpha_j+\beta_j t\geqslant 0 \quad (j=1, 2, \ldots, n). \tag{8.1.12}$$

Nevienādību sistēmas (8.1.12) atrisinājumu intervāla

$$k_0\leqslant t\leqslant k_1 \tag{8.1.13}$$

galapunktus k_0 un k_1 nosaka, risinot atsevišķi to nevienādību sistēmas (8.1.12) apakšsistēmu, kurā $\beta_j>0$, un apakšsistēmu, kurā $\beta_j<0$.

Ja $\beta_j>0$, tad $t\geqslant -\frac{\alpha_j}{\beta_j}$ un

$$k_0=\max_{\beta_j>0}\left(-\frac{\alpha_j}{\beta_j}\right)=-\frac{\alpha_{j0}}{\beta_{j0}}. \tag{8.1.14}$$

Ja $\beta_j<0$, tad $t\leqslant -\frac{\alpha_j}{\beta_j}$ un

$$k_1=\min_{\beta_j<0}\left(-\frac{\alpha_j}{\beta_j}\right)=-\frac{\alpha_{j1}}{\beta_{j1}}. \tag{8.1.15}$$

Ja nevienādību sistēmā (8.1.12) visi $\beta_j\leqslant 0$, tad t maiņas intervāls uz apakšu nav ierobežots, t. i., $k_0=-\infty$.

Ja turpretim visi $\beta_j\geqslant 0$, tad t maiņas intervāls nav ierobežots uz augšu, t. i., $k_1=+\infty$.

Ar nevienādībām (8.1.13) ir noteiktas robežas, kurās var izmainīties dotās mērķa funkcijas (8.1.1) parametrs t, nemainot fiksētai parametra vērtībai $t=k$ atbilstošo optimālo plānu. Katrai t vērtībai, kas pieder pie intervāla $k_0\leqslant t\leqslant k_1$, protams, atbilst sava maksimālā mērķa funkcijas vērtība

$$\max z=\alpha_0+\beta_0 t. \tag{8.1.16}$$

Ja kādas iterāciju tabulas nulltajā rindā ir negatīvs skaitlis, piemēram, $\alpha_u+\beta_u k<0$, un zem tā esošajā ierobežojumu sistēmas koeficientu kolonnā nav neviena pozitīva skaitļa, tad mērķa funkcija z_k, kas definēta ar izteiksmi (8.1.9), ir neierobežota. Tas tomēr nenozīmē, ka dotā uzdevuma mērķa funkcija z, kas definēta ar izteiksmi (8.1.1) resp. (8.1.6), ir neierobežota pie visām citām parametra t vērtībām, kuras nav vienādas ar pirmajā risināšanas posmā pieņemto parametra t vērtību $t=k$. Kad $\beta_u\leqslant 0$, tad α_u+ $+\beta_u t\leqslant 0$ visiem $t\geqslant\frac{\alpha_u}{\beta_u}$ un mērķa funkcija z ir neierobežota pie visām parametra vērtībām, kuras aizpilda intervālu $\frac{\alpha_u}{\beta_u}\leqslant t\leqslant +\infty$.

Ja turpretim $\beta_u > 0$, tad $\alpha_u + \beta_u t < 0$ visiem $t < -\frac{\alpha_u}{\beta_u}$ un funkcijai z nav maksimuma pie visiem t, kuri aizpilda intervālu $-\infty \leqslant t < -\frac{\alpha_u}{\beta_u}$. Kāda ir funkcijas z optimālā vērtība, ja $t = -\frac{\alpha_u}{\beta_u}$, tas jānoskaidro atsevišķi. Tam nolūkam jāaprēķina iterāciju tabulas nulltās rindas elementi, ievietojot k vietā skaitli $\left(-\frac{\alpha_u}{\beta_u}\right)$ un turpinot risinājumu iepriekš aprakstītajā kārtībā.

Intervāla (8.1.13) galapunktus, t. i., punktus $t = k_0$ un $t = k_1$, sauc par parametra *raksturīgajiem punktiem.*

Ļaujot parametram t atstāt intervālu (8.1.13) uz augšu, t. i., ņemot t lielāku par skaitli k_1, skaitlis $\alpha_{j1} + \beta_{j1} t$ ir mazāks par nulli, kad $\beta_{j1} < 0$. Ja tajā iterāciju tabulas kolonnā, kurā atrodas skaitļi α_{j1} un β_{j1}, ir kaut viens pozitīvs ierobežojumu sistēmas koeficients, jāturpina uzdevuma risināšanas nākošais posms. Tam nolūkam jāpārrēķina iterāciju tabulas nulltās rindas elementi, ievietojot skaitļa $t = k$ vietā skaitli $t = k_1$ un turpinot uzdevuma risināšanu iepriekš aprakstītajā kārtībā līdz tam iterāciju solim, kurā iegūst jaunu parametra vērtībai $t = k_1$ atbilstošo optimālo atrisinājumu vai arī pārliecinās par jaunās mērķa funkcijas z_{k1} neierobežotību.

Jaunajā 2. risināšanas posmā iegūtā pēdējā iterācijas tabula dod iespēju iepriekš aprakstītajā kārtībā noteikt intervālu $k_1 \leqslant t \leqslant k_2$, kura robežās var mainīties mērķa funkcijas (8.1.1) resp. (8.1.6) parametrs t, nemainot fiksētai parametra vērtībai $t = k_1$ atbilstošo optimālo plānu.

Pietiekami daudz reižu atkārtojot šādus risināšanas posmus, iegūst monotoni augošu raksturīgo punktu virkni $k_0, k_1, k_2, \ldots, k_r$.

Ļaujot parametram t atstāt sākumā iegūto intervālu (8.1.13) uz apakšu, t. i., ņemot t mazāku par skaitli k_0, analogā veidā iegūst monotoni dilstošu raksturīgo punktu virkni $k_{-1}, k_{-2}, \ldots, k_{-s}$.

Apvienojot abas tā iegūtās raksturīgo punktu virknes, rodas dotā parametriskās programmēšanas uzdevuma (8.1.1)—(8.1.5) raksturīgo punktu monotoni augoša virkne

$$k_{-s}, k_{-(s+1)}, \ldots, k_{-1}, k_0, k_1, k_2, \ldots, k_r. \tag{8.1.17}$$

Katri divi blakus esošie raksturīgo punktu virknes (8.1.17) locekļi nosaka to intervālu, kurā var mainīt uzdevuma (8.1.1)—(8.1.5) parametra t vērtību, t. i., var izmainīt šī uzdevuma mērķa funkcijas koeficientus un tiem atbilstošo mērķa funkcijas vērtību, nemainot optimālo plānu.

Tādā gadījumā, kad, formulējot kādu parametriskās programmēšanas uzdevumu, ir iepriekš zināms parametra t maksimālais maiņas intervāls $\tau_1 \leqslant t \leqslant \tau_2$, uzdevuma risināšanu sāk ar vienu no šī intervāla galapunktiem. Tad nav jāmeklē tie raksturīgie punkti, kuri atrodas ārpus dotā maksimālā intervāla robežām.

Uzdevums. Atrast maksimumu funkcijai

$$z=(5-t)x_1+(t-8)x_2-x_3 \qquad (8.1.18)$$

pie ierobežojumiem

$$\begin{aligned} -x_1+\ x_2+2x_3&\leqslant 12, \\ x_1+2x_2-\ x_3&\geqslant 8, \\ x_j&\geqslant 0 \ \ (j=1,2,3), \end{aligned} \qquad (8.1.19)$$

kur t ir reāls mainīgs parametrs.

Atrisinājums. Mērķa funkciju (8.1.18) var pārrakstīt šādi:

$$z=z'+z''t, \qquad (8.1.20)$$

kur

$$z'=\ 5x_1-8x_2-x_3, \qquad (8.1.21)$$

$$z''=-x_1+x_2. \qquad (8.1.22)$$

Tā kā pēc uzdevuma nosacījumiem parametra t vērtība var būt jebkurš reāls skaitlis, tad uzdevuma risināšanas pirmajā posmā pieņemsim, ka $t=k=0$. Saskaņā ar šo pieņēmumu pirmajā posmā jāatrod maksimums lineārai funkcijai $z_0=5x_1-8x_2-x_3$ pie ierobežojumiem (8.1.19). Ievedot attiecīgus papildu nezināmos, pārveidojam doto ierobežojumu sistēmu kanoniskajā formā un sastādām sākuma iterācijas tabulu

$+1$	-1	x_1	x_2	x_3	(I)
s	-8	-1	-2	1	M
z_0	0	-5	8	1	0
z'	0	-5	8	1	$0'$
z''	0	$\boxed{1}$	-1	0	$0''$
y_1	12	-1	1	2	1
$-y_2$	8	1	2	-1	2
σ	11	-11	15	3	3

(8.1.23)

Izpildot vienu iterāciju soli ar galveno elementu 1, kas tabulā (8.1.23) apvilkts ar taisnstūri, iegūstam tabulu (8.1.24):

$+1$	-1	y_2	x_2	x_3	(II)
s	0	0	0	0	M
z_0	40	-5	18	-4	0
z'	40	-5	18	-4	$0'$
z''	-8	1	-3	1	$0''$
y_1	20	-1	3	$\boxed{1}$	1
x_1	8	-1	2	-1	2
σ	99	-12	37	-8	3

(8.1.24)

Tabulā (8.1.24) nolasāms atbalsta plāns: $x_1=8$, $x_2=x_3=0$. Tam atbilstošā funkcijas z_0 vērtība 40 nav šīs funkcijas maksimums, jo nulltajā rindā ir negatīvi skaitļi -5 un -4. Tā kā zem negatīvā skaitļa esošajā ierobežojumu sistēmas koeficientu kolonnā nav pozitīva skaitļa, tad funkcija z_0 var augt neierobežoti. Funkcija z ir neierobežota pie visām tām parametra t vērtībām, kuras apmierina nevienādību $-5+t<0$ vai $-\infty<t<5$.

Lai noskaidrotu, vai mērķa funkcijai z pie parametra vērtības $t=5$ ir maksimums, tad tabulas (8.1.24) rindā 0 nolasāmā funkcija $z_0=z'+0\cdot z''$ jāaizstāj ar funkciju $z_5=z'+5z''$ un jāmeklē funkcijas z_5 maksimums. Risināšanu turpinām ar tabulu (8.1.25).

+1	−1	y_2	x_2	x_3	(III)
z_5	0	0	3	1	0
z'	40	−5	18	−4	0′
z''	−8	1	−3	1	0″
y_1	20	−1	3	1	1
x_1	8	−1	2	−1	2

(8.1.25)

Tabulā (8.1.25) nolasāmais plāns $x_1=8$, $x_2=x_3=0$ ir optimāls ne vien vērtībai $t=5$, bet arī visām tām t vērtībām, kuras apmierina nevienādības $-5+t\geqslant 0$, $18-3t\geqslant 0$, $-4+t\geqslant 0$, t. i., $t\geqslant 5$, $t\leqslant 6$, $t\geqslant 4$ jeb $5\leqslant t\leqslant 6$ un $\max z=40-8t$. Atvietojot tabulā (8.1.25) funkciju z_5 ar $z_6=z'+6z''$, iegūstam tabulu (8.1.26), kurai seko tabula (8.1.27):

+1	−1	y_2	x_2	x_3	(IV)
z_6	−8	1	0	2	0
z'	40	−5	18	−4	0′
z''	−8	1	−3	1	0″
y_1	20	−1	3	1	1
x_1	8	−1	$\boxed{2}$	−1	2
σ	51	−6	19	−2	3

(8.1.26)

+1	−1	y_2	x_1	x_3	(V)
z_6	−8	1	0	2	0
z'	−32	4	−9	5	0′
z''	4	−1/2	3/2	−1/2	0″
y_1	8	1/2	−3/2	5/2	1
x_2	4	−1/2	1/2	−1/2	2
σ	−25	7/2	−19/2	15/2	3

(8.1.27)

Tabulā (8.1.27) nolasāmais plāns $x_1=x_3=0$, $x_2=4$ ir optimāls visām tām parametra vērtībām, kuras apmierina nevienādības $4-\frac{1}{2}t\geqslant 0$, $-9+\frac{3}{2}t\geqslant 0$, $5-\frac{1}{2}t\geqslant 0$ jeb $6\leqslant t\leqslant 8$ un $\max z=-32+4t$. Atvietojot tabulā (8.1.27) funkciju z_6 ar $z_8=z'+8z''$, iegūst tabulu (8.1.28). Tai seko tabula (8.1.29):

$+1$	-1	y_2	x_1	x_3	(VI)
z_8	0	0	3	1	0
z'	-32	4	-9	5	$0'$
z''	4	$-1/2$	$3/2$	$-1/2$	$0''$
y_1	8	$\boxed{1/2}$	$-3/2$	$5/2$	1
x_2	4	$-1/2$	$1/2$	$-1/2$	2
σ	-17	$5/2$	$-13/2$	$13/2$	3

(8.1.28)

$+1$	-1	y_1	x_1	x_3	(VII)
z_8	0	0	3	1	0
z'	-96	-8	3	-15	$0'$
z''	12	1	0	2	$0''$
y_2	16	2	-3	5	1
x_2	12	1	-1	2	2
σ	-57	-5	1	-6	3

(8.1.29)

Tā kā tabulas (8.1.29) rindā $0''$ visi nebāzes mainīgo koeficienti ir nenegatīvi, tad visiem t no intervāla $8\leqslant t<+\infty$ šajā tabulā nolasāmais plāns ir optimāls un ar jebkuru t no šī intervāla $\max z=-96+12t$. Raksturīgo punktu virkne uzdevumam (8.1.18), (8.1.19) tātad ir $t_1=5$, $t_2=6$, $t_3=8$. Ja parametra vērtības sakrīt ar raksturīgajiem punktiem, uzdevumam var eksistēt alternatīvi optimālie plāni.

8.2. §. **Matemātiskās programmēšanas uzdevums, kura lineārās ierobežojumu sistēmas brīvie locekļi ir viena parametra lineāras funkcijas.** Tādu uzdevumu vispārīgā veidā vienmēr var formulēt šādi.

Atrast maksimumu lineārai funkcijai

$$z=\sum_{j=1}^{n} c_j x_j \tag{8.2.1}$$

pie ierobežojumiem

$$\sum_{j=1}^{n} a_{ij}x_j \leqslant b'_i + b''_i t \quad (i=1, 2, \ldots, p<h), \tag{8.2.2}$$

$$\sum_{j=1}^{n} a_{ij}x_j \geqslant b'_i + b''_i t \quad (i=p+1, \ldots, h<m), \tag{8.2.3}$$

$$\sum_{j=1}^{n} a_{ij}x_j = b'_i + b''_i t \quad (i=h+1, \ldots, m), \tag{8.2.4}$$

$$x_j \geqslant 0 \quad (j=1, 2, \ldots, n), \tag{8.2.5}$$

kur x_j ir nezināmi reāli lielumi un t — reāls lielums, kas mainās noteiktā intervālā. Pārējie lielumi ir doti reāli skaitļi.

Uzdevuma (8.2.1)—(8.2.5) risināšanu vienmēr var veikt ar iepriekšējā paragrāfā aprakstīto metodi, ja dotā uzdevuma vietā risina tam atbilstošo duālo uzdevumu.

Parametriskās programmēšanas uzdevumu var vēl paplašināt dažādos virzienos, ievedot vairākus parametrus vai aplūkojot lineāras mērķa funkcijas koeficientus (ierobežojumu sistēmas brīvos locekļus) kā nelineāras viena parametra funkcijas. Var aplūkot arī tādus parametriskās programmēšanas uzdevumus, kuru ierobežojumu sistēmas koeficienti ir kāda parametra funkcijas. Šos jautājumus neiztirzāsim.

8.3. §. Daļveida lineārā programmēšana. Par daļveida lineārās programmēšanas uzdevumu sauc tādu matemātiskās programmēšanas uzdevumu, kura mērķa funkcija ir divu lineāru funkciju dalījums, t. i.,

$$z = \frac{c'_1x_1 + c'_2x_2 + \ldots + c'_nx_n}{c''_1x_1 + c''_2x_2 + \ldots + c''_nx_n} = \frac{z_1}{z_2}, \tag{8.3.1}$$

bet šīs funkcijas definīcijas apgabalu vispārīgā gadījumā nosaka lineāru vienādojumu vai nevienādību sistēma:

$$\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i \quad (i=1, 2, \ldots, p<h), \tag{8.3.2}$$

$$\sum_{j=1}^{n} a_{ij}x_j \geqslant b_i \quad (i=p+1, \ldots, h<m), \tag{8.3.3}$$

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \quad (i=h+1, \ldots, m), \tag{8.3.4}$$

$$x_j \geqslant 0 \quad (j=1, 2, \ldots, n), \tag{8.3.5}$$

kur a_{ij}, $b_i \geqslant 0$, c'_j un c''_j ir dotie reālie skaitļi, bet x_j — nezināmi reāli lielumi.

Uzdevumā jāatrod maksimums (vai minimums) nelineārai funkcijai (8.3.1) pie ierobežojumiem (8.3.2)—(8.3.5). Šādā uzdevumā mērķa funkcijas (8.3.1) saucējs z_2 pie ierobežojumiem (8.3.2)—(8.3.5) noteiktā apgabalā nevar būt vienāds ar nulli. Tas nozīmē, ka dotajā apgabalā funkcija z_2 savu zīmi nemaina.

Šajā paragrāfā parādīsim, ka tādu nelineārās programmēšanas uzdevumu var atrisināt ar simpleksa metodi, soli pa solim izpildot vienkāršus papildu aprēķinus katrā simpleksa metodes iterāciju solī pēc tam, kad parastajā kārtībā ir atrasts kāds no dotā uzdevuma atbalsta plāniem.

Ar daļveida lineārās programmēšanas uzdevumiem ekonomikā jāsastopas, ja mērķa funkcija (optimalitātes kritērijs) ir produkcijas ražošanas rentabilitāte.

Lai atrisinātu daļveida lineārās programmēšanas uzdevumu ar simpleksa metodi, tad parastajā kārtībā sastādāmās simpleksa metodes iterāciju tabulās nulltās rindas vietā ieved divas rindas, kuras apzīmēsim ar 0′ un 0″. Pirmās iterāciju tabulas rindā 0′ raksta mērķa funkcijas (8.3.1) lineārā skaitītāja z_1 koeficientus c'_j, bet rindā 0″ — saucēja z_2 koeficientus c''_j. Šo abu rindu elementus katrā iterāciju solī pārveido pilnīgi analogi tam, kā pārveido jebkuras iterāciju tabulas rindu elementus. Pieņemsim, ka pēc v iterāciju soļiem no bāzes mainīgo skaita ir izslēgti visi mākslīgie un ar mīnusa zīmi ierakstītie papildu nezināmie. Tas nozīmē, ka dotā uzdevuma ierobežojumu sistēma ir saderīga un attiecīgais bāzes atrisinājums nolasāms pēc v soļiem iegūtajā iterāciju tabulā, kura vispārīgā gadījumā var būt šāda:

$+1$	-1	y_1	$\cdots$	x_k	$\cdots$	x_n
z_1	Q'_v	q'_1	$\cdots$	q'_k	$\cdots$	q'_n
z_2	Q''_v	q''_1	$\cdots$	q''_k	$\cdots$	q''_n
x_1	b_{10}	b_{11}	$\cdots$	b_{1k}	$\cdots$	b_{1n}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
y_r	b_{r0}	b_{r1}	$\cdots$	b_{rk}	$\cdots$	b_{rn}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
y_m	b_{m0}	b_{m1}	$\cdots$	b_{mk}	$\cdots$	b_{mn}
σ	σ_0	σ_1	$\cdots$	σ_k	$\cdots$	σ_n

(8.3.6)

Tabulā (8.3.6) nolasāmajam atbalsta plānam atbilst mērķa funkcijas z vērtība, kuru apzīmēsim ar

$$Q_v = \frac{Q'_v}{Q''_v}. \tag{8.3.7}$$

Ja ar kādu tabulas (8.3.6) elementu $b_{rk} > 0$ kā ar galveno elementu izpildītu vēl vienu iterācijas soli, lai iegūtu citu atbalsta plānu, tad jaunajā tabulā nolasāmajam atbalsta plānam atbilstu funkcijas z vērtība, kuru apzīmēsim ar

$$Q_{v+1}=\frac{Q'_{v+1}}{Q''_{v+1}}, \tag{8.3.8}$$

kur saskaņā ar simpleksa metodes algoritmu

$$\left.\begin{aligned} Q'_{v+1}&=Q'_v-\frac{q'_k b_{r0}}{b_{rk}},\\ Q''_{v+1}&=Q''_v-\frac{q''_k b_{r0}}{b_{rk}}. \end{aligned}\right\} \tag{8.3.9}$$

Viena tāda iterācijas soļa rezultātā mērķa funkcija iegūst pieaugumu

$$\Delta z=Q_{v+1}-Q_v$$

jeb, ņemot vērā formulas (8.3.7)—(8.3.9),

$$\Delta z=\frac{(Q'_v q''_k-Q''_v q'_k)}{Q''_v Q''_{v+1}}\cdot\frac{b_{r0}}{b_{rk}}. \tag{8.3.10}$$

Saskaņā ar simpleksa metodes galvenās rindas noteikšanas algoritmu attiecība $\theta=b_{r0}/b_{rk}$ ir nenegatīvs skaitlis. Pēc uzdevuma nosacījuma mērķa funkcijas saucēja zīme ierobežojumu apgabalā nevar mainīties. Tādēļ $Q''_v\cdot Q''_{v+1}>0$ un Δz zīme ir vienāda ar determinanta

$$d_k=Q'_v q''_k-Q''_v q'_k=\begin{vmatrix} Q'_v & Q''_v \\ q'_k & q''_k \end{vmatrix}$$

zīmi. Tas nozīmē, ka funkcija z iegūst pozitīvu pieaugumu, ja nākošā iterācijas soļa izpildīšanai galveno elementu var izvēlēties tādā kolonnā, kurai atbilstošais determinants

$$-d_k=\begin{vmatrix} Q''_v & Q'_v \\ q''_k & q'_k \end{vmatrix}=Q''_v q'_k-Q'_v q''_k \tag{8.3.11}$$

ir negatīvs. Ja tajā iterācijas tabulā, kurā nolasāms kāds atrasts atbalsta plāns, nav tādas kolonnas, kurai atbilstošais determinants (8.3.11) ir negatīvs, tad šajā tabulā atrodams mērķa funkcijas z maksimums. To aprēķina ar formulu (8.3.7) resp. (8.3.8).

Ja tajā iterācijas tabulā, kurā nolasāms atrastais atbalsta plāns, nav tādas kolonnas, kurai atbilstošais determinants (8.3.11) ir pozitīvs, tad šajā tabulā atrodams mērķa funkcijas z minimums. To aprēķina ar formulu (8.3.7) resp. (8.3.8).

U z d e v u m s 8.3.1°. Atrast maksimumu funkcijai

$$z=\frac{2x_1+x_2+3x_3}{x_1+3x_2+4x_3} \tag{8.3.12}$$

pie ierobežojumiem

$$\left.\begin{aligned} -2x_1+5x_2+\ x_3 &\leqslant 3,\\ -2x_1+7x_2+3x_3 &\leqslant 12,\\ 2x_1-3x_2+3x_3 &\geqslant 1, \end{aligned}\right\} \qquad (8.3.13)$$

$$x_j \geqslant 0 \ (j=1, 2, 3).$$

Atrisinājums. Ievedot attiecīgus papildu nezināmos, pārveidojam doto ierobežojumu sistēmu kanoniskajā formā un sastādām sākuma iterācijas tabulu:

(8.3.14)

+1	−1	x_1	x_2	x_3	(I)
s	−1	−2	3	−3	M
z_1	0	−2	−1	−3	0′
z_2	0	−1	−3	−4	0″
y_1	3	−2	5	1	1
y_2	12	−2	7	3	2
$-y_3$	1	2	−3	$\boxed{3}$	3
σ	14	−8	7	−4	4

Izdarot vienu iterāciju soli ar galveno elementu 3, kas tabulā (8.3.14) apvilkts ar taisnstūri, iegūstam jaunu iterācijas tabulu (8.3.15). Tās beigās pievienosim rindu, kurā ierakstīti turpmāk izpildīto blakus aprēķinu rezultāti:

(8.3.15)

+1	−1	x_1	x_2	y_3	(II)
s	0	0	0	0	M
z_1	1	0	−4	−1	0′
z_2	4/3	5/3	−7	−4/3	0″
y_1	8/3	−8/3	6	1/3	1
y_2	11	−4	10	1	2
x_3	1/3	$\boxed{2/3}$	−1	−1/3	3
σ	46/3	−16/3	3	−7/3	4
	$z=3/4$	−5/3	5/3	0	5

Pielīdzinot tabulas (8.3.15) nebāzes mainīgos nullei, nolasām atbalsta plānu $x_1=x_2=0$, $x_3=1/3$. Šim plānam atbilstošā mērķa funkcijas vērtība $z=3/4$ iegūstama, izdalot tabulā esošo $z_1=1$ ar $z_2=4/3$. Pārējie tabulas (8.3.15) papildu rindā 5 ierakstītie skaitļi ir determinanti, kas katrai kolonnai aprēķināti pēc formulas (8.3.11), jo

$$-d_1=\frac{4}{3}\cdot 0 \quad -1\cdot\frac{5}{3}=-\frac{5}{3},$$

$$-d_2=\frac{4}{3}\cdot(-4)-1\cdot(-7)=\frac{5}{3},$$

$$-d_3=\frac{4}{3}\cdot(-1)-1\cdot\left(-\frac{4}{3}\right)=0.$$

Tā kā tabulas (8.3.15) 5. rindā ir negatīvs skaitlis $(-5/3)$, tad funkcijas (8.3.12) vērtību var palielināt, ja nākošā iterācijas soļa izpildīšanai par galveno izvēlas kolonnu, kurā atrodas šis negatīvais skaitlis. Izdarot nākošo iterāciju soli ar galveno elementu 2/3, kas tabulā (8.3.15) apvilkts ar taisnstūri, un pēc iterācijas soļa izpildīšanas jau aprakstītajā kārtībā no jauna aizpildot papildu rindu 5, iegūst tabulu (8.3.16):

$+1$	-1	x_3	x_2	y_3	(III)
z_1	1	0	-4	-1	0′
z_2	1/2	$-5/2$	$-9/2$	$-1/2$	0″
y_1	4	4	2	-1	1
y_2	13	6	4	-1	2
x_1	1/2	3/2	$-3/2$	$-1/2$	3
σ	18	8	-5	-5	4
	$z=2$	5/2	5/2	0	5

(8.3.16)

Tā kā tabulas (8.3.16) papildu rindā 5 nav neviena negatīva determinanta, tad $z=2$ ir meklējamais dotā uzdevuma mērķa funkcijas maksimums. To šī funkcija sasniedz, ja argumentu vērtības ir $x_1=1/2$, $x_2=x_3=0$.

Uzdevums 8.3.2°. Atrast iepriekšējā uzdevuma mērķa funkcijas (8.3.12) minimumu pie ierobežojumiem (8.3.13).

Atrisinājums. Uzdevuma risinājuma sākums saskan ar tabulām (8.3.14), (8.3.15), bet tabulā (8.3.15) par galveno jāizvēlas tāda kolonna, kurā atrodas papildu rindas 5 pozitīvs skaitlis. Tāds skaitlis ir 5/3, un nākošais iterācijas solis jāizpilda ar galveno elementu 6. Pēc šāda iterācijas soļa iegūst tabulu (8.3.17):

$+1$	-1	x_1	y_1	y_3	(III′)
z_1	25/9	−16/9	2/3	−7/9	0′
z_2	40/9	−13/9	7/6	−17/18	0″
x_2	4/9	−4/9	1/6	1/18	1
y_2	59/9	4/9	−5/3	4/9	2
x_3	7/9	2/9	1/6	−5/18	3
σ	14	−4	−1/2	−5/2	4
	$z=5/8$	−35/9	−5/18	−5/6	5

(8.3.17)

Tā kā tabulas (8.3.17) papildu rindā 5 nav neviena pozitīva determinanta, tad $z=5/8$ ir meklējamais mērķa funkcijas minimums. To šī funkcija sasniedz, ja argumentu vērtības ir $x_1=0$, $x_2=4/9$ un $x_3=7/9$.

II DAĻA

LINEĀRĀS PROGRAMMĒŠANAS TEORIJAS PAPILDINĀJUMI

9. nodaļa

LINEĀRĀS PROGRAMMĒŠANAS UZDEVUMU PLĀNU KOPA

9.1. §. n-dimensiju vektoru telpas un izliektas punktu kopas jēdzieni. Analītiskajā ģeometrijā par vektoru sauc orientētu taisnes nogriezni. Vektora projekcijas uz taisnleņķa koordinātu sistēmas asīm sauc par vektora koordinātēm. Trīsdimensiju telpā vektoru $\boldsymbol{a}$ viennozīmīgi definē kā triju sakārtotu skaitļu kopu (a_1, a_2, a_3). Pēc analoģijas ar trīsdimensiju telpas vektoru n sakārtotu skaitļu kopu ($a_1, a_2, \ldots, a_n$) sauc par *n-dimensiju vektoru.* Skaitļus $a_1, a_2, \ldots, a_n$ sauc par n-dimensiju vektora koordinātēm. Ņemot vērā, ka jebkura rādiusvektora koordinātes ir vienādas ar vektora galapunkta koordinātēm, termini «punkts» un «vektors» turpmāk lietoti kā sinonīmi. Vektora koordinātes var pierakstīt *kolonnā* (ailē), rakstot tās numuru kārtībā citu zem citas, vai arī *rindā,* rakstot tās citu aiz citas.

Ja vektora koordinātes pierakstāmas kolonnā, tad tā apzīmēšanai lietosim latīņu alfabēta mazos burtus pustreknā iespiedumā bez indeksa vai arī ar indeksu apakšā pie burta, ar kuru apzīmēts vektors, piemēram,

$$\boldsymbol{a}=\begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_n \end{bmatrix} \qquad \boldsymbol{a}_k=\begin{bmatrix} a_{1k} \\ a_{2k} \\ \cdot \\ \cdot \\ \cdot \\ a_{nk} \end{bmatrix}$$

Ja vektora koordinātes pierakstāmas rindā, tad tā apzīmēšanai lietosim latīņu alfabēta mazos burtus pustreknā iespiedumā ar burtu T vai kādu citu indeksu augšā pie burta, ar kuru apzīmēts vektors, piemēram, $\boldsymbol{a}^{\mathrm{T}}=(a_1, a_2, \ldots, a_n)$ vai $\boldsymbol{a}^k=(a_{k1}, a_{k2}, \ldots, a_{kn})$.

Ja $\boldsymbol{a}^{\mathrm{T}}=(a_1, a_2, \ldots, a_n)$, tad $(\boldsymbol{a}^{\mathrm{T}})^{\mathrm{T}}=(a_1, a_2, \ldots, a_n)^{\mathrm{T}}=\boldsymbol{a}$.

Saskaņā ar šiem apzīmējumiem jebkuras taisnstūrveida skaitļu tabulas (matricas) rinda vai kolonna ir vektors.

Divi vektori $\boldsymbol{a}=(a_1, a_2, \ldots, a_n)^{\mathrm{T}}$ un $\boldsymbol{b}=(b_1, b_2, \ldots, b_n)^{\mathrm{T}}$ ir vienādi, t. i., $\boldsymbol{a}=\boldsymbol{b}$, tad un tikai tad, ja visiem $i=1, 2, \ldots, n$ izpildās vienādība $a_i=b_i$; $\boldsymbol{a}\leqslant\boldsymbol{b}$, ja $a_i\leqslant b_i$ un $\boldsymbol{a}\geqslant\boldsymbol{b}$, ja $a_i\geqslant b_i$.

Par *vektora* $\boldsymbol{a}=(a_1, a_2, \ldots, a_n)^{\mathrm{T}}$ *reizinājumu ar skaitli* c sauc vektoru

$$c\boldsymbol{a}=(ca_1, ca_2, \ldots, ca_n)^{\mathrm{T}}. \tag{9.1.1}$$

Par vektoru $\boldsymbol{a}=(a_1, a_2, \ldots, a_n)^{\mathrm{T}}$ un $\boldsymbol{b}=(b_1, b_2, \ldots, b_n)^{\mathrm{T}}$ *summu* sauc vektoru

$$\boldsymbol{a}+\boldsymbol{b}=(a_1+b_1, a_2+b_2, \ldots, a_n+b_n)^{\mathrm{T}}. \tag{9.1.2}$$

Par divu vektoru $\boldsymbol{a}=(a_1, a_2, \ldots, a_n)^{\mathrm{T}}$ un $\boldsymbol{b}=(b_1, b_2, \ldots, b_n)^{\mathrm{T}}$ *skalāro reizinājumu* sauc skaitli

$$\boldsymbol{a}^{\mathrm{T}}\boldsymbol{b}=\boldsymbol{b}^{\mathrm{T}}\boldsymbol{a}=\sum_{i=1}^{n} a_i b_i. \tag{9.1.3}$$

Skaitli $|\boldsymbol{a}|=\sqrt{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{a}}=\sqrt{a^2{}_1+a^2{}_2+\ldots+a^2{}_n}$ sauc par vektora $\boldsymbol{a}=(a_1, a_2, \ldots, a_n)^{\mathrm{T}}$ *moduli* vai garumu. Vektoru $\boldsymbol{a}^{\circ}$, kura modulis $|\boldsymbol{a}^{\circ}|=1$, sauc par *vienības vektoru.*

Izteiksmi

$$x_1\boldsymbol{a}_1+x_2\boldsymbol{a}_2+\ldots+x_n\boldsymbol{a}_n=\boldsymbol{b} \tag{9.1.4}$$

sauc par vektoru sistēmas $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n$ *lineāru kombināciju,* bet skaitļus $x_1, x_2, \ldots, x_n$ — par šīs lineārās kombinācijas koeficientiem. Ievērosim, ka izteiksme (9.1.4) ir viens no lineāru algebrisku vienādojumu sistēmas pierakstiem, ja vektori $\boldsymbol{a}_i$ $(i=1, 2, \ldots, n)$ ir vienādojumu sistēmas koeficientu matricas kolonnas, $\boldsymbol{b}$ — brīvo locekļu vektors un $\boldsymbol{x}=(x_1, x_2, \ldots, x_n)$ — sistēmas atrisinājums.

Vektoru, kuram visas koordinātes ir nulles, sauc par *nullvektoru.* To apzīmēsim ar $\boldsymbol{o}$ vai $\boldsymbol{o}^{\mathrm{T}}$.

Vektoru sistēmu $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k$ sauc par *lineāri atkarīgu,* ja vektoriālajam vienādojumam

$$x_1\boldsymbol{a}_1+x_2\boldsymbol{a}_2+\ldots+x_k\boldsymbol{a}_k=\boldsymbol{o} \tag{9.1.5}$$

eksistē netriviāls atrisinājums $(x_1, x_2, \ldots, x_k)^{\mathrm{T}}\neq\boldsymbol{o}$. Vektoru sistēmu $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k$ sauc par *lineāri neatkarīgu,* ja vektoriālajam vienādojumam (9.1.5) nav netriviālu atrisinājumu. Lineārā atkarība vai neatkarība tātad ir vektoru sistēmas, bet ne atsevišķa šīs sistēmas vektora īpašība.

Kopas jēdziens pieder pie matemātikas pamatjēdzieniem, kas nav definējami ar citiem, vienkāršākiem jēdzieniem. Vārdu «kopa» bieži sastopam arī ikdienas valodā. Vispār, ja objekts A kaut kā definē objektus $a, b, c, \ldots$ un, otrādi, ja objekti $a, b, c, \ldots$ savukārt definē objektu A, tad šo savstarpējo sakarību var izteikt ar vārdiem: *kopa* A *sastāv no elementiem* $a, b, c, \ldots$. Atkarībā no kopai piederošo elementu daudzuma izšķir galīgas un bezgalīgas kopas. Kopa ir galīga, ja tās elementus var sanumurēt un numuru skaits nav neierobežots. Elementa a piederību pie kopas A apzīmē ar simbolu $a \in A$. Kopu sauc par tukšu, ja tajā nav neviena elementa. Tukšu kopu apzīmē ar simbolu $\varnothing$. Ja katrs kopas A elements pieder arī kopai B, tad A ir kopas B *apakškopa*. To pieraksta ar simboliem $A \subseteq B$ vai $B \supseteq A$. Ja vienlaicīgi $A \subseteq B$ un $B \subseteq A$, tad raksta $A = B$ vai $B = A$ un saka, ka kopas A un B ir vienādas. Ja tās nav vienādas, tad var rakstīt $A \neq B$.

Par divu kopu A un B *apvienojumu* (jeb summu) sauc kopu $S = A \cup B$, kuras katrs elements pieder kopai A vai kopai B (vai abām šīm kopām).

Par divu kopu *šķēlumu* (jeb reizinājumu) sauc kopu $M = A \cap B$, kuras katrs elements pieder kā kopai A, tā arī kopai B.

Ja $M = A \cap B = \varnothing$, tad kopām A un B nav neviena kopīga elementa. Tādas kopas sauc par *disjunktīvām* kopām.

Par kopu A un B *starpību* sauc kopu $D = A \setminus B$, kuras katrs elements pieder kopai A, bet neviens tās elements nepieder kopai B.

Ja kopa A ir kopas G apakškopa, tad par kopas A *papildkopu* (attiecībā uz kopu G) sauc kopu $\bar{A} = G \setminus A$.

Ja aplūkojam bezgalīgas punktu kopas plaknē, tad visiem tikko definētajiem jēdzieniem ir uzskatāma ģeometriska interpretācija, kas attēlota zīmējumā 9.1.

n-dimensiju vektoru kopu E_n, kuras elementiem ir īpašības, kas definētas ar formulām (9.1.1)—(9.1.3), sauc par *n-dimensiju Eiklīda telpu*, ja šajā kopā eksistē lineāri neatkarīga n vektoru sistēma un katra šai kopai piederošā $n+1$ vektoru sistēma ir lineāri atkarīga.

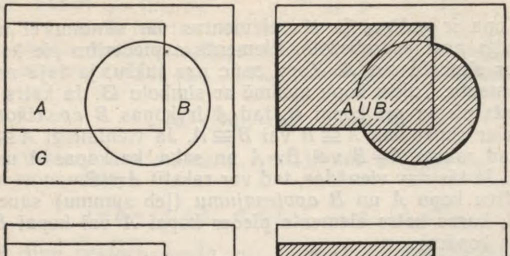
Par telpas E_n *bāzi* sauc katru lineāri neatkarīgu n vektoru sistēmu. Jebkuru telpai E_n piederošu vektoru var viennozīmīgi izteikt kā bāzes vektoru lineāru kombināciju. Viena no visparocīgākajām telpas E_n bāzēm ir vienības vektoru sistēma (ortonormēta bāze):

$$\left.\begin{aligned} \boldsymbol{e}_1 &= (1, 0, \ldots, 0)^T, \\ \boldsymbol{e}_2 &= (0, 1, \ldots, 0)^T, \\ &\ldots\ldots\ldots\ldots \\ \boldsymbol{e}_n &= (0, 0, \ldots, 1)^T. \end{aligned}\right\} \qquad (9.1.6)$$

Lineārās programmēšanas teorijā svarīga nozīme ir izliektas (konveksas) kopas jēdzienam. Šis jēdziens savukārt cieši saistīts ar n-dimensiju Eiklīda telpas punkta apkārtnes un nogriežņa jēdzieniem.

Definīcija 1°. *Par punkta $\boldsymbol{a}$ ε-apkārtni sauc visu to punktu $\boldsymbol{x}$ kopu, kuru attālums $|\boldsymbol{a}-\boldsymbol{x}|$ no punkta $\boldsymbol{a}$ ir mazāks par dotu pozitīvu skaitli ε.*

Definīcija 2°. *Punktu $\boldsymbol{x}$ sauc par punktu kopas $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$*



Zīm. 9.1.

$\ldots, x_p$ *izliektu lineāru kombināciju, ja* $x = \lambda_1 x_1 + \lambda_2 x_2 + \ldots + \lambda_p x_p$, kur $\lambda_i \geqslant 0$ $(i = 1, 2, \ldots, p)$ un $\lambda_1 + \lambda_2 + \ldots + \lambda_p = 1$.

Tā, piemēram, ja ir doti divi punkti $a = (2; 4)^T$ un $b = (8; 2)^T$, tad izteiksme $\lambda a + (1 - \lambda) b$, kur $0 \leqslant \lambda \leqslant 1$, ir jebkura punktu a un b izliekta lineāra kombinācija (zīm. 9.2).

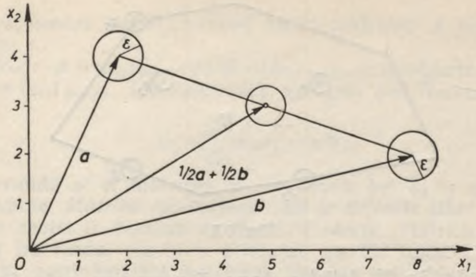
Definīcija 3°. Par nogriezni, kas savieno punktus x_1 un x_2 n-dimensiju telpā, sauc visu to punktu kopu, kuru rādiusvektorus x nosaka formula

$$x = \lambda x_1 + (1 - \lambda) x_2, \quad 0 \leqslant \lambda \leqslant 1 \tag{9.1.7}$$

jeb

$$x = \lambda_1 x_1 + \lambda_2 x_2, \quad \lambda_1 + \lambda_2 = 1. \tag{9.1.8}$$

Ņemot vērā definīciju 2°, jebkurš nogriežņa (x_1, x_2) punkts x ir tā galapunktu izliekta lineāra kombinācija. Šeit un arī turpmāk tiek pieņemts, ka $x_1 \neq x_2$, resp., ka nogrieznis nav sarucis par vienu punktu.



Zīm. 9.2.

D e f i n ī c i j a 4°. *n-dimensiju telpas E_n apakškopu K sauc par izliektu kopu, ja no punktu* $\boldsymbol{a} \in K$ un $\boldsymbol{b} \in K$ *piederības pie K izriet, ka nogrieznis* $(\boldsymbol{a}, \boldsymbol{b})$ *ir kopas K apakškopa, resp.,* $(\boldsymbol{a}, \boldsymbol{b}) \subseteq K$.

Saskaņā ar šo definīciju, piemēram, riņķis ir izliekta punktu kopa, bet riņķa līnija nav izliekta punktu kopa. Vienu atsevišķu n-dimensiju telpas punktu uzskata par izliektu kopu, kurā ietilpst tikai viens elements.

Vispārīgā gadījumā visus izliektas kopas punktus var sadalīt divās grupās: iekšējos un virsmas punktos.

D e f i n ī c i j a 5°. *Punktu* $\boldsymbol{a} \in K$ *sauc par izliektas kopas K iekšējo punktu, ja eksistē tāds* $\varepsilon > 0$, *ka* $\boldsymbol{a}$ *visa* ε*-apkārtne pieder pie kopas K.*

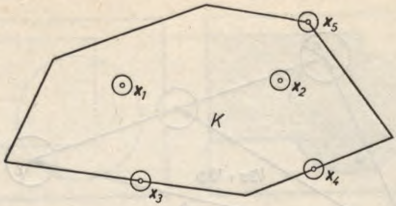
D e f i n ī c i j a 6°. *Punktu* $\boldsymbol{a} \in K$ *sauc par izliektas kopas K virsmas punktu, ja šī punkta* $\boldsymbol{a}$ ε*-apkārtnē pie jebkura* $\varepsilon > 0$ *ietilpst gan kopas K iekšējie punkti, gan arī tādi punkti, kuri nepieder pie kopas K.*

Izliekta kopa ir *noslēgta*, ja pie tās pieder visi šīs kopas virsmas punkti.

Starp izliektas kopas virsmas punktiem īpaša nozīme ir t. s. ekstremālajiem punktiem jeb *virsotnēm*. Izliektas kopas K virsmas punktu $\boldsymbol{p}$ sauc par virsotni, ja nav tāda nogriežņa $(\boldsymbol{a}, \boldsymbol{b}) \subseteq K$, kura viduspunkts sakristu ar punktu $\boldsymbol{p}$.

Tā, piemēram, zīmējumā 9.3 attēlotais daudzstūris veido izliektu punktu kopu K. Punkti $\boldsymbol{x}_1 \in K$ un $\boldsymbol{x}_2 \in K$ ir šīs kopas iekšējie punkti, bet $\boldsymbol{x}_3 \in K$ un $\boldsymbol{x}_4 \in K$ ir virsmas punkti. Virsmas punkts $\boldsymbol{x}_5 \in K$ ir virsotne.

Izliektas *kopas dimensiju skaitu* nosaka lineāri neatkarīgo vektoru skaits sistēmā $(\boldsymbol{x} - \boldsymbol{a}_0) \in K$, kur $\boldsymbol{a}_0 \in K$ ir brīvi izraudzīts kopas K elements, bet $\boldsymbol{x} \in K$ — katrs šīs kopas elements. Tas nozīmē, piemēram, ka punkts ir nulldimensionāla, nogrieznis — viendimensionāla, riņķis — divdimensionāla, lode — trīsdimensionāla izliekta punktu kopa.



Zīm. 9.3.

Definīcija 7°. *Izliektu kopu K sauc par neierobežotu, ja katram punktam $\boldsymbol{x} \in K$ atbilst tāds vektors $\boldsymbol{h} \neq \boldsymbol{o}$, ka punkts $(\boldsymbol{x}+\lambda \boldsymbol{h}) \in K$ pie jebkura pēc patikas liela $\lambda \geqslant 0$. Pretējā gadījumā K ir ierobežota kopa.*

Definīcija 8°. *Izliektu kopu sauc par izliektu daudzskaldni (vai poliedru), ja tā ir ierobežota un tai ir galīgs virsotņu skaits.* Tā, piemēram, tetraedrs ir izliekts daudzskaldnis, bet lode tāda nav, jo tās virsotņu skaits ir bezgalīgi liels.

9.2. §. Matricu algebras elementi. Divas matricas $\boldsymbol{A}_{(m\times n)} = (a_{ij})_{(m\times n)}$ un $\boldsymbol{B}_{(m\times n)} = (b_{ij})_{(m\times n)}$ ir vienādas, $\boldsymbol{A}=\boldsymbol{B}$, ja tām ir vienāds rindu skaits m, vienāds kolonnu skaits n un ja visi matricas $\boldsymbol{A}$ elementi a_{ij} $(i=1, 2, \ldots, m;\ j=1, 2, \ldots, n)$ ir vienādi ar attiecīgajiem matricas $\boldsymbol{B}$ elementiem $b_{ij}=a_{ij}$. Ja visiem i un j izpildās nevienādības $a_{ij} \leqslant b_{ij}$, tad $\boldsymbol{A} \leqslant \boldsymbol{B}$, bet, ja visiem i un j izpildās nevienādības $a_{ij} \geqslant b_{ij}$, tad $\boldsymbol{A} \geqslant \boldsymbol{B}$. Matricu, kurai visi elementi vienādi ar 0, sauc par *nullmatricu*. To apzīmēsim ar $\boldsymbol{O}$.

Apmainot lomām matricas $\boldsymbol{A}_{(m\times n)}$ rindas ar kolonnām, iegūsim jaunu matricu $\boldsymbol{A}^{\mathrm{T}}_{(n\times m)}$, kuru sauc par matricas $\boldsymbol{A}$ *transponēto matricu*. Saskaņā ar transpozīcijas definīciju $(\boldsymbol{A}^{\mathrm{T}})^{\mathrm{T}}=\boldsymbol{A}$.

Pierakstot vienā taisnstūrveida tabulā blakus matricas $\boldsymbol{A}_{(m\times n)}$ un $\boldsymbol{B}_{(m\times s)}$, kam ir vienāds rindu skaits, iegūst *blokmatricu* $\boldsymbol{C}_{[m\times(n+s)]} = [\boldsymbol{A} \mid \boldsymbol{B}]$, kurai ir m rindas un $n+s$ kolonnas. Matricas $\boldsymbol{A}$ un $\boldsymbol{B}$ šādā gadījumā sauc par matricas $\boldsymbol{C}$ *apakšmatricām*. Analogi matricas $\boldsymbol{D}_{(r\times p)}$ un $\boldsymbol{F}_{(s\times p)}$, kam ir vienāds kolonnu skaits, var būt apakšmatricas blokmatricai

$$\boldsymbol{G}_{[(r+s)\times p]} = \left[\begin{array}{c}\boldsymbol{D}\\ \hline \boldsymbol{F}\end{array}\right] = [\boldsymbol{D}^{\mathrm{T}} \mid \boldsymbol{F}^{\mathrm{T}}]^{\mathrm{T}}.$$

Matricu $\boldsymbol{A}=(a_{ij})_{(m\times n)}$ un $\boldsymbol{B}=(b_{ij})_{(m\times n)}$ *summa* ir matrica $\boldsymbol{C}=\boldsymbol{A}+\boldsymbol{B}$, kuras elementi $c_{ij}=a_{ij}+b_{ij}$. *Summa ir definēta tikai vienāda tipa matricām,* t. i., tādām matricām, kurām nav dažāda rindu vai kolonnu skaita.

Matricas $\boldsymbol{A}=(a_{ij})_{(m\times n)}$ *reizinājums ar skaitli c* ir matrica $c\boldsymbol{A}$, kuras elementus iegūst, reizinot katru matricas $\boldsymbol{A}$ elementu a_{ij} ar skaitli c.

Matricu $\boldsymbol{A}=(a_{ij})_{(m\times n)}$ un $\boldsymbol{B}=(b_{ij})_{(n\times p)}$ *reizinājums ir matrica* $\boldsymbol{C}=\boldsymbol{AB}=(c_{ij})_{(m\times p)}$. Tās elementus aprēķina pēc formulas

$$c_{ij}=\sum_{k=1}^{n} a_{ik}b_{kj}=\boldsymbol{a}^i\boldsymbol{b}_j.$$

Šajā formulā $\boldsymbol{a}^i$ ir matricas $\boldsymbol{A}$ i-tā rinda, bet $\boldsymbol{b}_j$ ir matricas $\boldsymbol{B}$ j-tā kolonna. *Matricu reizinājums* $\boldsymbol{AB}$ *ir definēts tikai tad, ja* $\boldsymbol{A}$ *kolonnu skaits ir vienāds ar* $\boldsymbol{B}$ *rindu skaitu.* Matricu reizinājumam ir tik rindu, cik to ir pirmajai, un tik kolonnu, cik to ir otrajai no reizināmajām matricām. Matricu reizinājumam ir asociatīvā un distributīvā īpašība, bet vispārīgā gadījumā nav komutatīvās īpašības, t. i., vispār $\boldsymbol{AB}\neq\boldsymbol{BA}$. Tādēļ, pierakstot matricu reizinājumu, jāievēro reizinātāju kārtība, jo, piemēram, ja matricas $\boldsymbol{B}$ kolonnu skaits nav vienāds ar matricas $\boldsymbol{A}$ rindu skaitu, tad reizinājums $\boldsymbol{BA}$ neeksistē, kaut arī eksistētu $\boldsymbol{AB}$. Matricu reizinājuma transpozīcijai pastāv sakarība $(\boldsymbol{AB})^{\mathrm{T}}=\boldsymbol{B}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}$.

Matricu $\boldsymbol{A}_{(n\times n)}$ sauc par *n-tās kārtas kvadrātisku matricu.* Šīs matricas rindu un kolonnu skaits ir vienāds. Kvadrātiskas matricas elementus, kuru rindas un kolonnas numuri ir vienādi, sauc par galvenās diagonāles elementiem.

Kvadrātisku matricu, kurai visi elementi ir nulles, izņemot galvenās diagonāles elementus, sauc par *diagonālmatricu.*

Diagonālmatricu, kuras visi galvenās diagonāles elementi ir vieninieki, sauc par *vienības matricu.* To apzīmēsim ar $\boldsymbol{E}$. Acīm redzami $\boldsymbol{E}=[\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n]$, kur $\boldsymbol{e}_i$ $(i=1, 2, \ldots, n)$ ir vienības vektori (9.1.6).

Vienības matrica ir komutatīva ar jebkuru tās pašas kārtas kvadrātisku matricu, t. i., $\boldsymbol{EA}=\boldsymbol{AE}$.

Ja matrica $\boldsymbol{B}$ ir komutatīva ar matricu $\boldsymbol{A}$ un reizinājums $\boldsymbol{AB}=\boldsymbol{BA}$ ir vienības matrica, tad $\boldsymbol{B}$ sauc par matricas $\boldsymbol{A}$ *inverso matricu.* To apzīmē ar $\boldsymbol{A}^{-1}$. Saskaņā ar definīciju $\boldsymbol{A}^{-1}\boldsymbol{A}=\boldsymbol{AA}^{-1}=$ $=\boldsymbol{E}$. Ja $\boldsymbol{A}^{-1}$ eksistē, tad $\boldsymbol{A}$ ir *nesingulāra* matrica.

Ja ir definēts matricu $\boldsymbol{A}$ un $\boldsymbol{B}$ reizinājums $\boldsymbol{AB}$ un šīm matricām eksistē inversās matricas $\boldsymbol{A}^{-1}$ un $\boldsymbol{B}^{-1}$, tad $(\boldsymbol{AB})^{-1}=\boldsymbol{B}^{-1}\boldsymbol{A}^{-1}$.

Matricu $\boldsymbol{A}$ sauc par *simetrisku,* ja $\boldsymbol{A}=\boldsymbol{A}^{\mathrm{T}}$, bet par *antisimetrisku,* ja $\boldsymbol{A}=-\boldsymbol{A}^{\mathrm{T}}$.

Ja $\boldsymbol{A}$ un $\boldsymbol{B}$ ir blokmatricas, tad, ievērojot vajadzīgos nosacījumus, to summas un reizinājuma veidošanai var izmantot attiecīgās apakšmatricas (blokus). Divu vienāda skaita, tipa un izvietojuma blokos sadalītu matricu $\boldsymbol{A}$ un $\boldsymbol{B}$ summu iegūst, saskaitot šo matricu attiecīgos blokus kā patstāvīgas matricas. Tā, piemēram, matricu

$$\boldsymbol{A}=\left[\begin{array}{c|c}\boldsymbol{A}_{11} & \boldsymbol{A}_{12}\\ \hline \boldsymbol{A}_{21} & \boldsymbol{A}_{22}\end{array}\right] \text{ un } \boldsymbol{B}=\left[\begin{array}{c|c}\boldsymbol{B}_{11} & \boldsymbol{B}_{12}\\ \hline \boldsymbol{B}_{21} & \boldsymbol{B}_{22}\end{array}\right]$$

summa ir

$$A+B=\left[\begin{array}{c|c} A_{11}+B_{11} & A_{12}+B_{12} \\ \hline A_{21}+B_{21} & A_{22}+B_{22} \end{array}\right],$$

ja bloku A_{ij} un B_{ij} tipi visiem i un j ir vienādi.

Blokmatricu A un B reizinājumu veidojot, jāievēro cits nosacījums. Reizinājums AB ir blokmatrica

$$\left[\begin{array}{c|c} A_{11}B_{11}+A_{12}B_{21} & A_{11}B_{12}+A_{12}B_{22} \\ \hline A_{21}B_{11}+A_{22}B_{21} & A_{21}B_{12}+A_{22}B_{22} \end{array}\right],$$

ja vien attiecīgo bloku reizinājumi eksistē, t. i., ja bloka A_{ik} kolonnu skaits ir vienāds ar bloka B_{kj} rindu skaitu visiem i, j un k.

9.3. §. Lineārās programmēšanas plānu kopas īpašības.

Lietojot matricu un vektoru simboliku, jebkuru lineārās programmēšanas uzdevumu var formulēt šādi.

Atrast maksimumu lineārai funkcijai

$$z(x)=c^{T}x \tag{9.3.1}$$

pie ierobežojumiem

$$Ax=b \quad (b\geqslant o), \tag{9.3.2}$$

$$x\geqslant o, \tag{9.3.3}$$

kur A ir dotā matrica ar m rindām un $n>m$ kolonnām, c — dots n-dimensiju telpas E_n vektors, x — nezināms telpas E_n vektors, bet $b\geqslant o$ — dots m-dimensiju telpas vektors.

Pieņemsim, ka matricas A pirmās m kolonnas (vektori) a_j ($j=1, 2, \ldots, m$) veido lineāri neatkarīgu vektoru sistēmu, kas ir viena no m dimensiju telpas bāzēm. Atbilstoši šādam pieņēmumam sadalīsim matricu A blokos, apzīmējot ar B bloku, kurā ietilpst matricas A pirmās m kolonnas, bet ar D — bloku, kurā ietilpst atlikušās $n-m$ kolonnas. Tad izteiksmē $A=[B\,|\,D]$ matrica $B=(a_{ij})_{(m\times m)}$ ir m-tās kārtas nesingulāra kvadrātiska matrica, bet D — taisnstūra matrica ar m rindām un $n-m$ kolonnām. Atbilstoši tam sadalīsim arī vektorus c un x blokos: $c=(c^{B}\,|\,c^{D})$ un $x=(x_B{}^{T}\,|\,x_D{}^{T})^{T}$. Pēc tam uzdevumu (9.3.1), (9.3.2) var pārrakstīt šādi.

Atrast maksimumu lineārai funkcijai

$$z(x)=c^{B}x_B+c^{D}x_D \tag{9.3.4}$$

pie ierobežojumiem

$$Bx_B+Dx_D=b, \tag{9.3.5}$$

$$\left.\begin{aligned} x_B&\geqslant o,\\ x_D&\geqslant o. \end{aligned}\right\} \tag{9.3.6}$$

Reizinot vienādojumu (9.3.5) no kreisās puses ar inverso matricu $\boldsymbol{B}^{-1}$, iegūstam vienādojumu

$$\boldsymbol{x}_B+\boldsymbol{B}^{-1}\boldsymbol{D}\boldsymbol{x}_D=\boldsymbol{B}^{-1}\boldsymbol{b}. \tag{9.3.7}$$

Vektoru $\boldsymbol{x}$, kas apmierina ierobežojumus (9.3.2), (9.3.3), sauc par dotā uzdevuma plānu, bet vektoru $\boldsymbol{x}_A=(\boldsymbol{x}_B{}^{\mathrm{T}} \,\vdots\, \boldsymbol{o}^{\mathrm{T}})^{\mathrm{T}}$ — par šī uzdevuma atbalsta plānu. Katrā atbalsta plānā ietilpst m lineāri neatkarīgu vektoru sistēma. Par optimālo plānu sauc tādu vektoru $\boldsymbol{x}^*$, kurš apmierina uzdevuma ierobežojumus un kuram ir pareiza vienādība $\max z(\boldsymbol{x})=z(\boldsymbol{x}^*)$.

Atbalsta plānu $\boldsymbol{x}_A$ sauc par singulāru, ja kaut viena no vektora $\boldsymbol{x}_B$ koordinātēm ir nulle.

Izsakot $\boldsymbol{x}_B$ no vienādības (9.3.7) un ievietojot izteiksmē (9.3.4), iegūstam

$$z(\boldsymbol{x})=\boldsymbol{c}^B\boldsymbol{B}^{-1}\boldsymbol{b}-(\boldsymbol{c}^B\boldsymbol{B}^{-1}\boldsymbol{D}-\boldsymbol{c}^D)\boldsymbol{x}_D. \tag{9.3.8}$$

Ja pieņemam apzīmējumus

$$\boldsymbol{d}^{\mathrm{T}}=\boldsymbol{c}^B\boldsymbol{B}^{-1}\boldsymbol{D}-\boldsymbol{c}^D \tag{9.3.9}$$

un

$$\boldsymbol{u}^{\mathrm{T}}=\boldsymbol{c}^B\boldsymbol{B}^{-1}, \tag{9.3.10}$$

tad izteiksmi (9.3.8) varam pārrakstīt šādi:

$$z(\boldsymbol{x})=\boldsymbol{u}^{\mathrm{T}}\boldsymbol{b}-\boldsymbol{d}^{\mathrm{T}}\boldsymbol{x}_D. \tag{9.3.11}$$

No izteiksmes (9.3.11) izriet, ka $z(\boldsymbol{x})$ skaitliskā vērtība var būt lielāka par $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{b}$ tad un tikai tad, ja kaut viena no vektora $\boldsymbol{d}$ koordinātēm ir negatīva un šai koordinātei atbilstošā vektora $\boldsymbol{x}_D$ koordināte ir pozitīva. Ja neviena no vektora $\boldsymbol{d}$ koordinātēm nav negatīva, tad, ievietojot izteiksmē (9.3.8) $\boldsymbol{x}_D=\boldsymbol{o}$, iegūstam dotā uzdevuma optimālo atrisinājumu

$$\max z(\boldsymbol{x})=\boldsymbol{c}^B\boldsymbol{B}^{-1}\boldsymbol{b}, \tag{9.3.12}$$

ja

$$\boldsymbol{x}_B=\boldsymbol{B}^{-1}\boldsymbol{b} \text{ un } \boldsymbol{x}_D=\boldsymbol{o}. \tag{9.3.13}$$

Reizinot vienādību (9.3.9) no labās puses ar $(n-m)$-tās kārtas vienības matricu, iegūstam $n-m$ formulas vektora $\boldsymbol{d}$ koordinātu aprēķināšanai:

$$d_j=z_j-c_j, \tag{9.3.14}$$

kur

$$z_j=\boldsymbol{c}^B\boldsymbol{B}^{-1}\boldsymbol{a}_j \tag{9.3.15}$$

un $\boldsymbol{a}_j$ ir matricas $\boldsymbol{A}$ j-tā kolonna. Skaitļus, kuri aprēķināti ar formulām (9.3.14), mēdz saukt par dotā uzdevuma vektoru novērtējumiem. Vektoru novērtējumus var pierakstīt arī šādi:

$$d_j=\boldsymbol{u}^{\mathrm{T}}\boldsymbol{a}_j-c_j, \tag{9.3.16}$$

kur

$$\boldsymbol{u}^{\mathrm{T}}=\boldsymbol{c}^B\boldsymbol{B}^{-1}. \tag{9.3.17}$$

Vektoru novērtējumi kārtējā simpleksa metodes iterāciju tabulā nav nekas cits, kā pārveidotie mērķa funkcijas koeficienti, resp., nulltās rindas elementi, kad dotajā iterācijas tabulā ir nolasāms uzdevuma atbalsta plāns. Bāzes vektoru novērtējumi ir vienādi ar nulli.

Teorēma 1°. *Lineārās programmēšanas uzdevuma plānu kopa ir izliekta kopa.*

Pierādījums. Pieņemsim, ka lineārās programmēšanas uzdevums ir kanoniskajā formā (9.3.1), (9.3.2) un ka šim uzdevumam eksistē vismaz divi plāni $\boldsymbol{x}_1$ un $\boldsymbol{x}_2$. Tad $\boldsymbol{A}\boldsymbol{x}_1=\boldsymbol{b}$, ja $\boldsymbol{x}_1\geqslant\boldsymbol{o}$, un $\boldsymbol{A}\boldsymbol{x}_2=\boldsymbol{b}$, ja $\boldsymbol{x}_2\geqslant\boldsymbol{o}$. Plānu $\boldsymbol{x}_1$ un $\boldsymbol{x}_2$ lineāra kombinācija $\boldsymbol{x}=\lambda\boldsymbol{x}_1+(1-\lambda)\boldsymbol{x}_2$ arī ir plāns, jo $\boldsymbol{A}[\lambda\boldsymbol{x}_1+(1-\lambda)\boldsymbol{x}_2]=\lambda\boldsymbol{b}+(1-\lambda)\boldsymbol{b}=\boldsymbol{b}$. Tā kā saskaņā ar definīciju $\boldsymbol{x}=\lambda\boldsymbol{x}_1+(1-\lambda)\boldsymbol{x}_2$, ja $0\leqslant\lambda\leqslant1$, ir izliekta punktu kopa, tad teorēma ir pierādīta.

Apzīmējot ar $\boldsymbol{a}^i$ $(i=1, 2, \ldots, m)$ lineārās programmēšanas uzdevuma ierobežojumu sistēmas (9.3.2) matricas $\boldsymbol{A}$ rindas, šī uzdevuma izliektā plānu kopa K ģeometriski nozīmē hiperplakņu $\boldsymbol{a}^i\boldsymbol{x}=\boldsymbol{b}_i$ $(i=1, 2, \ldots, m)$ norobežoto pustelpu šķēlumu, ja $\boldsymbol{x}=(x_1, x_2, \ldots, x_n)^{\mathrm{T}}\geqslant\boldsymbol{o}$. Šis šķēlums, ja tas nav tukša kopa, ir izliekts daudzskaldnis (poliedrs) n-dimensiju telpas E_n «nenegatīvajā oktantā». Jēdziens telpas E_n «nenegatīvais oktants» šeit lietots pēc analoģijas ar trīsdimensiju telpas pirmo oktantu Dekarta ortogonālajā koordinātu sistēmā. Pēc tās pašas analoģijas lineāru algebrisku vienādojumu $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{b}$, kurā nezināms ir vektors $\boldsymbol{x}=(x_1, x_2, \ldots, x_n)^{\mathrm{T}}$, sauc par hiperplakni n-dimensiju telpā. Hiperplaknes analogs trīsdimensiju telpā ir plakne, bet divdimensiju telpā tās analogs ir taisne.

Teorēma 2°. *Ja ir zināms, ka vektoru sistēma* $\boldsymbol{a}_j$ $(j=1, 2, \ldots, k)$ *ir lineāri neatkarīga un* $x_1\boldsymbol{a}_1+x_2\boldsymbol{a}_2+\ldots+x_k\boldsymbol{a}_k=\boldsymbol{a}_0$, *kur visi* $x_j\geqslant0$ $(j=1, 2, \ldots, k)$, *tad punkts* $\boldsymbol{x}=(x_1, x_2, \ldots, x_k, 0, \ldots, 0)^{\mathrm{T}}$ *ir izliektas plānu kopas* K *virsotne. Te* $\boldsymbol{x}$ *ir n-dimensiju telpas* E_n *punkts, kura pēdējās* $n-k$ *koordinātes ir vienādas ar nulli.*

Pierādījums. Pieņemsim, ka punkts $\boldsymbol{x}$ nav plānu kopas virsotne. Tā kā $\boldsymbol{x}$ ir plāns, bet pēc pieņēmuma nav kopas virsotne, tad tas var būt kāda nogriežņa $(\boldsymbol{x}_1, \boldsymbol{x}_2)\subseteq K$ punkts un to var izteikt kā punktu $\boldsymbol{x}_1$ un $\boldsymbol{x}_2$ izliektu lineāru kombināciju $\boldsymbol{x}=\lambda\boldsymbol{x}_1+(1-\lambda)\boldsymbol{x}_2$, kur $0<\lambda<1$. Tā kā $\boldsymbol{x}_1$ un $\boldsymbol{x}_2$ ir plāni, tad $\boldsymbol{A}\boldsymbol{x}_1=\boldsymbol{b}$ un $\boldsymbol{A}\boldsymbol{x}_2=\boldsymbol{b}$, kur $\boldsymbol{x}_1=(x_{11}, x_{21}, \ldots, x_{k1}, 0, \ldots, 0)^{\mathrm{T}}$ un $\boldsymbol{x}_2=(x_{12}, x_{22}, \ldots, x_{k2}, 0, \ldots, 0)^{\mathrm{T}}$ jeb $x_{11}\boldsymbol{a}_1+x_{21}\boldsymbol{a}_2+\ldots+x_{k1}\boldsymbol{a}_k=\boldsymbol{a}_0$ un $x_{12}\boldsymbol{a}_1+x_{22}\boldsymbol{a}_2+\ldots+x_{k2}\boldsymbol{a}_k=\boldsymbol{a}_0$. Tas nozīmē, ka vektors $\boldsymbol{a}_0$ ir izteikts ar divām dažādām vektoru sistēmas $\boldsymbol{a}_j$ $(j=1, 2, \ldots, k)$ lineārām kombinācijām, kas nav iespējams, jo vektoru sistēma pēc teorēmas nosacījuma ir lineāri neatkarīga. Tādēļ $\boldsymbol{x}=\boldsymbol{x}_1=\boldsymbol{x}_2$ nevar būt kāda nogriežņa $(\boldsymbol{x}_1, \boldsymbol{x}_2)\subseteq K$ viduspunkts un tas ir kopas K virsotne, ko arī vajadzēja pierādīt.

Teorēma 3°. *Ja* $\boldsymbol{x}=(x_1, x_2, \ldots, x_n)^{\mathrm{T}}$ *ir lineārās programmēšanas plānu kopas* K *virsotne, tad vektori* $\boldsymbol{a}_j$, *kuru koeficienti*

$x_j \geqslant 0$ *lineārajā kombinācijā* $x_1 a_1 + x_2 a_2 + \ldots + x_n a_n = b$ *ir pozitīvi, veido lineāri neatkarīgu vektoru sistēmu.*

No tā izriet, ka gadījumā, ja punkts $x = (x_1, x_2, \ldots, x_n)^T$ ir plānu kopas K virsotne, starp tā koordinātēm $x_j \geqslant 0$ $(j = 1, 2, \ldots, n)$ nav vairāk kā $m < n$ pozitīvu skaitļu.

Pierādījums. Neierobežojot vispārīgumu, pieņemsim, ka pozitīvas ir punkta x pirmās k koordinātes. Tad

$$x_1 a_1 + x_2 a_2 + \ldots + x_k a_k = a_0. \tag{9.3.18}$$

Pieņemsim, ka eksistē tāda vektoru a_j $(j = 1, 2, \ldots, k)$ lineāra kombinācija, kas vienāda ar o:

$$\lambda_1 a_1 + \lambda_2 a_2 + \ldots + \lambda_k a_k = o, \tag{9.3.19}$$

kur vismaz viens no koeficientiem $\lambda_j \neq 0$.

Šāds pieņēmums nozīmē, ka vektoru sistēma a_j $(j = 1, 2, \ldots, k)$ ir lineāri atkarīga. Teorēma būs pierādīta, ja konstatēsim, ka šis pieņēmums ir nepareizs un noved pie pretrunas. Lai to pierādītu, fiksēsim kādu pēc patikas mazu skaitli $\lambda > 0$ un reizināsim ar to vienādības (9.3.19) abas puses. Pieskaitot un atņemot iegūtos rezultātus no izteiksmes (9.3.18), iegūstam

$$\sum_{j=1}^{k} x_j a_j + \lambda \sum_{j=1}^{k} \lambda_j a_j = a_0,$$

$$\sum_{j=1}^{k} x_j a_j - \lambda \sum_{j=1}^{k} \lambda_j a_j = a_0.$$

Tas nozīmē, ka vienādojumam (9.3.18) ir divi atrisinājumi:

$$\left.\begin{aligned} x_1 &= (x_1 + \lambda\lambda_1,\ x_2 + \lambda\lambda_2,\ \ldots,\ x_k + \lambda\lambda_k,\ 0,\ \ldots,\ 0)^T, \\ x_2 &= (x_1 - \lambda\lambda_1,\ x_2 - \lambda\lambda_2,\ \ldots,\ x_k - \lambda\lambda_k,\ 0,\ \ldots,\ 0)^T. \end{aligned}\right\} \tag{9.3.20}$$

Tā kā visi $x_j \geqslant 0$, tad λ var izvēlēties tik mazu, lai vektoru x_1 un x_2 pirmās k koordinātes būtu pozitīvas un šie vektori būtu dotā lineārās programmēšanas uzdevuma plāni. No izteiksmēm (9.3.20) izriet, ka punkts

$$x = \frac{1}{2} x_1 + \frac{1}{2} x_2$$

ir punktu x_1 un x_2 viduspunkts. Tas ir pretrunā ar teorēmas nosacījumu, ka x ir plānu kopas K virsotne. Tātad pieņēmums, ka vektoru sistēma a_j $(j = 1, 2, \ldots, k)$ ir lineāri atkarīga, noved pie pretrunas; līdz ar to teorēma ir pierādīta.

Tā kā m-dimensiju telpā nevar būt tāda lineāri neatkarīga vektoru sistēma, kurā būtu vairāk nekā m vektori, tad tikko pierādītā teorēma apgalvo, ka katrai plānu kopas K virsotnei atbilst m-dimensiju vektoru sistēmas $a_1, a_2, \ldots, a_n$ apakšsistēma ar

$k \leqslant m$ vektoriem, kura ir lineāri neatkarīga. Ja $k=m$, tad tādas lineāri neatkarīgas m vektoru sistēmas eksistence ir pierādīta.

Pieņemsim, ka $k<m$ un ka eksistē lineāri neatkarīga vektoru sistēma $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k, \boldsymbol{a}_{k+1}, \ldots, \boldsymbol{a}_r$. Ja $r<m$, tad $m-r$ vektori ir vektoru $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_r$ lineāras kombinācijas, bet tas ir pretrunā ar m lineāri neatkarīgu vektoru apakšsistēmas eksistenci sistēmā $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$. Tādēļ $r=m$.

Apvienojot pierādītās teorēmas un secinājumus no tām, varam apgalvot, ka $\boldsymbol{x}=(x_1, x_2, \ldots, x_n)^{\mathrm{T}}$ *ir izliektas kopas K virsotne tad un tikai tad, ja šī punkta pozitīvās koordinātes ir koeficienti pie lineāri neatkarīgas vektoru apakšsistēmas izteiksmē* $x_1\boldsymbol{a}_1+x_2\boldsymbol{a}_2+\ldots+x_n\boldsymbol{a}_n=\boldsymbol{a}_0$.

Šāds apgalvojums ļauj izdarīt secinājumu, ka *katram lineārās programmēšanas uzdevuma atbalsta plānam atbilst kāda noteikta plānu kopas K virsotne, un otrādi, katrai plānu kopas virsotnei atbilst viens noteikts atbalsta plāns.*

Bez pierādījuma atzīmēsim, ka jebkuru izliektas kopas punktu var izteikt kā šīs kopas virsotņu izliektu lineāru kombināciju.

T e o r ē m a 4°. *Lineārās programmēšanas uzdevuma mērķa funkcija sasniedz maksimumu kādā plānu kopas K virsotnē. Ja mērķa funkcija sasniedz maksimumu divās vai vairākās virsotnēs, tad tāds pats maksimums ir visos plānu kopas punktos, kuri ir šo virsotņu izliekta lineāra kombinācija.*

P i e r ā d ī j u m s. Pieņemsim, ka dotā lineārās programmēšanas uzdevuma plānu kopa K ir izliekts daudzskaldnis ar p virsotnēm $\boldsymbol{x}_1^0, \boldsymbol{x}_2^0, \ldots, \boldsymbol{x}_p^0$ un apzīmēsim optimālo plānu ar $\boldsymbol{x}^*$, kuram atbilst mērķa funkcijas maksimums:

$$\max z(\boldsymbol{x}) = z(\boldsymbol{x}^*).$$

Pieņemsim, ka $\boldsymbol{x}^*$ nav plānu kopas K virsotne. Tādā gadījumā $\boldsymbol{x}^*$ var izteikt kā šīs kopas virsotņu izliektu lineāru kombināciju:

$$\boldsymbol{x}^* = \sum_{j=1}^{p} \lambda_j \boldsymbol{x}_j^0, \tag{9.3.21}$$

kur $\lambda_j \geqslant 0$ visiem j un $\sum\limits_{j=1}^{p} \lambda_j = 1$.

Ievietojot $\boldsymbol{x}^*$ izteiksmi (9.3.21) mērķa funkcijas $z(\boldsymbol{x})$ izteiksmē, iegūstam

$$z(\boldsymbol{x}^*) = \lambda_1 z(\boldsymbol{x}_1^0) + \lambda_2 z(\boldsymbol{x}_2^0) + \ldots + \lambda_p z(\boldsymbol{x}_p^0). \tag{9.3.22}$$

Vislielāko no p skaitļiem $z(\boldsymbol{x}_j^0)$ $(j=1, 2, \ldots, p)$ apzīmēsim ar μ. Izteiksmes (9.3.22) labā puse nesamazināsies, ja visus $z(\boldsymbol{x}_j^0)$ atvietosim ar μ. Izdarot tādu atvietošanu, iegūstam

$$z(\boldsymbol{x}^*) \leqslant \lambda_1\mu + \lambda_2\mu + \ldots + \lambda_p\mu = \mu,$$

jo visi $\lambda_j \geqslant 0$ un $\lambda_1 + \lambda_2 + \ldots + \lambda_p = 1$.

Tā kā $z(\boldsymbol{x}^*) \geqslant z(\boldsymbol{x})$ visiem $\boldsymbol{x} \in K$, tad $z(\boldsymbol{x}^*) = \mu$. Tātad mērķa funkcija sasniedz maksimumu kādā no plānu kopas K virsotnēm, un līdz ar to teorēmas pirmais izteikums ir pierādīts.

Lai pierādītu teorēmas otro izteikumu, pieņemsim, ka mērķa funkcija sasniedz maksimumu q virsotnēs $\boldsymbol{x}_1^0, \boldsymbol{x}_2^0, \ldots, \boldsymbol{x}_q^0$. Tad $z(\boldsymbol{x}_j^0) = \mu$ visiem $j = 1, 2, \ldots, q$. Ja $\boldsymbol{x}$ ir kāda punktu $\boldsymbol{x}_j^0$ ($j = 1, 2, \ldots, q$) izliekta lineāra kombinācija

$$\boldsymbol{x} = \sum_{j=1}^{q} \alpha_j \boldsymbol{x}_j^0,$$

kur $\alpha_j \geqslant 0$ un $\sum_{j=1}^{q} \alpha_j = 1$, tad

$$z(\boldsymbol{x}) = \alpha_1 z(\boldsymbol{x}_1^0) + \alpha_2 z(\boldsymbol{x}_2^0) + \ldots + \alpha_q z(\boldsymbol{x}_q^0) =$$

$$= \mu(\alpha_1 + \alpha_2 + \ldots + \alpha_q) = \mu,$$

ko arī vajadzēja pierādīt.

Saskaņā ar pierādītajām teorēmām *lineārās programmēšanas uzdevuma optimālais plāns ir kāds no atbalsta plāniem.*

9.4. §. Atbalsta plānu veidošana. Pieņemsim, ka uzdevumā (9.3.1)—(9.3.3) vektoru sistēma $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_m$ ir m-dimensiju telpas bāze un $\boldsymbol{x}_A = (\boldsymbol{x}_B^{\mathrm{T}} \,\vdots\, \boldsymbol{o}^{\mathrm{T}})^{\mathrm{T}}$ ir šī uzdevuma atbalsta plāns.

Tad ir spēkā vienādība

$$x_1 \boldsymbol{a}_1 + x_2 \boldsymbol{a}_2 + \ldots + x_m \boldsymbol{a}_m = \boldsymbol{b}. \tag{9.4.1}$$

Jebkuru vektoru $\boldsymbol{a}_k$ $(m+1 \leqslant k \leqslant n)$, kurš nepieder pie bāzes vektoriem, var izteikt kā bāzes vektoru lineāru kombināciju:

$$y_{1k} \boldsymbol{a}_1 + y_{2k} \boldsymbol{a}_2 + \ldots + y_{mk} \boldsymbol{a}_m = \boldsymbol{a}_k, \tag{9.4.2}$$

kur $\boldsymbol{y}_k = \boldsymbol{B}^{-1} \boldsymbol{a}_k = (y_{1k}, y_{2k}, \ldots, y_{mk})^{\mathrm{T}}$ un $\boldsymbol{B} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_m)^{\mathrm{T}}$ ir no bāzes vektoriem sastādīta, nesingulāra matrica.

Reizināsim vienādību (9.4.2) ar kādu pozitīvu skaitli θ un atņemsim iegūto rezultātu no vienādības (9.4.1). Tad iegūstam

$$\begin{gathered}(x_1 - \theta y_{1k}) \boldsymbol{a}_1 + (x_2 - \theta y_{2k}) \boldsymbol{a}_2 + \ldots + \\ + (x_m - \theta y_{mk}) \boldsymbol{a}_m + \theta \boldsymbol{a}_k = \boldsymbol{b}.\end{gathered} \tag{9.4.3}$$

No izteiksmes (9.4.3) izriet, ka vektors

$$\boldsymbol{x}' = (x_1 - \theta y_{1k};\ x_2 - \theta y_{2k};\ \ldots;\ x_m - \theta y_{mk};\ \theta;\ 0;\ \ldots;\ 0)^{\mathrm{T}} \tag{9.4.4}$$

ir dotā uzdevuma plāns, ja izpildās nosacījumi

$$\left.\begin{aligned} x_i - \theta y_{ik} &\geqslant 0, \\ \theta &> 0. \end{aligned}\right\} \tag{9.4.5}$$

Lai izpildītos nosacījumi (9.4.5), tad skaitlis θ jāizvēlas intervālā

$$0<\theta<\min_i \frac{x_i}{y_{ik}}, \tag{9.4.6}$$

kur attiecības minimumu meklē visiem i, kam atbilstošā koordināte y_{ik} ir pozitīva.

Pieņemsim, ka izteiksmes (9.4.6) labajā pusē atrastais minimālais skaitlis ir

$$\theta_0=\frac{x_r}{y_{rk}},\quad y_{rk}>0. \tag{9.4.7}$$

Ar tādu $\theta=\theta_0$ vektora (9.4.4) r-tā koordināte ir vienāda ar nulli.

Lineārajā algebrā ir pazīstama šāda teorēma. Ja vektoru sistēma $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_m$ ir kādas vektoru sistēmas K bāze un izteiksmē $\boldsymbol{a}_k=x_1\boldsymbol{a}_1+x_2\boldsymbol{a}_2+\ldots+x_m\boldsymbol{a}_m$ koeficients pie kāda no bāzes vektoriem nav nulle, tad, izslēdzot šo vienu vektoru no bāzes un liekot tā vietā vektoru $\boldsymbol{a}_k\in K$, iegūst jaunu sistēmas K bāzi.

Ievērojot to, vektors $\boldsymbol{x}'=\begin{bmatrix}\boldsymbol{x}'_B\\ \hline \boldsymbol{o}\end{bmatrix}$ ir jauns dotā uzdevuma atbalsta plāns, kurā

$$\boldsymbol{x}'_B=\sum_{i=1}^{k-1}(x_i-\theta_0 y_{ik})\boldsymbol{a}_i+\theta_0\boldsymbol{a}_k+\sum_{i=k+1}^{m}(x_i-\theta_0 y_{ik})\boldsymbol{a}_i.$$

Aprēķināsim šim atbalsta plānam atbilstošo mērķa funkcijas vērtību

$$z(\boldsymbol{x}'_B)=\sum_{i=1}^{k-1}c_i(x_i-\theta_0 y_{ik})+c_k\theta_0+\sum_{i=k+1}^{m}c_i(x_i-\theta_0 y_{ik})$$

jeb

$$z(\boldsymbol{x}'_B)=\sum_{i=1}^{m}c_i x_i-\theta_0\left(\sum_{i=1}^{m}c_i y_{ik}-c_k\right). \tag{9.4.8}$$

Tā kā izteiksmē (9.4.8) $\sum\limits_{i=1}^{m}c_i x_i=\boldsymbol{c}^B\boldsymbol{x}_B$ ir vecajai bāzei atbilstošā mērķa funkcijas vērtība $z(\boldsymbol{x})$ un

$$\sum_{i=1}^{m}c_i y_{ik}-c_k=\boldsymbol{c}^B\boldsymbol{y}_k-c_k=\boldsymbol{c}^B\boldsymbol{B}^{-1}\boldsymbol{a}_k-c_k=d_k,$$

tad

$$z(\boldsymbol{x}'_B)=z(\boldsymbol{x}_B)-\theta_0 d_k, \tag{9.4.9}$$

kur $\theta_0>0$ un d_k ir vektora $\boldsymbol{a}_k$ novērtējums. Ja šis novērtējums ir negatīvs, tad no izteiksmes (9.4.9) izriet, ka

$$z(\boldsymbol{x}_B)<z(\boldsymbol{x}'_B).$$

Tas nozīmē, ka, ievedot bāzē jaunu vektoru, kam ir negatīvs novērtējums, aprakstītajā kārtībā iegūst jaunu atbalsta plānu, kuram atbilst palielināta mērķa funkcijas vērtība.

Ja $d_k=0$, tad $z(\boldsymbol{x}'_B)=z(\boldsymbol{x}_B)$. Ja neviena no vektora $\boldsymbol{a}_k$ koordinātēm nav pozitīva, tad nav iespējams atrast pozitīvu θ, kas izslēgtu no bāzes vektoru sistēmas $\boldsymbol{a}_j$ $(j=1, 2, \ldots, m)$ kaut vienu šīs sistēmas vektoru apmaiņā pret $\boldsymbol{a}_k$. Tādā gadījumā lineārās programmēšanas uzdevumam nav optimālā atrisinājuma.

9.5. §. Pirmās dualitātes teorēmas pierādījums. Šī teorēma tika formulēta 5.2. § (teorēma 1°). Lai šo teorēmu pierādītu, formulēsim tiešajam lineārās programmēšanas uzdevumam (9.3.1)—(9.3.3) atbilstošo duālo uzdevumu šādi.

Atrast minimumu lineārai funkcijai

$$w(\boldsymbol{u})=\boldsymbol{b}^{\mathrm{T}}\boldsymbol{u} \tag{9.5.1}$$

pie ierobežojumiem

$$A^{\mathrm{T}}\boldsymbol{u}\geqslant\boldsymbol{c}. \tag{9.5.2}$$

Pieņemsim, ka ar simpleksa metodi ir atrasts tiešā uzdevuma optimālais plāns $\boldsymbol{x}^*$ un ka tas ir atbalsta plāns $\boldsymbol{x}_A=(\boldsymbol{x}_B{}^{\mathrm{T}}\,|\,\boldsymbol{o}^{\mathrm{T}})^{\mathrm{T}}$. Tad

$$z(\boldsymbol{x}^*)=\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^*, \tag{9.5.3}$$

bet saskaņā ar sakarībām (9.3.12) un (9.3.17)

$$z(\boldsymbol{x}^*)=\boldsymbol{c}^{\mathrm{B}}\boldsymbol{B}^{-1}\boldsymbol{b}=\boldsymbol{u}^{\mathrm{T}}\boldsymbol{b}=\boldsymbol{b}^{\mathrm{T}}\boldsymbol{u}. \tag{9.5.4}$$

Tā kā $\boldsymbol{x}^*$ ir tiešā uzdevuma optimālais plāns, tad visi simpleksa metodes pēdējās iterācijas tabulā esošo m-dimensiju vektoru novērtējumi $d_j=\boldsymbol{u}^{\mathrm{T}}\boldsymbol{a}_j-c_j$ ir nenegatīvi, t. i.,

$$\boldsymbol{d}^{\mathrm{T}}=\boldsymbol{u}^{\mathrm{T}}A-\boldsymbol{c}^{\mathrm{T}}\geqslant\boldsymbol{o}, \tag{9.5.5}$$

kur $\boldsymbol{d}^{\mathrm{T}}=(d_1, d_2, \ldots, d_n)$.

No izteiksmes (9.5.5) izriet, ka $A^{\mathrm{T}}\boldsymbol{u}\geqslant\boldsymbol{c}$. Tas savukārt nozīmē, ka $\boldsymbol{u}$ ir duālā uzdevuma plāns. Tā kā no izteiksmēm (9.5.3), (9.5.4) izriet, ka

$$\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^*=\boldsymbol{b}^{\mathrm{T}}\boldsymbol{u}, \tag{9.5.6}$$

tad saskaņā ar 5.2. § pierādīto lemmu 2° $\boldsymbol{u}$ izteiksmē (9.5.6) ir duālā uzdevuma optimālais plāns.

Ja tiešā uzdevuma mērķa funkcija uz augšu nav ierobežota, tad saskaņā ar 5.2. § lemmu 1° jāizpildās nosacījumam $\infty\leqslant w(\boldsymbol{u})$. No šāda nosacījuma izriet, ka duālā uzdevuma plānu kopa ir $\varnothing$. Ja duālā uzdevuma mērķa funkcija uz apakšu nav ierobežota, tad saskaņā ar 5.2. § lemmu 1° jāizpildās nosacījumam $z(\boldsymbol{x})\leqslant-\infty$. No tā izriet, ka tiešā uzdevuma plānu kopa ir $\varnothing$. Ar to 5.2. § teorēma 1° ir pierādīta.

9.6. §. **Singularitātes problēma.** Šo problēmu un ar to saistīto saciklošanās problēmu pieminējām 4.5. § un atzīmējām (bez pierādījuma), ka saciklošanos var novērst, ja, lietojot simpleksa metodi, ievēro 4.2. § aprakstīto galvenās rindas noteikšanas priekšrakstu. Aplūkosim šī priekšraksta matemātisko pamatojumu. Pieņemsim, ka lineārās programmēšanas uzdevumā (9.3.1)—(9.3.3) dažas no vektora $\boldsymbol{b}$ koordinātēm ir nulles. Tādu uzdevumu sauc par *singulāru.* Saciklošanos var novērst, ja nodrošina viennozīmīgu galvenā elementa izvēli katrā simpleksa metodes iterāciju solī.

Pieņemsim, ka $\varepsilon>0$ ir pēc patikas mazs skaitlis, un definēsim vektoru $\boldsymbol{b}_0=(\varepsilon, \varepsilon^2, \varepsilon^3, \dots, \varepsilon^m)^{\mathrm{T}}>\boldsymbol{o}$, kur m vienāds ar matricas $\boldsymbol{A}$ rindu skaitu. Tad vektors $\boldsymbol{b}'=\boldsymbol{b}+\boldsymbol{b}_0>\boldsymbol{o}$. Ja vektoru $\boldsymbol{b}$ izteiksmē (9.3.2) atvieto ar vektoru $\boldsymbol{b}'$, singularitāte ir novērsta. Izteiksmes (9.3.7) labajā pusē tādā gadījumā ir vektors $\boldsymbol{B}^{-1}\boldsymbol{b}'=$ $=\boldsymbol{B}^{-1}(\boldsymbol{b}+\boldsymbol{b}_0)$, kura koordinātes ir m-tās kārtas polinomi

$$\beta_i(\varepsilon)=\beta_{i0}+\beta_{i1}\varepsilon+\beta_{i2}\varepsilon^2+\dots+\beta_{im}\varepsilon^m \qquad (9.6.1)$$
$$(i=1, 2, \dots, m).$$

Skaitļi $\beta_{i1}, \beta_{i2}, \dots, \beta_{im}$ izteiksmē (9.6.1) ir inversās matricas $\boldsymbol{B}^{-1}$ i-tās rindas elementi.

Pieņemsim tālāk, ka simpleksa metodē galvenais elements jāizvēlas j-tajā kolonnā un ka šajā kolonnā ir divi elementi g_r un g_s, no kuriem viens jāizvēlas par galveno. Atbilstošās attiecības ir

$$\begin{aligned} \theta_r&=\frac{\beta_r(\varepsilon)}{g_r}=\theta_{r0}+\theta_{r1}\varepsilon+\theta_{r2}\varepsilon^2+\dots+\theta_{rm}\varepsilon^m,\\ \theta_s&=\frac{\beta_s(\varepsilon)}{g_s}=\theta_{s0}+\theta_{s1}\varepsilon+\theta_{s2}\varepsilon^2+\dots+\theta_{sm}\varepsilon^m. \end{aligned} \qquad (9.6.2)$$

No vienādību (9.6.2) labās puses polinomu koeficientiem var izveidot matricu

$$\begin{bmatrix} \theta_{r0} & \theta_{r1} & \dots & \theta_{rm} \\ \theta_{s0} & \theta_{s1} & \dots & \theta_{sm} \end{bmatrix} \qquad (9.6.3)$$

Matricas (9.6.3) elementiem vismaz vienā kolonnā jābūt atšķirīgiem. Pretējā gadījumā matricai $\boldsymbol{B}$ jābūt singulārai, kas nav iespējams. Tātad, ja izvēlas pietiekami mazu ε, tad $\theta_r\neq\theta_s$. Ja pārlūkojam matricas (9.6.3) rindas no kreisās puses uz labo pusi un atrodam k-tajā kolonnā atšķirīgus elementus, tad

$$\theta_r>\theta_s, \text{ kad } \theta_{rk}>\theta_{sk},$$

un

$$\theta_r<\theta_s, \text{ kad } \theta_{rk}<\theta_{sk},$$

jo pie maziem ε polinoma augstākās kārtas locekļus var atmest.

Singularitāte ģeometriski nozīmē, ka vairākiem atbalsta plāniem atbilst viena un tā pati izliektās plānu kopas virsotne.

9.7. §. Transporta uzdevuma matricas rangs. Aplūkojot transporta uzdevuma ierobežojumu sistēmu (6.1.7), (6.1.8), redzam, ka šajā lineārajā vienādojumu sistēmā visi koeficienti pie nezināmajiem ir vai nu vieninieki, vai nulles. Tas uzskatāmi parādīts zīmējumā 6.1. Sistēmas matricai $\boldsymbol{A}$ ir $m\cdot n$ kolonnas un $m+n$ rindas.

Ja saskaita transporta uzdevuma ierobežojumu sistēmas matricas $\boldsymbol{A}$ pirmās m rindas kā vektorus, tad iegūst $m\cdot n$-dimensiju vektoru $\boldsymbol{e}^{\mathrm{T}}=(1,1,\ldots,1)$, kuram visas koordinātes ir vieninieki. Tādu pašu vektoru $\boldsymbol{e}^{\mathrm{T}}$ iegūst, ja saskaita šīs matricas pēdējās n rindas. Tas nozīmē, ka jebkura no matricas $\boldsymbol{A}$ $m+n$ rindām ir pārējo $m+n-1$ rindu lineāra kombinācija. Tātad matricas $\boldsymbol{A}$ rangs r nav lielāks par $m+n-1$, t. i.,

$$r\leqslant m+n-1. \tag{9.7.1}$$

Nosvītrojot matricas $\boldsymbol{A}$ pēdējo rindu un $s=m\cdot n-(m+n-1)$ kolonnas, paliek $m+n-1$-ās kārtas augšējā trijstūrmatrica

$$\boldsymbol{D}=\begin{bmatrix}
1 & 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1\\
0 & 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0\\
0 & 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0\\
. & . & . & \cdots & . & . & . & \cdots & .\\
0 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0\\
0 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0\\
0 & 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0\\
. & . & . & \cdots & . & . & . & \cdots & .\\
0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1
\end{bmatrix}$$

Matricas $\boldsymbol{D}$ determinants $|D|=1$ ir matricas $\boldsymbol{A}$ $m+n-1$-ās kārtas minors. Tā kā šis minors nevar būt vienāds ar nulli, tad ierobežojumu sistēmas matricas $\boldsymbol{A}$ rangs nevar būt mazāks par $m+n-1$, t. i.,

$$r\geqslant m+n-1. \tag{9.7.2}$$

No sakarībām (9.7.1), (9.7.2) izriet, ka $r=m+n-1$. Tas nozīmē, ka katra klasiskā transporta uzdevuma ierobežojumu sistēmas (6.1.7), (6.1.8) vispārīgā atrisinājuma $\boldsymbol{x}$ koordinātes x_{ij} $(i=1,2,\ldots,m;\ j=1,2,\ldots,n)$ vienmēr var izteikt kā $s=m\cdot n-(m+n-1)$ brīvi izraudzītu parametru t_k $(k=1,2,\ldots,s)$ lineāras funkcijas. Pielīdzinot šos s parametrus nullei, iegūst transporta uzdevuma atbalsta plānu. Tādēļ transporta uzdevuma atbalsta plānu skaits ir galīgs lielums. Šo skaitu var aprēķināt ar 6.3. § sākumā minēto formulu.

10. nodaļa

SIMPLEKSA METODES VARIANTI

10.1. §. Duālā simpleksa metode. Ja, formulējot lineārās programmēšanas uzdevumu vispārīgā veidā, neizvirza prasību, lai visas ierobežojumu sistēmas brīvo locekļu vektora koordinātes būtu nenegatīvas, tad jebkuru lineārās programmēšanas uzdevumu var izteikt šādi.

Atrast maksimumu lineārai funkcijai

$$z(\boldsymbol{x}) = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \tag{10.1.1}$$

pie ierobežojumiem

$$\left.\begin{aligned} A\boldsymbol{x} &\leqslant \boldsymbol{b}, \\ \boldsymbol{x} &\geqslant \boldsymbol{o}, \end{aligned}\right\} \tag{10.1.2}$$

kur $\boldsymbol{c}^{\mathrm{T}} = (c_1, c_2, \ldots, c_n)$ un $\boldsymbol{b} = (b_1, b_2, \ldots, b_m)^{\mathrm{T}}$ ir doti vektori, A — dota $(m \times n)$ tipa matrica un $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$ ir nezināms vektors.

Šādu uzdevumu vienmēr var pierakstīt tabulas formā:

+1	−1	$\boldsymbol{x}^{\mathrm{T}}$	
z	0	$-\boldsymbol{c}^{\mathrm{T}}$	0-tā rinda
$\boldsymbol{y}$	$\boldsymbol{b}$	A	m rindas

(10.1.3)

Tabulā (10.1.3) papildu vektora $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)^{\mathrm{T}}$ visas koordinātes ir nenegatīvas, jo $A\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{b}$. Ja šajā tabulā visas vektoru $\boldsymbol{c}^{\mathrm{T}}$ un $\boldsymbol{b}$ koordinātes ir nenegatīvas, tad optimālais atrisinājums ir $z(\boldsymbol{x}) = 0$ ar $\boldsymbol{x}^{\mathrm{T}} = \boldsymbol{o}^{\mathrm{T}}$ un $\boldsymbol{y}^{\mathrm{T}} = \boldsymbol{b}^{\mathrm{T}}$. Atšķirības labad sauksim līdz šim aplūkoto simpleksa metodi par parasto, bet šeit aplūkojamo — par duālo simpleksa metodi.

Duālajā simpleksa metodē risināšanas algoritmu sadala divos atsevišķos posmos. Pirmajā posmā tabulā (10.1.3) pierakstīto vienādojumu sistēmu transformē tā, lai visas pārveidotā vektora $\boldsymbol{c}$ koordinātes būtu nenegatīvas. Risinājuma otrajā posmā, ja iespējams, panāk, lai visas pārveidotā vektora $\boldsymbol{b}$ koordinātes būtu nenegatīvas. Ja tas nav iespējams, uzdevuma plānu kopa ir tukša.

Risināšanas pirmajā posmā vispirms jāizvēlas tabulas galvenā rinda. Par tādu var izraudzīties jebkuru rindu, ja vien tās krustojumā ar kolonnu, kurā ir vektora $\boldsymbol{c}$ negatīva koordināte, atrodas pozitīvs skaitlis. Galvenais elements ir tas izraudzītās galvenās rindas skaitlis a_{rk}, kuram atbilst $\theta_k = \min\limits_{j \in J} \left| \frac{c_j}{a_{rj}} \right| = \frac{c_k}{a_{rk}}$. Te J ir to indeksu kopa J, kuriem attiecība $\theta_j = \frac{c_j}{a_{rj}} < 0$ ir negatīvs skaitlis.

Ja izdara vienu iterācijas soli ar tādā veidā izraudzītu galveno elementu, brīvo locekļu kolonnā esošā pārveidotā vektora koordināte iegūst pretēju zīmi, bet pārējās šī vektora koordinātes patur savas zīmes. Pirmajā risināšanas posmā vektora koordinātu zīmes var būt dažādas. Kad pirmais risināšanas posms noslēdzies, visas pārveidotā vektora $\boldsymbol{c}$ koordinātes c_j ir nenegatīvas.

Otrajā posmā par galveno rindu jāizvēlas tā rinda, kurai atbilst kāda no pārveidotā vektora $\boldsymbol{b}'$ negatīvajām koordinātēm. Galvenais elements šajā risināšanas posmā ir tas izraudzītās r-tās rindas elements a'_{rk}, kuram atbilst $\theta'_k = \min_{j \in J} \left| \frac{c'_j}{a'_{rj}} \right| = \frac{c'_k}{a'_{rk}}$. Te J ir to indeksu kopa, kuriem attiecība $\theta_j = \frac{c_j}{a_{rj}} \leqslant 0$ ir nepozitīvs skaitlis.

P i e m ē r s 10.1.1°. Atrast maksimumu lineārai funkcijai

$$z = -5x_1 - 12x_2 + 5x_3$$

pie ierobežojumiem

$$\begin{aligned} x_1 + 10x_2 - 4x_3 &\leqslant 11, \\ -2x_1 - 4x_2 + 2x_3 &\leqslant 8, \\ -3x_1 + 4x_2 - 2x_3 &\leqslant -16, \\ x_1, x_2, x_3 &\geqslant 0. \end{aligned}$$

Pārrakstīsim doto uzdevumu tabulas formā:

$+1$	-1	x_1	x_2	x_3	(I)
z	0	5	12	-5	0
y_1	11	1	10	-4	1
y_2	8	-2	-4	$\boxed{2}$	2
y_3	-16	-3	4	-2	3
σ	2	0	21	-10	4

(10.1.4)

Nulltajā rindā ir viens negatīvs koeficients (-5). Šī koeficienta kolonnā pozitīvs skaitlis ir tikai 2. rindā. Tādēļ uzdevuma risināšanas pirmajā posmā galvenais elements ir tabulā (10.1.4) ar taisnstūri apvilktais skaitlis 2. Izpildot ar šo elementu vienu iterāciju soli, iegūstam tabulu (10.1.5):

$+1$	-1	x_1	x_2	y_2	(II)
z	20	0	2	5/2	0
y_1	27	-3	2	2	1
x_3	4	-1	-2	1/2	2
y_3	-8	$\boxed{-5}$	0	1	3
σ	42	-10	1	5	4

(10.1.5)

Tā kā nulltajā rindā nav negatīvu skaitļu, tad risināšanas pirmais posms ir noslēdzies.

Tabulas (10.1.5) brīvo locekļu kolonnā ir tikai viens negatīvs skaitlis (-8) un tā rindā savukārt ir tikai viens negatīvs skaitlis (-5), tāpēc šis skaitlis arī jāizvēlas par galveno elementu. Izpildot nākošo iterāciju soli, iegūstam tabulu (10.1.6):

$+1$	-1	y_3	x_2	y_2	(III)
z	20	0	2	5/2	0
y_1	159/5	$-3/5$	2	7/5	1
x_3	28/5	$-1/5$	-2	3/10	2
x_1	8/5	$-1/5$	0	$-1/5$	3
σ	58	-2	1	3	4

(10.1.6)

Tabulā (10.1.6) parastajā kārtībā nolasāms dotā uzdevuma optimālais atrisinājums:

$$z_{\max}=20, \text{ ja } x_1=1{,}6,\ x_2=0,\ x_3=5{,}6.$$

10.2. §. Modificētā simpleksa metode. Pieņemsim, ka lineārās programmēšanas uzdevums dots normālformā:

$$\left.\begin{array}{l} z(\boldsymbol{x})=\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}\rightarrow\max, \\ \boldsymbol{A}\boldsymbol{x}\leqslant\boldsymbol{b}\ (\boldsymbol{b}\geqslant\boldsymbol{o}), \\ \boldsymbol{x}\geqslant\boldsymbol{o}. \end{array}\right\} \tag{10.2.1}$$

Tādu uzdevumu, kā zināms, var vienmēr pierakstīt ar tabulu (10.1.3). Pārveidojot šo uzdevumu kanoniskajā formā, to varam izteikt šādi:

$$\left.\begin{array}{l} z(\boldsymbol{x})=\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}+\boldsymbol{o}^{\mathrm{T}}\boldsymbol{y}\rightarrow\max, \\ \boldsymbol{A}\boldsymbol{x}+\boldsymbol{E}\boldsymbol{y}=\boldsymbol{b}\ (\boldsymbol{b}\geqslant\boldsymbol{o}), \\ \boldsymbol{x}\geqslant\boldsymbol{o},\ \boldsymbol{y}\geqslant\boldsymbol{o}, \end{array}\right\} \tag{10.2.2}$$

kur E ir m-tās kārtas vienības matrica un $\boldsymbol{y}=(y_1,\ldots,y_m)^{\mathrm{T}}$ ir papildu nezināmo vektors.

No uzdevuma (10.2.2) koeficientiem veidojas blokmatrica

$$\boldsymbol{L}_0=\left[\begin{array}{c|c|c} 0 & -\boldsymbol{c}^{\mathrm{T}} & \boldsymbol{o}^{\mathrm{T}} \\ \hline \boldsymbol{b} & \boldsymbol{A} & \boldsymbol{E} \end{array}\right] \tag{10.2.3}$$

No matricas $\boldsymbol{L}_0$ kolonnām var sastādīt vienības blokmatricu

$$\boldsymbol{B}_0=\left[\begin{array}{c|c} 1 & \boldsymbol{o}^{\mathrm{T}} \\ \hline \boldsymbol{o} & \boldsymbol{E} \end{array}\right] \tag{10.2.4}$$

Matricas B_0 kolonnas (vektori) ir $m+1$-dimensiju telpas bāze. Apmainot šīs bāzes vektorus pēc s iterāciju soļiem ar matricas L_0 citiem vektoriem, iegūst jaunu bāzi

$$B_s=\left[\begin{array}{c|c} 1 & -d^T \\ \hline o & D \end{array}\right] \tag{10.2.5}$$

Tad matricas L_0 kolonnas (vektorus) jaunajā bāzē nosaka matrica

$$L_s=B_s^{-1}L_0. \tag{10.2.6}$$

Tā kā

$$B_s^{-1}=\left[\begin{array}{c|c} 1 & d^T D^{-1} \\ \hline o & D^{-1} \end{array}\right] \tag{10.2.7}$$

tad, reizinot matricu L_0, kas definēta ar (10.2.3), no kreisās puses ar B_s^{-1}, iegūstam

$$L_s=\left[\begin{array}{c|c|c} d^T D^{-1}b & d^T D^{-1}A-c^T & d^T D^{-1} \\ \hline D^{-1}b & D^{-1}A & D^{-1} \end{array}\right] \tag{10.2.8}$$

Atbilstošā simpleksa metodes iterāciju tabula pēc s soļiem tātad ir šāda:

$+1$	-1	x^T	y^T
z	$d^T D^{-1}b$	$dD^{-1}A-c^T$	$d^T D^{-1}$
u	$D^{-1}b$	$D^{-1}A$	D^{-1}

(10.2.9)

Vektora u koordinātes ir vektoru x^T un y^T koordinātu apakškopa.

Tabula (10.2.9) atšķiras no parastās simpleksa metodes iterāciju tabulas tikai ar to, ka šajā tabulā parādās arī visi bāzes vektori (kolonnas).

Parastajā simpleksa metodē katru matricas L_s elementu aprēķina pēc pilna matricas L_{s-1} elementu pieraksta. Modificētajā simpleksa metodē nepievērš uzmanību visiem matricas L_s elementiem, bet tikai tām matricas L_s kolonnām (vektoriem), kurām pieder vektora y^T koordinātes. Šie vektori ir inversās matricas B_s^{-1} kolonnas. Šo vektoru koordinātes aprēķina parastajā kārtībā, pārējos matricas L_s elementus var aprēķināt ar formulu (10.2.6), ja vispār ir nepieciešams šos elementus aprēķināt. Parasti tādas nepieciešamības nav.

Rezumējot teikto, formulēsim šādu modificētās simpleksa metodes algoritmu.

1. solis. Pieraksta uzdevumu (10.2.2) sākuma tabulas formā:

$+1$	-1	x^T	y^T
z	0	$-c^T$	o^T
y	b	A	E

(10.2.10)

2. solis. Izvēlas galveno elementu, kas var būt vektora $-c^T$ vismazākā negatīvā koordināte. Praksē bieži galveno elementu izvēlas tajā kolonnā, kurā sastopama vektora $-c^T$ pirmā negatīvā koordināte. Pēc galvenā elementa izvēles aprēķina to vektoru koordinātes, kuri pieder vektoram y^T:

$+1$	-1	x^T	y^T
z			$d^T D^{-1}$
u			D^{-1}

(10.2.11)

3. solis. Aprēķina brīvo locekļu kolonnas elementus un nulltās rindas elementus:

$+1$	-1	x^T	y^T
z	$d^T D^{-1} b$	$d^T D^{-1} A - c^T$	$d^T D^{-1}$
u	$D^{-1} b$		D^{-1}

(10.2.12)

4. solis. Ja nulltajā rindā visi vektoru x^T un y^T koeficienti ir nenegatīvi, tad ir iegūts optimālais plāns $\max z(x) = d^T D^{-1} b$ pie $u = D^{-1} b$. Pretējā gadījumā, kad nulltās rindas j-tajā kolonnā ir negatīvs skaitlis, iegūst šīs kolonnas elementus, reizinot matricas A j-to kolonnu Ae_k no kreisās puses ar matricu D^{-1}.

2., 3. un 4. soli atkārto tik daudz reižu, kamēr iegūst optimālo plānu vai arī pārliecinās, ka dotā uzdevuma plānu kopa ir tukša.

Modificētās simpleksa metodes priekšrocība ir tā, ka matricas L_s elementu aprēķināšanai izmanto uzdevuma sākotnējās informācijas datus.

Šī iemesla dēļ samazinās noapaļošanas kļūdu ietekme. Izņēmums ir inversās matricas D^{-1} aprēķināšana, bet par tās elementu pareizību katrā solī ir iespējams pārliecināties, sastādot reizinājumu DD^{-1}. Šim reizinājumam jābūt identiski vienādam ar vienības matricu E. Bez tam izrādās, ka, risinot lineārās programmēšanas uzdevumu ar modificēto simpleksa metodi elektroniskajā skaitļošanas mašīnā, vispār vajag mazāk operatīvās atmiņas šūnu nekā tad, ja uzdevumu, kam ir tāds pats sākotnējās informācijas apjoms, risinātu ar parasto simpleksa metodi. Modificētajai simpleksa metodei ir īpašas priekšrocības, ja matricas A kolonnu skaits ievērojami pārsniedz rindu skaitu.

Ja lineārās programmēšanas uzdevums nav dots normālformā, tad, kā zināms (sk. 2.2. § punktu 3°), šādu uzdevumu vienmēr var pārveidot modificētajā normālformā.

Tādā gadījumā uzdevumu var formulēt šādi:

$$\begin{aligned} &z(\boldsymbol{x}) = \boldsymbol{c}^T\boldsymbol{x} \to \max, \\ &A_1\boldsymbol{x}_1 \leqslant \boldsymbol{b}_1 \ (\boldsymbol{b}_1 \geqslant \boldsymbol{o}), \\ &A_2\boldsymbol{x}_2 = \boldsymbol{b}_2 \ (\boldsymbol{b}_2 \geqslant \boldsymbol{o}), \\ &\boldsymbol{x} = (\boldsymbol{x}_1 \mid \boldsymbol{x}_2)^T \geqslant \boldsymbol{o}. \end{aligned} \qquad (10.2.13)$$

No uzdevuma (10.2.13) koeficientiem var izveidot blokmatricu

$$L_0 = \left[\begin{array}{c|c|c} -s_0 & -\boldsymbol{s}^T & \boldsymbol{o}^T \\ \hline \boldsymbol{b} & A & E \end{array}\right]$$

kur $\boldsymbol{b} = (0 \mid \boldsymbol{b}_1 \mid \boldsymbol{b}_2)^T$; $A = (-\boldsymbol{c}^T \mid A_1 \mid A_2)^T$, $s_0 = \boldsymbol{e}^T\boldsymbol{b}_2$, $\boldsymbol{s} = \boldsymbol{e}^T A_2$; ar $\boldsymbol{e}^T$ apzīmēts vektors (rinda), kuram visas koordinātes ir vieninieki.

Ja lieto šādu simboliku, uzdevums reducējas uz aplūkoto normālformā dota uzdevuma risināšanas algoritmu, tikai uzdevuma risinājumam ir divi posmi. Pirmajā posmā maksimizē sekundāro mērķa funkciju un atrod atbalsta plānu, ja vien šī uzdevuma plānu kopa nav tukša.

Otrajā posmā maksimizē dotā uzdevuma mērķa funkciju, ievērojot, protams, tos šīs funkcijas koeficientu pārveidojumus, kas iegūti pirmajā posmā.

P i e m ē r s 10.2.1°. Lietojot modificēto simpleksa metodi, atrast maksimumu lineārai funkcijai

$$z = 2x_1 - 2x_2 - x_3 + x_4$$

pie ierobežojumiem

$$\begin{aligned} 2x_1 + 4x_2 + x_3 \quad &\leqslant 100, \\ x_1 \quad + 5x_3 \quad &\leqslant 200, \\ x_2 + 4x_3 + 2x_4 &= 200, \\ x_1 + x_2 \quad + x_4 &= 150, \\ x_j \geqslant 0 \ (j = 1, 2, 3, 4). \end{aligned}$$

Šajā gadījumā uzdevums dots modificētajā normālformā un tā risināšanas pirmajā posmā jāmaksimizē sekundārā mērķa funkcija $s = -350 + x_1 + 2x_2 + 4x_3 + 3x_4$, lai iegūtu dotā uzdevuma atbalsta plānu.

Ievedot papildu nezināmos y_1 un y_2 un mākslīgos palīgnezināmos $\overline{y}_3$ un $\overline{y}_4$, sastādām sākuma tabulu:

(10.2.14)

$+1$	-1	x_1	x_2	x_3	x_4	y_1	y_2	$\overline{y}_3$	$\overline{y}_4$	
s	-350	-1	-2	-4	-3	0	0	0	0	M
z	0	-2	2	1	-1	0	0	0	0	0
y_1	100	2	4	1	0	1	0	0	0	1
y_2	200	1	0	5	0	0	1	0	0	2
$\overline{y}_3$	200	0	1	4	2	0	0	1	0	3
$\overline{y}_4$	150	1	1	0	1	0	0	0	1	4

Tabulu (10.2.14) īsuma labad sauksim par *lielo tabulu*. Par mazo sauksim tādu tabulu, kurā ietilpst nesingulāra kvadrātiska matrica, kas no kreisās puses ir reizināma ar lielās tabulas matricu. Sākumā tā ir vienības matrica. *Mazajā tabulā*, ar kuru izpilda modificētās simpleksa metodes iterāciju soļus, katrā solī pievieno atbilstoši pārveidotu brīvo locekļu kolonnu un galveno kolonnu. Mūsu piemērā pirmā mazā tabula ir (10.2.15):

I	s	z	y_1	y_2	$\overline{y}_3$	$\overline{y}_4$	-1	x_4	
s	1	0	0	0	0	0	−350	−3	M
z	0	1	0	0	0	0	0	−1	0
y_1	0	0	1	0	0	0	100	0	1
y_2	0	0	0	1	0	0	200	1	2
$\overline{y}_3$	0	0	0	0	1	0	200	$\boxed{2}$	3
$\overline{y}_4$	0	0	0	0	0	1	150	1	4
σ	0	0	0	0	0	0	299	−1	5

(10.2.15)

Izpildot parastajā kārtībā vienu iterāciju soli, iegūst jaunās mazās tabulas matricu un šai tabulai atbilstošo brīvo locekļu kolonnu:

II	s	z	y_1	y_2	$\overline{y}_3$	$\overline{y}_4$	-1	x_1
s	1	0	0	0	1,5	0	−50	−1
z	0	1	0	0	0,5	0	100	−2
y_1	0	0	1	0	0	0	100	2
y_2	0	0	0	1	−0,5	0	100	1
x_4	0	0	0	0	0,5	0	100	0
$\overline{y}_4$	0	0	0	0	−0,5	1	50	$\boxed{1}$
σ	0	0	0	0	0,5	0	399	0

(10.2.16)

Lai noteiktu jaunai tabulai (10.2.16) pievienojamo galveno kolonnu, nav nepieciešams aprēķināt visus vektoru novērtējumus, t. i., visus rindas M elementus. Pietiek atrast tādu j, kuram atbilstošais skalārais reizinājums $\boldsymbol{m}^2\boldsymbol{l}_j$ ir negatīvs. Te ar $\boldsymbol{m}^2$ apzīmēta r-tā iterāciju soļa mazās tabulas matricas pirmā rinda, bet ar $\boldsymbol{l}_j$ — jebkura lielās tabulas kolonna, izņemot mākslīgajiem nezināmajiem atbilstošās koeficientu kolonnas.

Tā kā vektoru $\boldsymbol{m}^2=(1; 0; 0; 0; 1{,}5; 0)$ un $l_1=(-1; -2; 2; 1; 0; 1)^{\mathrm{T}}$ skalārais reizinājums ir -1, tad nezināmā x_1 koeficientu kolonna pēc tās reizināšanas no kreisās puses ar mazās tabulas matricu pievienojama tabulai (10.2.16) kā galvenā kolonna.

Ievērosim, ka tabulas (10.2.16) brīvo locekļu kolonna arī iegūstama, reizinot lielās tabulas brīvo locekļu kolonnu no kreisās puses ar mazās tabulas matricu.

Nākošajā iterāciju solī iegūst jaunu mazo tabulu (10.2.17):

III	s	z	y_1	y_2	$\bar{y}_3$	$\bar{y}_4$	-1
s	1	0	0	0	1	1	0
z	0	1	0	0	$-0{,}5$	2	200
y_1	0	0	1	0	1	-2	0
y_2	0	0	0	1	0	-1	50
x_4	0	0	0	0	0,5	0	100
x_1	0	0	0	0	$-0{,}5$	1	50
σ	0	0	0	0	0,5	0	399

(10.2.17)

Tā kā sekundārās mērķa funkcijas maksimums saskaņā ar tabulu (10.2.17) ir nulle, tad ir atrasts dotā uzdevuma atbalsta plāns un modificētās simpleksa metodes pirmais posms aplūkojamā piemērā ir noslēdzies.

Otrajā posmā kā lielajā, tā mazajā tabulā atmet sekundārās mērķa funkcijas rindu. Mazajā tabulā bez tam atmet arī pirmo kolonnu. Pēc tam nosaka un pievieno mazajai tabulai galveno kolonnu. Iterāciju soļus atkārto tik daudz reižu, cik nepieciešams, lai iegūtu optimālo plānu vai arī lai pārliecinātos, ka mērķa funkcija dotajā apgabalā ir neierobežota. Aplūkojamā piemērā tālāk seko tabulas (10.2.18) un (10.2.19) (sk. 162. lpp.).

IV	z	y_1	y_2	$\bar{y}_3$	$\bar{y}_4$	-1	x_3
z	1	0	0	$-0{,}5$	2	200	-1
y_1	0	1	0	1	-2	0	$\boxed{5}$
y_2	0	0	1	0	-1	50	5
x_4	0	0	0	0,5	0	100	2
x_1	0	0	0	$-0{,}5$	1	50	-2
σ	0	0	0	$-0{,}5$	-1	399	8

(10.2.18)

Vektora $\boldsymbol{m}^5 = (1; 0{,}2; 0; -0{,}3; 1{,}6)$ skalārie reizinājumi ar lielās tabulas kolonnām ir skaitļi 0; 4,1; 0; 0; 0,2; 0; $-0{,}3$; 1,6. Šajā skaitļu kopā vienīgais negatīvais skaitlis ir $-0{,}3$, bet tas ir koeficients pie mākslīgā nezināmā $\bar{y}_3$, kas nevar būt bāzes mainīgais. Tādēļ ar tabulu (10.2.19) ir atrasts dotā uzdevuma optimālais atrisinājums: $\max z=200$, ja $x_1=50$, $x_2=0$, $x_3=0$, $x_4=100$.

V	z	y_1	y_2	$\overline{y_3}$	$\overline{y_4}$	-1
z	1	0,2	0	$-0,3$	1,6	200
x_3	0	0,2	0	0,2	$-0,4$	0
y_2	0	-1	1	-1	1	50
x_4	0	$-0,4$	0	0,1	0,8	100
x_1	0	0,4	0	$-0,1$	0,2	50
σ	0	$-1,6$	0	$-2,1$	2,2	399

(10.2.19)

P i e m ē r s 10.2.2°. Lietojot modificēto simpleksa metodi, atrast atbalsta plānu lineārai nevienādību sistēmai

$$\begin{aligned} 5x_1-2x_2+x_3+2x_4-x_5&\leqslant 3,\\ x_1+x_2-2x_3+3x_4-2x_5&\geqslant 1,\\ -3x_1+x_2-x_3+x_4-3x_5&\leqslant 3,\\ x_j\geqslant 0\ (j=1,\dots,5).& \end{aligned}$$

Pārveidojot doto nevienādību sistēmu kanoniskajā formā ar attiecīgajiem papildu un mākslīgajiem nezināmajiem, sastādām lielo tabulu ar sekundāro mērķa funkciju, kura šajā gadījumā ir $s=-1+x_1+x_2-2x_3+3x_4-2x_5+y_2$:

(10.2.20)

$+1$	-1	x_1	x_2	x_3	x_4	x_5	y_2	y_1	$\overline{y_2}$	y_3
s	-1	-1	-1	2	-3	2	1	0	0	0
y_1	3	5	-2	1	2	-1	0	1	0	0
$\overline{y_2}$	1	1	1	-2	3	-2	-1	0	1	0
y_3	3	-3	1	-1	1	-3	0	0	0	1

Šai lielajai tabulai atbilstošā pirmā mazā tabula ir šāda:

I	s	y_1	$\overline{y_2}$	y	-1	x_4
s	1	0	0	0	-1	-3
y_1	0	1	0	0	3	2
$\overline{y_2}$	0	0	1	0	1	$\boxed{3}$
y_3	0	0	0	1	3	1
σ	0	0	0	0	5	2

(10.2.21)

Izpildot vienu iterāciju soli, iegūstam

II	s	y_1	$\overline{y_2}$	y_3	-1
s	1	0	1	0	0
y_1	0	1	$-2/3$	0	7/3
x_4	0	0	1/3	0	1/3
y_3	0	0	$-1/3$	1	8/3
σ	0	0	$-2/3$	0	13/3

(10.2.22)

Tā kā jau pēc pirmā iterāciju soļa ir iegūts sekundārās mērķa funkcijas maksimums, kas vienāds ar 0, tad dotajai nevienādību sistēmai ir pēdējā iterāciju tabulā nolasāmais atbalsta plāns: $x_1=x_2=x_3=x_5=0$, $x_4=1/3$, ja $y_1=7/3$, $y_2=\overline{y_2}=0$ un $y_3=8/3$.

III DAĻA

NELINEĀRĀS PROGRAMMĒŠANAS UN SPĒĻU TEORIJAS ELEMENTI

11. nodaļa

NELINEĀRĀS PROGRAMMĒŠANAS VISPĀRĪGIE JĒDZIENI

11.1. §. Dažādi matemātiskās programmēšanas uzdevumu tipi. Matemātiskās programmēšanas uzdevumu vispārīgā veidā var formulēt šādi.

Atrast vislielāko vai vismazāko vērtību reālai n argumentu mērķa funkcijai

$$f(x_1, x_2, \ldots, x_n) \qquad (11.1.1)$$

pie m ierobežojumiem

$$G_i(x_1, x_2, \ldots, x_n) \leqslant b_i \quad (i=1, 2, \ldots, p<h), \qquad (11.1.2)$$

$$G_i(x_1, x_2, \ldots, x_n) \geqslant b_i \quad (i=p+1, \ldots, h<m), \qquad (11.1.3)$$

$$G_i(x_1, x_2, \ldots, x_n) = b_i \quad (i=h+1, \ldots, m), \qquad (11.1.4)$$

kur $b_i \geqslant 0$ ir doti reāli skaitļi.

Tā kā naturālie skaitļi n un m nav savstarpēji saistīti, tad nav izslēgts arī tāds gadījums, ka ierobežojumu nav. Parasti matemātiskās programmēšanas uzdevumos vismaz viena daļa no nezināmajiem x_j $(j=1, 2, \ldots, n)$ nevar būt negatīvi skaitļi. Ierobežojumi var būt arī tāda rakstura, ka visiem nezināmajiem vai arī daļai nezināmo jābūt diskrētiem, piemēram, veseliem skaitļiem. Iespējams arī tāds gadījums, ka visi nezināmie x_j vai arī daļa no tiem ir citu mainīgu lielumu funkcijas. Ja bez izņēmuma visas sakarības (11.1.1)—(11.1.4) ir lineāras, uzdevumu var atrisināt ar kādu no I vai II daļā aplūkotajiem simpleksa metodes variantiem. Ja kaut viena no sakarībām (11.1.1)—(11.1.4) ir nelineāra vai arī pastāv nezināmo diskrētības nosacījums, uzdevumu (11.1.1)—(11.1.4) sauc par *nelineārās programmēšanas uzdevumu.*

Starp nelineārās programmēšanas uzdevumiem īpaša nozīme ir t. s. *klasiskajiem optimizācijas uzdevumiem.* Tie ir tādi nelineārās programmēšanas uzdevumi, kuru ierobežojumu sistēmā (11.1.2)—(11.1.4) nav nevienādību, bet var būt tikai (11.1.4) tipa ierobežojumi, turklāt ierobežojumu skaitam jābūt vismaz par vienu mazākam nekā nezināmo skaits n, mērķa funkcijai un visos ierobežojumos dotajām funkcijām jābūt nepārtrauktām un vismaz divas reizes diferencējamām. Klasisko optimizācijas uzdevumu īpatnība ir tā, ka šie uzdevumi vismaz principā ir atrisināmi ar diferenciālrēķinos pazīstamām relatīvo ekstrēmu atrašanas metodēm. Šīs metodes tomēr saistītas ar tādām skaitļošanas grūtībām, kuru pārvarēšana bieži vien kļūst praktiski neiespējama. Tādēļ klasiskajiem optimizācijas uzdevumu atrisināšanas paņēmieniem ir maza praktiskā nozīme ekonomiski matemātisko metožu pielietojumos. Tie paliek nozīmīgi kā citu, efektīvāku metožu teorētiskās analīzes pamati.

Definīcija 1°. *Funkciju* $f(x_1, x_2, \ldots, x_n)$, resp., $f(\boldsymbol{x})$, *kuras definīcijas apgabals ir izliekta punktu* $\boldsymbol{x}$ *kopa* K, *sauc par izliektu šajā apgabalā, ja jebkuriem diviem punktiem* $\boldsymbol{x}_1 \in K$ *un* $\boldsymbol{x}_2 \in K$ *un jebkuram* λ *no intervāla* $0 \leqslant \lambda \leqslant 1$ *izpildās nevienādība*

$$f[\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2] \leqslant \lambda f(\boldsymbol{x}_1) + (1-\lambda) f(\boldsymbol{x}_2). \qquad (11.1.5)$$

Definīcija 2°. *Funkciju* $f(\boldsymbol{x})$ *sauc par ieliektu izliektā definīcijas apgabalā* K, *ja jebkuriem diviem punktiem* $\boldsymbol{x}_1 \in K$ *un* $\boldsymbol{x}_2 \in K$ *un jebkuram* λ *no intervāla* $0 \leqslant \lambda \leqslant 1$ *izpildās nevienādība*

$$f[\lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2] \geqslant \lambda f(\boldsymbol{x}_1) + (1-\lambda) f(\boldsymbol{x}_2). \qquad (11.1.6)$$

Izliektu funkciju $f(\boldsymbol{x})$ ir izdevīgi saukt par *izliektu uz apakšu*, bet ieliektu funkciju — par *izliektu uz augšu*.

Uz apakšu izliektas viena argumenta funkcijas grafiskā attēla loks atrodas zem hordas, kas šo loku savelk, bet uz augšu izliektas funkcijas loks — virs savelkošās hordas. Lineāras funkcijas var uzlūkot kā izliektas gan uz augšu, gan uz apakšu visā to definīcijas apgabalā.

Nelineārās programmēšanas apakšnodaļu, kura nodarbojas ar izliektu funkciju minimizāciju vai ieliektu funkciju maksimizāciju, sauc par *izliekto programmēšanu.*

Izliektās programmēšanas uzdevumu risināšanai pastāv vairākas skaitliskās metodes, kas bazējas uz priekšstatiem, kuri saistās ar *visstraujākā krituma* vai *visstraujākā kāpuma (gradienta) virzieniem* mērķa funkcijas definīcijas apgabalā.

Pat tādiem nelineārās programmēšanas speciālgadījumiem, kuros ierobežojumu sistēma (11.1.2)—(11.1.4) ir lineāra, praktiski noderīgi skaitliskās atrisināšanas algoritmi pastāv tikai speciāliem mērķa funkcijas tipiem. Viens no tādiem nelineārās programmēšanas veidiem ir t. s. *kvadrātiskā programmēšana.* Kvadrātiskās programmēšanas uzdevumu vispārīgā veidā var formulēt šādi.

Atrast vislielāko vai vismazāko vērtību funkcijai

$$f(\boldsymbol{x}) = \boldsymbol{c}^T \boldsymbol{x} + \boldsymbol{x}^T \boldsymbol{D} \boldsymbol{x}$$

pie ierobežojumiem

$$Ax=b,$$
$$x \geqslant o,$$

kur A ir dota $(m \times n)$ tipa reālu skaitļu matrica, D — simetriska n-tās kārtas matrica, c^T un $b \geqslant o$ — doti vektori, x — nezināms n-dimensiju vektors.

Definīcija 3°. *Funkciju $f(x_1, x_2, \ldots, x_n)$ sauc par separablu, ja pastāv identitāte*

$$f(x_1, x_2, \ldots, x_n) \equiv \sum_{i=1}^{n} f_i(x_i),$$

kur $f_i(x_i)$ ir viena argumenta funkcijas.

Nelineārās programmēšanas uzdevumus, kuru mērķa funkcija un ierobežojumu sistēmā dotās funkcijas ir separablas, sauc par *separablās programmēšanas uzdevumiem.*

Daudzos praktiski nozīmīgos uzdevumos sastopamie parametri nav determinēti, bet ir gadījuma lielumi. Tādus uzdevumus sauc par *stohastiskās programmēšanas uzdevumiem.*

Pie nelineārās programmēšanas uzdevumiem pieder arī tādi matemātiskās programmēšanas uzdevumi, kuru mērķa funkcija un ierobežojumu sistēmas izteiksmes ir lineāras, bet pastāv nosacījums, ka visiem vai vismaz dažiem nezināmajiem optimālajā atrisinājumā jābūt veseliem skaitļiem. Tādus uzdevumus sauc par *diskrētās programmēšanas uzdevumiem.* Šādus uzdevumus var atrisināt, atkārtoti pielietojot simpleksa metodi, ja soli pa solim attiecīgi pārveido lineāro ierobežojumu sistēmu. Kā tādi praktiski nozīmīgi uzdevumi atrisināmi, to aplūkosim 12. nodaļā.

11.2. §. Jēdziens par globālo un lokālo maksimumu un minimumu. Definīcija 1°. *Saka, ka funkcija $f(x)$, kuras definīcijas apgabals ir noslēgta lineārās n-dimensiju Eiklīda telpas E_n punktu kopa K, sasniedz globālu maksimumu punktā $x^* \in K$, ja jebkuram $x \in K$ izpildās nevienādība $f(x) \leqslant f(x^*)$. Funkcija sasniedz globālu minimumu, ja $f(x) \geqslant f(x^*)$.*

Ja punktu kopa K ir slēgta un ierobežota, tad nepārtraukta funkcija $f(x)$ noteikti sasniedz globālu maksimumu (minimumu) vienā vai vairākos punktos $x \in K$. Ja kopa K ir neierobežota, tad funkcijai $f(x)$ var arī nebūt globāla maksimuma (minimuma) nevienā punktā $x \in K$.

Definīcija 2°. *Saka, ka funkcija $f(x)$, kura definēta visos punktos kāda noteikta telpas E_n punkta x_0 δ-apkārtnē, sasniedz šajā punktā lokālu maksimumu, ja eksistē tāds skaitlis ε, $0<\varepsilon<\delta$, ka katram x, kam $0<|x-x_0|<\varepsilon$, izpildās nevienādība $f(x) \leqslant f(x_0)$. Tā sasniedz lokālu minimumu, ja $f(x) \geqslant f(x_0)$.*

Ja funkcijai $f(x)$ ir globāls maksimums (minimums) punktā x^* vai lokāls maksimums (minimums) punktā x_0, tad funkcijai

$-f(x)$ ir globāls minimums (maksimums) punktā x^* vai lokāls minimums (maksimums) punktā x_0.

11.3. §. Nelineārās programmēšanas uzdevuma ekonomiskās interpretācijas piemērs. Kā piemēru aplūkosim uzdevuma 1.2.1° variantu, pieņemot, ka sākotnējā informācija ir dota ar tabulu (1.2.1) bez šīs tabulas pēdējās rindiņas. Par optimālo uzskatīsim to ražošanas plānu, kuram atbilst maksimālā peļņas summa. Peļņa kā starpība starp uzņēmuma vairumcenu un produkcijas pilno pašizmaksu dažādu iemeslu dēļ var būt mainīgs lielums.

Pieņemsim, ka aplūkojamā piemērā peļņa no katras produkcijas B_2 vienības ir 30 rbļ. neatkarīgi no saražoto šīs produkcijas vienību skaita, turpretim peļņa no katras produkcijas B_1 vienības ir mainīgs lielums. Par pirmām 3 produkcijas B_1 vienībām peļņa no katras vienības ir 40 rbļ., par nākošām 4 vienībām — 38 rbļ., bet par visām turpmākajām — 35 rbļ. Tādiem nosacījumiem atbilstošā optimālā ražošanas plāna sastādīšanas uzdevuma matemātiskajā modelī mērķa funkcijas definīcijas apgabalu gan var noteikt ar lineāru nevienādību sistēmu (1.2.3), (1.2.4), bet šajā apgabalā mērķa funkcija nav lineāra. Šis piemērs ir ļoti elementārs izliektās programmēšanas uzdevums. To var atvietot ar šādu lineārās programmēšanas uzdevumu.

Atrast maksimumu mērķa funkcijai

$$z=40x_{11}+38x_{12}+35x_{13}+30x_2$$

pie ierobežojumiem

$$\begin{aligned}
x_{11}+x_{12}+x_{13}+7x_2 &\leq 84,\\
2x_{11}+2x_{12}+2x_{13}+5x_2 &\leq 69,\\
5x_{11}+5x_{12}+5x_{13}+5x_2 &\leq 105,\\
4x_{11}+4x_{12}+4x_{13} &\leq 72,\\
x_{11} &\leq 3,\\
x_{12} &\leq 4,
\end{aligned}$$

kur x_{11}, x_{12}, x_{13} ir palīgnezināmie; to summa $x_{11}+x_{12}+x_{13}=x_1$ ir produkcijas B_1 vienību skaits, bet x_2 — produkcijas B_2 vienību skaits.

Vispārinot aplūkoto piemēru, pieņemsim, ka

a_{ij} ir i-tā ražošanas faktora ($i=1, 2, \ldots, m$) patēriņa norma vienas j-tās ($j=1, 2, \ldots, n$) produkcijas vienības ražošanai,

x_j — j-tās produkcijas vienību daudzums,

b_i — i-tā ražošanas resursa krājumu apjoms,

$f_j(x_j)$ — peļņa no vienas j-tā veida produkcijas vienības.

Tad uzņēmuma maksimālās peļņas iegūšanas uzdevums formulējams šādi.

Atrast globālo maksimumu separablai funkcijai

$$f(\boldsymbol{x}) = \sum_{j=1}^{n} x_j f_j(x_j)$$

pie ierobežojumiem

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i \quad (i=1, 2, \ldots, m),$$

$$x_j \geqslant 0 \quad (j=1, 2, \ldots, n).$$

11.4. §. Nelineārās programmēšanas uzdevumu ģeometriskās interpretācijas piemēri. Tāpat kā lineārajā programmēšanā, arī nelineārās programmēšanas uzdevumiem ir uzskatāma ģeometriska interpretācija ortogonālā koordinātu sistēmā plaknē, ja uzdevumam ir tikai divi nezināmie.

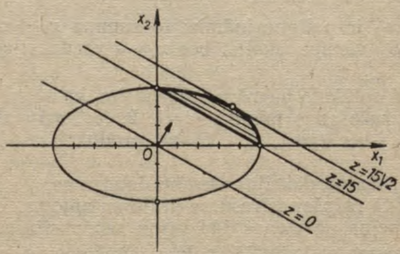
Uzdevums 11.4.1°. Atrast globālo maksimumu un globālo minimumu funkcijai

$$z = 3x_1 + 5x_2$$

pie ierobežojumiem

$$\begin{aligned} &9x_1^2 + 25x_2^2 \leqslant 225, \\ &3x_1 + 5x_2 \geqslant 15, \\ &x_1 \geqslant 0, \ x_2 \geqslant 0. \end{aligned} \tag{11.4.1}$$

Uzdevuma grafiskais atrisinājums attēlots zīmējumā 11.1. Mērķa funkcijas definīcijas apgabals ir izliekta punktu kopa, kuru no vienas puses norobežo uz augšu izliekts elipses loks starp punktiem (0; 3) un (5; 0), bet no otras puses — taisnes nogrieznis starp tiem pašiem punktiem. Mērķa funkcija šajā uzdevumā ir lineāra, un tās līmeņa līnijas ir taišņu saime, kas perpendikulāra pret vektoru (3; 5).



Zīm. 11.1.

Ja (x^*_1, x^*_2) ir dotā uzdevuma plāns, kuram atbilst mērķa funkcijas globālais maksimums z^*, tad no zīmējuma 11.1 izriet, ka jāizpildās vienādībām

$$\left.\begin{aligned} &z^*=3x^*_1+5x^*_2,\\ &9(x^*_1)^2+25(x^*_2)^2=225,\\ &x^*_1=\frac{5}{3}x^*_2. \end{aligned}\right\} \tag{11.4.2}$$

Vienādojumu sistēmas (11.4.2) trešo vienādojumu iegūst, salīdzinot dotās elipses pieskares virziena koeficientu, resp., atvasinājuma $\frac{dx_2}{dx_1}=-\frac{9x_1}{25x_2}$ vērtību punktā (x^*_1, x^*_2), ar mērķa funkcijas līmeņa līniju virziena koeficientu $(-3/5)$. Atrisinot vienādojumu sistēmu (11.4.2), iegūstam $z^*=15\sqrt{2}$, ja $x^*_1=\frac{5\sqrt{2}}{2}$ un $x^*_2=\frac{3\sqrt{2}}{2}$.

Mērķa funkcijas globālais minimums ir jebkurā punktā uz taisnes nogriežņa, kas savieno punktus (0; 3) un (5; 0).

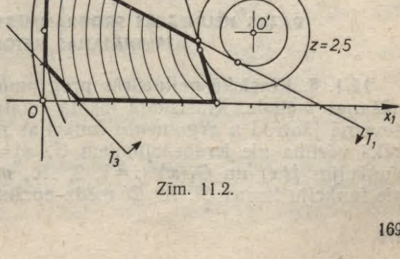
Uzdevums 11.4.2°. Atrast globālo maksimumu un globālo minimumu funkcijai

$$f(\boldsymbol{x})=(x_1-6)^2+(x_2-2)^2$$

pie ierobežojumiem

$$\begin{aligned} x_1+2x_2&\leqslant 8,\\ 3x_1+x_2&\leqslant 15,\\ x_1+x_2&\geqslant 1,\\ x_1, x_2&\geqslant 0. \end{aligned}$$

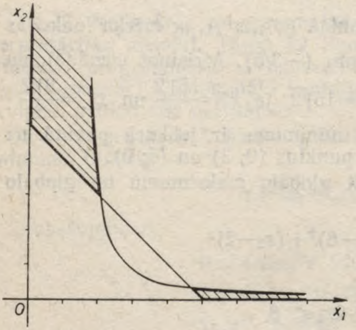
Atšķirībā no iepriekšējā piemēra šajā uzdevumā nelineāra ir mērķa funkcija, kuras līmeņa līnijas ir riņķa līnijas ar centru punktā O' (6; 2). Ierobežojumu sistēmu turpretim veido tikai lineāras funkcijas. Kā redzams no zīmējuma 11.2, vismazākais rādiuss ir tai riņķa līnijai, kuras pieskare ir taisne T_1, bet pieskaršanās punkts nepieder pie dotās mērķa funkcijas definīcijas



Zīm. 11.2.

apgabala. Nākošā, riņķa līniju centram tuvākā ir taisne T_2, kuras virziena koeficients ir -3. Tam jābūt vienādam ar atbilstošās riņķa līnijas pieskares virziena koeficientu pieskaršanās punktā. Ja pieskaršanās punkta koordinātes ir x^*_1 un x^*_2, tad jāizpildās vienādībām

$$\left.\begin{aligned} &f(\boldsymbol{x}^*) = (x^*_1-6)^2+(x^*_2-2)^2, \\ &3x^*_1+x^*_2=15, \\ &-\frac{x^*_1-6}{x^*_2-2}=-3. \end{aligned}\right\} \qquad (11.4.3)$$



Zīm. 11.3.

Atrisinot vienādojumu sistēmu (11.4.3), iegūstam $x^*_1=4{,}5$ un $x^*_2=1{,}5$, kam atbilst mērķa funkcijas globālais minimums $f(\boldsymbol{x}^*)=2{,}5$.

Vistālāk no līmeņa līniju centra atrodas definīcijas apgabala punkts (0; 4), kuram atbilst mērķa funkcijas globālais maksimums $z=40$.

Ja nelineārās programmēšanas ierobežojumu sistēmā ietilpst nelineāras funkcijas, tad plānu kopa var nebūt izliekta. Tā var saturēt pat vairākus atsevišķus apgabalus, kā tas ir, piemēram, zīmējumā 11.3. Tajā attēlots mērķa funkcijas $f(\boldsymbol{x})$ definīcijas apgabals, kuru nosaka ierobežojumi

$$\begin{aligned} &(x_1-2)x_2\leqslant 1, \\ &x_1+x_2\geqslant 5, \\ &x_1, x_2\geqslant 0. \end{aligned}$$

12. nodaļa

DAŽAS NELINEĀRĀS PROGRAMMĒŠANAS UZDEVUMU ATRISINĀŠANAS METODES

12.1. §. Klasiskā nelineārās programmēšanas uzdevumu atrisināšanas metode. Klasiskajā lokālā relatīvā ekstrēma teorijas uzdevumā jāatrod n argumentu funkcijas $f(\boldsymbol{x})$ vislielākā vai vismazākā vērtība pie ierobežojumiem $G_i(\boldsymbol{x})=b_i$ ($i=1, 2, \ldots, m<n$). Funkcijas $f(\boldsymbol{x})$ un $G_i(\boldsymbol{x})$ ($i=1, 2, \ldots, m$) pēc pieņēmuma pieder pie funkciju kopas C'. Ar C' mēdz apzīmēt visu to funkciju kopu,

kuras līdz ar saviem pirmās kārtas parciālajiem atvasinājumiem ir nepārtrauktas funkcijas.

Tādu uzdevumu atrisināšanai var izmantot Lagranža nenoteikto koeficientu metodi. Saskaņā ar šo metodi jāsastāda Lagranža funkcija

$$F(\boldsymbol{x}, \boldsymbol{l}) = f(\boldsymbol{x}) + \sum_{i=1}^{m} l_i [b_i - G_i(\boldsymbol{x})] \qquad (12.1.1)$$

un jāpielīdzina nullei visi tās parciālie atvasinājumi pēc vektoru $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$ un $\boldsymbol{l} = (l_1, l_2, \ldots, l_m)$ koordinātēm:

$$\frac{\partial F(\boldsymbol{x}, \boldsymbol{l})}{\partial x_j} = \frac{\partial f(\boldsymbol{x})}{\partial x_j} - \sum_{i=1}^{m} l_i \frac{\partial G_i(\boldsymbol{x})}{\partial x_j} = 0 \ (j = 1, \ldots, n), \qquad (12.1.2)$$

$$\frac{\partial F(\boldsymbol{x}, \boldsymbol{l})}{\partial l_i} = b_i - G_i(\boldsymbol{x}) = 0 \ (i = 1, 2, \ldots, m). \qquad (12.1.3)$$

Skaitļus $l_1, l_2, \ldots, l_m$ sauc par *Lagranža reizinātājiem*.

Atrisinot vienādojumu sistēmu (12.1.2), (12.1.3), iegūst funkcijas $f(\boldsymbol{x})$ stacionāro punktu koordinātes. Tie ir punkti, kuros funkcijai $f(\boldsymbol{x})$ var būt lokāls maksimums vai minimums. Salīdzinot savā starpā visas funkcijas lokālā maksimuma (minimuma) vērtības, atrod globālo maksimumu (minimumu).

U z d e v u m s 12.1.1°. Atrast lokālo relatīvo maksimumu funkcijai $z = 3x_1 + 5x_2$ pie ierobežojuma $9x_1^2 + 25x_2^2 = 225$.

Lai uzdevumu atrisinātu, sastādām tam atbilstošo Lagranža funkciju

$$F = 3x_1 + 5x_2 + l_1(225 - 9x_1^2 - 25x_2^2)$$

un pielīdzinām nullei visus trīs tās parciālos atvasinājumus:

$$\left.\begin{array}{l} 3 - 18 l_1 x_1 = 0, \\ 5 - 50 l_1 x_2 = 0, \\ 9x_1^2 + 25x_2^2 - 225 = 0. \end{array}\right\} \qquad (12.1.4)$$

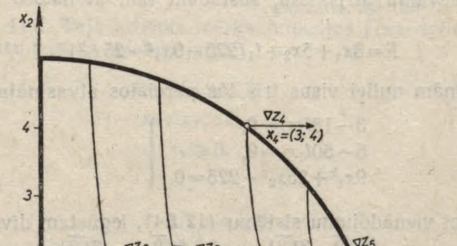
Atrisinot vienādojumu sistēmu (12.1.4), iegūstam divus stacionāros punktus $P_1\left(\frac{5\sqrt{2}}{2}; \frac{3\sqrt{2}}{2}\right)$, $P_2\left(\frac{-5\sqrt{2}}{2}; \frac{-3\sqrt{2}}{2}\right)$, kuriem atbilstošās funkcijas z vērtības ir $z_1 = 15\sqrt{2}$, $z_2 = -15\sqrt{2}$. Lokālais maksimums $z = 15\sqrt{2}$, kuru funkcija z sasniedz punktā ar koordinātēm $x_1 = \frac{5\sqrt{2}}{2}$, $x_2 = \frac{3\sqrt{2}}{2}$, šajā gadījumā ir arī globālais maksimums.

12.2. §. Jēdziens par gradienta metodi. Pieņemsim, ka $z = f(\boldsymbol{x}) \in C'$ ir n argumentu funkcija ar definīcijas apgabalu K. Tad katram $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$ var aprēķināt pirmās kārtas atvasinājuma vērtības $\frac{\partial z}{\partial x_j}$ $(j=1, 2, \ldots, n)$. *Vektoru, kura koordinātes ir funkcijas $z=f(\boldsymbol{x})$ pirmās kārtas parciālie atvasinājumi, sauc par šīs funkcijas gradientu.* To apzīmē ar grad z vai ∇z. No šiem simboliem turpmāk lietosim apzīmējumu

$$\nabla z=\left(\frac{\partial z}{\partial x_1}, \frac{\partial z}{\partial x_2}, \ldots, \frac{\partial z}{\partial x_n}\right). \tag{12.2.1}$$

Funkcijas $z=f(\boldsymbol{x})$ definīcijas apgabalā fiksētam punktam $\boldsymbol{x}_k=(x_{1k}, x_{2k}, \ldots, x_{nk})^{\mathrm{T}}$ atbilstošo šīs funkcijas gradientu turpmāk apzīmēsim ar ∇z_k, tā norādot, ka parciālo atvasinājumu $\frac{\partial z}{\partial x_j}$ $(j=1, 2, \ldots, n)$ skaitliskās vērtības aprēķinātas pie $x_j=x_{jk}$ $(j=1, 2, \ldots, n)$. Ja gradienta ∇z_k koordinātes pierakstāmas kolonnas formā, tad lietosim simbolu ∇z_k^{T}.

Ģeometriskā interpretācijā gradients ir vektors, kas perpendikulārs pret pieskaru hiperplakni funkcijas $z=f(\boldsymbol{x})$ līmeņa virsmai $z_k=f(\boldsymbol{x})$ punktā $\boldsymbol{x}_k$. Divu argumentu funkcijas gradients ir perpendikulārs pret šīs funkcijas līmeņa līnijas pieskari pieskaršanās punktā. Kā piemēru aplūkosim divu argumentu funkciju $z=x_1-\frac{1}{x_2}$, kuras definīcijas apgabals dots ar nevienādībām $x_1^2+x_2^2\leqslant 25$, $x_1\geqslant 0$, $x_2>0$. Šīs funkcijas gradients jebkurā definīcijas apgabalam piederošā punktā $\boldsymbol{x}=(x_1, x_2)^{\mathrm{T}}$ ir $\nabla z=(1, \frac{1}{x_2^2})$. Zīmējumā



Zīm. 12.1.

12.1 attēlots funkcijas definīcijas apgabals un tajā iezīmētas līmeņa līnijas, kas vilktas caur punktiem $\boldsymbol{x}_1=(0;1)^{\mathrm{T}}$, $\boldsymbol{x}_2=(1;2)^{\mathrm{T}}$, $\boldsymbol{x}_3=(2;2)^{\mathrm{T}}$, $\boldsymbol{x}_4=(3;4)^{\mathrm{T}}$, $\boldsymbol{x}_5=(4;2)^{\mathrm{T}}$, $\boldsymbol{x}_6=(4{,}7;\ 1{,}7)^{\mathrm{T}}$.

Iezīmēto līmeņa līniju gradienti minētajos punktos attiecīgi ir $\nabla z_1=(1;1)$, $\nabla z_2=(1;0{,}25)$, $\nabla z_3=(1;0{,}25)$, $\nabla z_4=(1;0{,}0625)$, $\nabla z_5=(1;0{,}25)$, $\nabla z_6=(1;0{,}346)$.

Gradienta vērsums jebkurā mērķa funkcijas definīcijas apgabala punktā norāda šīs funkcijas visstraujākā kāpuma virzienu. Pretējais ir visstraujākā krituma virziens.

Pieņemsim, ka funkcijas $z=f(\boldsymbol{x})$ gradients ∇z_0 kādā brīvi izraudzītā plānu kopas punktā $\boldsymbol{x}_0$ nav nullvektors un ka izliektās programmēšanas uzdevums vispārīgā veidā formulēts šādi: atrast vislielāko vērtību n argumentu funkcijai

$$z=f(\boldsymbol{x}) \tag{12.2.2}$$

pie ierobežojumiem

$$y_i=G_i(\boldsymbol{x})\leqslant 0 \quad (i=1,2,\dots,m). \tag{12.2.3}$$

Lai izveidotu tādu uzdevuma (12.2.2), (12.2.3) plānu virkni $\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \dots$, kuriem atbilst monotoni augoša mērķa funkcijas vērtību virkne $z_0, z_1, z_2, \dots$, pieskaitīsim vektoram $\boldsymbol{x}_0$ tam atbilstošo gradientu ∇z_0^{T}, kas reizināts ar skaitli $t_0>0$. Vektors $\boldsymbol{x}_1=x_0+t_0\nabla z_0^{\mathrm{T}}$ apmierina visus dotā uzdevuma ierobežojumus (12.2.3) tad, ja skaitlis t_0 ir vienādojumu

$$y_{i0}=G_i(\boldsymbol{x}_0+t_0\nabla z_0^{\mathrm{T}})=0 \quad (i=1,2,\dots,m) \tag{12.2.4}$$

vismazākā pozitīvā sakne *.

Ja tāda sakne eksistē un tā ir t^*_0, tad, ievietojot vektora $\boldsymbol{x}$ vietā ierobežojumos (12.2.3) vektoru $\boldsymbol{x}_1=\boldsymbol{x}_0+t^*_0\nabla z_0^{\mathrm{T}}$, jāizpildās nevienādībām

$$-\varepsilon_1\leqslant G_i(\boldsymbol{x}_1)\leqslant 0 \quad (i\in I_1), \tag{12.2.5}$$

kur parametrs $\varepsilon_1>0$ ir pietiekami mazs pozitīvs skaitlis. Ar I_1 šeit apzīmēta to indeksu i kopa, kuriem izpildās nevienādība (12.2.5). Tā kā gradienta virzienā funkcijas z vērtība aug, tad vektoram $\boldsymbol{x}_1$ atbilstošā funkcijas vērtība $z_1>z_0$.

Ja vienādojumiem (12.2.4) nav pozitīvu sakņu, tad punkts $\boldsymbol{x}_0$ atrodas uz plānu kopas robežas tādā vietā, ka mērķa funkcijas pieaugums tās gradienta virzienā no šī punkta nav iespējams. Ejot gradienta virzienā, mēs nonāktu ārpus plānu kopas, t. i., nonāktu tādā punktā, kura koordinātes neapmierina kādu no ierobežojumiem (12.2.3). Ja vienādojumiem (12.2.4) nav pozitīvu sakņu, tad $\boldsymbol{x}_0=\boldsymbol{x}_1$ un eksistē vismaz viena no nevienādībām (12.2.5). Tādēļ varam pieņemt, ka virknē $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_k$ visi punkti atrodas uz

* Šie vienādojumi jāatrisina ar kādu no pazīstamām vienādojumu aptuvenās atrisināšanas metodēm.

plānu kopas robežas un mērķa funkcijas pieaugums tās gradienta virzienā no šiem punktiem nav iespējams.

Pieņemsim, ka esam atraduši punktu $\boldsymbol{x}_k$, kas apmierina nevienādības

$$-\varepsilon_k \leqslant y_{ik} \leqslant 0 \quad (i \in I_k), \tag{12.2.6}$$

kur $y_{ik} = G_i(\boldsymbol{x}_k)$, $\varepsilon_k > 0$ — pietiekami mazs skaitlis un I_k — indeksu i kopa, kuriem izpildās šīs nevienādības.

Lai atrastu tādu punktu $\boldsymbol{x}_{k+1}$, kurā izpildās prasība $z_{k+1} \geqslant z_k$, jānosaka *virziena vektors* $\boldsymbol{r}_k$ un *soļa koeficients* t_k, ar kuriem

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + t_k \boldsymbol{r}_k. \tag{12.2.7}$$

Vektors $\boldsymbol{r}_k = (r_{1k}, r_{2k}, \ldots, r_{nk})^{\mathrm{T}}$ nosakāms, vadoties no prasības, lai tā virzienā pieaugtu mērķa funkcija $z = f(\boldsymbol{x})$ un lai šis virziens vestu uz plānu kopas iekšējiem punktiem. Šo prasību ģeometriskā interpretācija ir tāda, ka leņķiem starp vektoru $\boldsymbol{r}_k$ un to funkciju gradientiem, kuras apmierina nevienādības (12.2.6), jābūt platiem. Platam jābūt arī leņķim starp vektoru $\boldsymbol{r}_k$ un $-\nabla z_k$. Tas nozīmē, ka vektoru ∇y_{ik} $(i \in I_k)$ skalārajiem reizinājumiem ar vektoru $\boldsymbol{r}_k$ jābūt negatīviem. Negatīvam jābūt arī skalārajam reizinājumam $-\nabla z_k^{\mathrm{T}} \cdot \boldsymbol{r}_k$. Bez tam vektora $\boldsymbol{r}_k$ modulim jābūt ierobežotam. Šīs prasības apmierina šāda lineārās programmēšanas uzdevuma atrisinājums.

Maksimizēt lineāru funkciju

$$v = \eta_k \tag{12.2.8}$$

pie ierobežojumiem

$$\left.\begin{aligned} -\nabla z_k^{\mathrm{T}} \cdot \boldsymbol{r}_k + \eta_k &\leqslant 0, \\ \nabla y_{ik}^{\mathrm{T}} \cdot \boldsymbol{r}_k + \eta_k &\leqslant 0 \quad (i \in I_k), \\ |r_{jk}| &\leqslant 1 \quad (j = 1, 2, \ldots, n). \end{aligned}\right\} \tag{12.2.9}$$

Kad uzdevums (12.2.8), (12.2.9) ir atrisināts un ir atrasts virziena vektors $\boldsymbol{r}_k$ ar tam atbilstošo $\max v = \eta^*_k$, jānosaka formulā (12.2.7) vajadzīgais soļa koeficients t_k. Šis skaitlis ir vienādojumu

$$G_i(\boldsymbol{x}_k + t_k \boldsymbol{r}_k) = 0 \quad (i = 1, 2, \ldots, m) \tag{12.2.10}$$

vismazākā pozitīvā sakne.

Ja tāda sakne eksistē un tā ir t^*_k, tad, ievietojot vektora $\boldsymbol{x}$ vietā ierobežojumos (12.2.3) vektoru $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + t^*_k \boldsymbol{r}_k$, jāizpildās nevienādībām

$$-\varepsilon_{k+1} \leqslant y_{i,k+1} \leqslant 0 \quad (i \in I_{k+1}), \tag{12.2.11}$$

kur $y_{i,k+1} = G_i(\boldsymbol{x}_{k+1})$; $\varepsilon_{k+1} > 0$ ir parametrs, kas nosakāms atkarībā no tā, kāda ir diference $\varepsilon_k - \eta^*_k$. Ar I_{k+1} apzīmēta indeksu i kopa, kuriem izpildās nevienādības (12.2.11).

Parametra $\varepsilon_{k+1}>0$ noteikšanai nevienādībās (12.2.11) izšķirsim 2 gadījumus: a) $\varepsilon_k-\eta^*_k<0$; b) $\varepsilon_k-\eta^*_k\geqslant 0$.

Gadījumā a ņemam $\varepsilon_{k+1}=\varepsilon_k$, bet gadījumā b — $\varepsilon_{k+1}=0{,}5\varepsilon_k$.

Ja $\max v=\eta^*_k=\eta^*_{k+1}=0$, tad $\boldsymbol{x}_{k+1}$ ir dotā uzdevuma optimālais plāns un $\max z=f(\boldsymbol{x}_{k+1})$.

Funkcijai $z=f(\boldsymbol{x})$ ir maksimums z^* punktā $\boldsymbol{x}^*$ tad un tikai tad, ja ∇z^* ir kolineārs ar to hipervirsmu $y^*_i=0$ gradientu ∇y^*_i summu, uz kurām atrodas punkts $\boldsymbol{x}^*$.

Rezumējot izklāstu par gradienta metodi, formulēsim vienu no iespējamiem algoritmiem izliektās programmēšanas uzdevumu (12.2.2), (12.2.3) atrisināšanai ar šo metodi.

Gradienta metodes algoritms sastāv no *sākotnējā soļa* un tam sekojošiem *iterāciju soļiem*. Sākotnējā solī pilnīgi brīvi jāizraugās kāds plānu kopas punkts $\boldsymbol{x}_0$, jānosaka parametrs $\varepsilon_1>0$, jāatrod ∇z_0, jāatrisina vienādojumi $G_i(\boldsymbol{x}_0+t_0\nabla z_0)=0$ $(i=1,2,\ldots,m)$ un jāatrod šo vienādojumu mazākā pozitīvā sakne t^*_0. Tad $\boldsymbol{x}_1=\boldsymbol{x}_0+t^*_0\nabla z_0$ ir plānu kopas robežpunkts, kas apmierina nevienādības (12.2.6), ja $k=1$. Ja t^*_0 neeksistē, tad $\boldsymbol{x}_1=\boldsymbol{x}_0$ un tālāk seko iterāciju soļi.

1. solis. Aprēķināt atrastam plānu kopas robežpunktam $\boldsymbol{x}_k$ atbilstošos z_k un y_{ik}.

2. solis. Noteikt pietiekami mazu parametru $\varepsilon_k>0$ un atrast šim ε_k atbilstošo indeksu i kopu I_k, kurā izpildās nevienādības (12.2.6).

3. solis. Aprēķināt ∇z_k un ∇y_{ik} $(i\in I_k)$.

4. solis. Sastādīt un atrisināt lineārās programmēšanas uzdevumu (12.2.8), (12.2.9), t. i., atrast $\max v=\eta^*_k$ un mērķa funkcijas pieauguma virziena vektoru $\boldsymbol{r}_k$.

5. solis. Atrast vienādojumu (12.2.10) mazāko pozitīvo sakni t^*_k.

6. solis. Atrast $\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+t^*_k\boldsymbol{r}_k$ ar šim punktam atbilstošiem z_{k+1} un $y_{i,k+1}$ $(i=1,2,\ldots,m)$.

7. solis. Noteikt jaunu parametru $\varepsilon_{k+1}=\varepsilon_k$, ja $\varepsilon_k-\eta^*_k<0$, vai $\varepsilon_{k+1}=0{,}5\varepsilon_k$, ja $\varepsilon_k-\eta^*_k\geqslant 0$.

Šie 7 soļi veido vienu iterāciju. Atvietojot katrā solī indeksu k ar $k+1$, iterācijas jāatkārto tik reižu, cik nepieciešams, lai iegūtu optimālo plānu ar vajadzīgo precizitāti. To raksturo parametrs ε_k un attiecīgo gradientu kolinearitāte noapaļojumu kļūdu robežās.

Piemērs 12.2.1°. Maksimizēt funkciju

$$z=x_1-\frac{1}{x_2} \tag{12.2.12}$$

pie ierobežojumiem

$$\left.\begin{aligned} y_1&\equiv x_1^2+x_2^2-25\leqslant 0,\\ y_2&\equiv -x_1\leqslant 0,\\ y_3&\equiv -x_2\leqslant 0.\end{aligned}\right\} \tag{12.2.13}$$

A t r i s i n ā j u m s. Pieņemsim par sākuma punktu $\boldsymbol{x}_0=(4;2)^T$ un par sākuma parametru $\varepsilon_1=0{,}1$. Tad visi y_i ir mazāki par ε_1, tāpēc $\boldsymbol{x}_0$ ir plānu kopas iekšējs punkts (sk. zīm. 12.1) un funkcija z visstraujāko kāpumu, neizejot ārpus plānu kopas, iegūst tās gradientu virzienā. ∇z un ∇y_i $(i=1, 2, 3)$ pie jebkura $\boldsymbol{x}$ ir

$$\left.\begin{aligned} \nabla z&=\left(1;\ \frac{1}{x_2^2}\right),\\ \nabla y_1&=(2x_1;\ 2x_2),\\ \nabla y_2&=(-1;\ 0),\\ \nabla y_3&=(0;\ -1).\end{aligned}\right\} \tag{12.2.14}$$

Saskaņā ar izteiksmēm (12.2.14) punktam $\boldsymbol{x}_0=(4;2)^T$ atbilstošas gradientu vērtības ir $\nabla z_0=(1;0{,}25)$; $\nabla y_{10}=(8;4)$; ∇y_2 un ∇y_3 pie jebkura $\boldsymbol{x}$ ir nemainīgi vektori. Punktam $\boldsymbol{x}_0$ atbilstošā mērķa funkcijas vērtība ir $z_0=4-0{,}5=3{,}5$. Tās pieaugumu iegūsim, ievietojot $\boldsymbol{x}_0$ vietā $\boldsymbol{x}_1=(4+t_0;\ 2+0{,}25t_0)^T$. Soļa koeficients $t^*_0=0{,}523$ ir vienādojuma $(4+t_0)^2+(2+0{,}25t_0)^2-25=0$ mazākā pozitīvā sakne. Tādēļ $\boldsymbol{x}_1=(4{,}52;\ 2{,}13)^T$; $z_1=4{,}05$; $y_{11}=-0{,}03$, resp., $-0{,}1<-0{,}03<0$. Tātad punkts $\boldsymbol{x}_1$ atrodas uz plānu kopas robežas. Ar to sākotnējais solis ir noslēdzies, un jāizpilda iterāciju soļi.

1. i t e r ā c i j a. *1. solis:* $\boldsymbol{x}_1=(4{,}52;2{,}13)^T$; $z_1=4{,}05$; $y_{11}=-0{,}03$; $y_{21}=-4{,}52$; $y_{31}=-2{,}53$.

2. solis: $\varepsilon_1=0{,}1$; $I_1=\{1\}$.

3. solis: $\nabla z_1=(1;\ 0{,}220)$; $\nabla y_{11}=(9{,}04;\ 4{,}26)$.

4. solis: Maksimizēt $v=\eta_1$ pie ierobežojumiem $-r_{11}-0{,}220r_{21}+\eta_1\leqslant 0$; $9{,}04r_{11}+4{,}26r_{21}+\eta_1\leqslant 0$; $r_{11}\leqslant 1$; $-r_{11}\leqslant 1$; $r_{21}\leqslant 1$; $-r_{21}\leqslant 1$. Atrisinot šo 4. soļa lineārās programmēšanas uzdevumu, atrod $\eta^*_1=0{,}23$; $r_{11}=0{,}45$, $r_{21}=-1$, t. i., $\boldsymbol{r}_1=(0{,}45;\ -1)^T$.

5. solis: soļa koeficients $t^*_1=0{,}19$ ir vienādojuma $(4{,}52+0{,}45t_1)^2+(2{,}13-t)^2-25=0$ mazākā pozitīvā sakne.

6. solis: $\boldsymbol{x}_2=(4{,}60;\ 1{,}94)^T$; $z_2=4{,}09$; $y_{12}=-0{,}08$; $y_{22}=-4{,}60$; $y_{32}=-1{,}94$.

7. solis: $\varepsilon_2=0{,}1$, jo $\varepsilon_1=0{,}1$, $\eta^*_1=0{,}23$ un $0{,}1-0{,}23=-0{,}13<0$.

2. i t e r ā c i j a. 1) $\boldsymbol{x}_2=(4{,}60;\ 1{,}94)^T$; $z_2=4{,}09$; $y_{12}=-0{,}08$. 2) $\varepsilon_2=0{,}1$; $I_2=\{1\}$. 3) $\nabla z_2=(1;\ 0{,}266)$; $\nabla y_{12}=(9{,}20;\ 3{,}88)$. 4) Maksimizēt $v=\eta_2$ pie ierobežojumiem $-r_{12}-0{,}266r_{22}+\eta_2\leqslant 0$; $9{,}20r_{12}+3{,}88r_{22}+\eta_2\leqslant 0$; $r_{12}\leqslant 1$; $-r_{12}\leqslant 1$; $r_{22}\leqslant 1$; $-r_{22}\leqslant 1$. Atrisinot šo 4. soļa lineārās programmēšanas uzdevumu, atrod $\eta^*_2=0{,}14$; $\boldsymbol{r}_2=(0{,}41;\ -1)^T$. 5) $t^*_2=0{,}15$ ir vienādojuma $(4{,}60+0{,}41t_2)^2+(1{,}94-t_2)^2-25=0$ mazākā pozitīvā sakne. 6) $\boldsymbol{x}_3=(4{,}66;\ 1{,}79)^T$; $z_3=4{,}10$; $y_{13}=-0{,}08$. 7) $\varepsilon_3=0{,}1$, jo $\varepsilon_2=0{,}1$; $\eta^*_2=0{,}14$ un $0{,}1-0{,}14=-0{,}04<0$.

3. i t e r ā c i j a. 1) $\boldsymbol{x}_3=(4{,}66;\ 1{,}79)^T$; $z_3=4{,}10$; $y_{12}=-0{,}08$. 2) $\varepsilon_3=0{,}1$; $I_3=\{1\}$. 3) $\nabla z_3=(1;\ 0{,}312)$; $\nabla y_{13}=(9{,}32;\ 3{,}58)$. 4) $\eta^*_3=0{,}07$; $\boldsymbol{r}_3=(0{,}39;\ -1)^T$. 5) $t^*_3=0{,}12$. 6) $\boldsymbol{x}_4=(4{,}71;\ 1{,}67)^T$; $z_4=4{,}111$; $y_{14}=-0{,}03$. 7) $\varepsilon_4=0{,}5\varepsilon_3=0{,}05$, jo $\varepsilon_3-\eta^*_3=0{,}03>0$.

4. i t e r ā c i j a. 1) $\boldsymbol{x}_4=(4{,}71; 1{,}67)^{\mathrm{T}}$; $z_4=4{,}11$; $y_{14}=-0{,}03$. 2) $\varepsilon_4=0{,}05$; $I_4=\{1\}$. 3) $\nabla z_4=(1; 0{,}358)$; $\nabla y_{14}=(9{,}42; 3{,}34)$. 4) $\eta^*{}_4=0{,}02$; $\boldsymbol{r}_4=(-0{,}34; 1)^{\mathrm{T}}$. 5) $t^*{}_4=0{,}03$. 6) $\boldsymbol{x}_5=(4{,}70; 1{,}70)^{\mathrm{T}}$; $z_5=4{,}112$; $y_{15}=-0{,}02$. 7) $\varepsilon_5=0{,}025$.

5. i t e r ā c i j a. 1) $\boldsymbol{x}_5=(4{,}70; 1{,}70)^{\mathrm{T}}$; $z_5=4{,}11$; $y_{15}=0{,}02$. 2) $\varepsilon_5=0{,}025$; $I_5=\{1\}$. 3) $\nabla z_5=(1; 0{,}346)$; $\nabla y_{15}=(9{,}40; 3{,}40)$. 4) $\eta^*{}_5=0{,}014$; $\boldsymbol{r}_5=(0{,}36; -1)^{\mathrm{T}}$. 5) $t^*{}_5=0{,}05$. 6) $\boldsymbol{x}_6=(4{,}72; 1{,}65)^{\mathrm{T}}$; $z_6=4{,}114$; $y_{16}=0{,}00$. 7) $\varepsilon_6=0{,}012$.

6. i t e r ā c i j a. 1) $\boldsymbol{x}_6=(4{,}72; 1{,}65)^{\mathrm{T}}$; $z_6=4{,}114$; $y_{16}=0{,}00$. 2) $\varepsilon_6=0{,}012$; $I_6=\{1\}$. 3) $\nabla z_6=(1; 0{,}367)$; $\nabla y_{16}=(9{,}44; 3{,}30)$. 4) $\eta^*{}_6=0{,}016$; $\boldsymbol{r}_6=(-0{,}351; 1)^{\mathrm{T}}$. 5) Nav tāda pozitīva $t^*{}_6$, kas apmierinātu vienādojumus $y_{i6}=0$ $(i=1, 2, 3)$. Tādēļ $\boldsymbol{x}_6=\boldsymbol{x}_5=(4{,}72; 1{,}65)^{\mathrm{T}}$ un $z_6=z_5=4{,}114$.

A t b i l d e. Dotā uzdevuma (12.2.12), (12.2.13) optimālais atrisinājums ir $z^*=4{,}114$, ja $\boldsymbol{x}^*=(4{,}72; 1{,}65)^{\mathrm{T}}$. Noapaļojumu kļūda nepārsniedz $\varepsilon=0{,}013$.

12.3. §. Gomori metode diskrētajā programmēšanā. Praksē bieži jāsastopas ar tādiem matemātiskās programmēšanas uzdevumiem, kuros vismaz dažu nezināmo lielumu vērtības var būt tikai veseli skaitļi. Šādus diskrētās programmēšanas uzdevumus, kad mērķa funkcija un ierobežojumi ir lineāri, mēdz atrisināt parastajā kārtībā, nepievēršot uzmanību diskrētības nosacījumiem. Pēc optimālā plāna atrašanas nezināmos lielumus, kuriem jāizpilda diskrētības nosacījumi, vienkārši noapaļo līdz tuvākajiem veselajiem skaitļiem. Tāda prakse, ja nezināmo vērtības ir relatīvi lieli skaitļi, parasti attaisnojas. Tomēr nereti tādā veidā var iegūt aplamus un pilnīgi nepieņemamus rezultātus, īpaši tad, ja nezināmie var pieņemt tikai divas vērtības: 0 vai 1.

Aplūkosim lineāra diskrētās programmēšanas uzdevuma risināšanas algoritmu, ko sauc par Gomori algoritmu (metodi). To var izmantot tādu uzdevumu risināšanai, kuri vispārīgā veidā formulējami šādi:

$$\left.\begin{aligned} z=\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}\rightarrow\max,\\ A\boldsymbol{x}=\boldsymbol{b},\ \boldsymbol{x}\geqslant\boldsymbol{o}, \end{aligned}\right\} \tag{12.3.1}$$

$$x_j \text{ — veseli skaitļi } (j\in I_1). \tag{12.3.2}$$

Te lietoto simbolu nozīme ir tāda pati kā 9. nodaļā. I_1 ir to indeksu j kopa, kuriem atbilstošajām vektora $\boldsymbol{x}$ koordinātēm x_j jābūt veseliem skaitļiem.

Tā kā lineāru vienādojumu sistēmu $A\boldsymbol{x}=\boldsymbol{b}$ vienmēr var pārveidot tā, lai visi matricas A elementi būtu veseli skaitļi, tad uzskatīsim, ka šis nosacījums ir izpildīts jau uzdevuma (12.3.1), (12.3.2) formulējumā. Saskaņā ar Gomori algoritmu vispirms risina uzdevumu (12.3.1), ignorējot diskrētības nosacījumus (12.3.2). Šādu uzdevuma (12.3.1), (12.3.2) apakšuzdevumu turpmāk apzīmēsim ar G_0. Ar apzīmējumiem (9.1.12), (9.1.13) uzdevuma

G_0 optimālais atrisinājums ir $z^*_0 = \boldsymbol{c}^{B}\boldsymbol{B}^{-1}\boldsymbol{b}$, ja bāzes mainīgie $\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b}$ un nebāzes mainīgie $\boldsymbol{x}_D = \boldsymbol{o}$. Ja vektora $\boldsymbol{x}_B$ koordinātes x_j $(j \in I_1)$ ir veseli skaitļi, tad uzdevums (12.3.1), (12.3.2), kuru turpmāk apzīmēsim ar G, ir atrisināts. Ja turpretim kāda no vektora $\boldsymbol{x}_B$ koordinātēm x_j $(j \in I_1)$ ir daļskaitlis, tad jārisina uzdevums G_1. Pirms šī uzdevuma formulēšanas jādefinē daži palīgjēdzieni un jāpieņem tiem apzīmējumi.

Definīcija 1°. *Par reāla skaitļa a veselo daļu sauc lielāko veselo skaitli $[a]$, kas apmierina nevienādību $[a] \leqslant a$.*

Definīcija 2°. *Par reāla skaitļa a daļveida daļu sauc skaitli $a - [a]$.*

Piemēri: $[6]=6$; $[6{,}2]=6$; $\left[\frac{5}{3}\right]=1$;

$[-6]=-6$; $[-6{,}2]=-7$; $\left[-\frac{5}{3}\right]=-2$.

Šo skaitļu daļveida daļas ir attiecīgi 0; 0,2; 2/3; 0; 0,8; 1/3.

Saskaņā ar definīcijām 1° un 2° vienmēr ir spēkā nevienādība $0 \leqslant t < 1$, kur $t = a - [a]$ ir reāla skaitļa a daļveida daļa.

Definīcija 3°. *Par nevienādībai $a_1x_1 + a_2x_2 + \ldots + a_nx_n \leqslant b$ atbilstošo Gomori šķēlumu sauc nevienādību $t_1x_1 + t_2x_2 + \ldots + t_nx_n \geqslant t$, kur $t_i = a_i - [a_i]$ $(i = 1, 2, \ldots, n)$ un $t = b - [b]$ ir skaitļu a_i un b daļveida daļas.*

Piemērs. Nevienādībai $\frac{4}{3}x_1 - 5x_2 + \frac{2}{5}x_3 - \frac{7}{4}x_4 \leqslant \frac{5}{2}$ atbilstošais Gomori šķēlums ir $\frac{1}{3}x_1 + 0x_2 + \frac{2}{5}x_3 + \frac{1}{4}x_4 \geqslant \frac{1}{2}$, resp., vienādojumam $x_5 - \frac{5}{2} + \frac{4}{3}x_1 - 5x_2 + \frac{2}{5}x_3 - \frac{7}{4}x_4 = 0$ atbilstošais Gomori šķēlums ir $-g - \frac{1}{2} + \frac{1}{3}x_1 + 0x_2 + \frac{2}{5}x_3 + \frac{1}{4}x_4 = 0$, kur $x_5 \geqslant 0$ un $g \geqslant 0$.

Tagad varam atgriezties pie uzdevuma G_1 formulēšanas. Uzdevumu G_1 iegūst, ja uzdevuma G_0 pēdējo iterācijas tabulu papildina ar vienu jaunu ierobežojumu. Jaunais ierobežojums jāformulē kā uzdevuma G_0 pēdējā iterācijas tabulā nolasāmajam vienādojumam atbilstošais Gomori šķēlums. Šķēluma sastādīšanai eksistē dažādi priekšraksti. Vienkāršākais no tiem ir sastādīt Gomori šķēlumu, kas atbilst tam ierobežojumu sistēmas vienādojumam, kuram pēdējā iterācijas tabulā ir lielākais brīvais loceklis. Pagarinot ar šo šķēlumu kā ar papildu ierobežojumu pēdējo iterācijas tabulu, iegūst uzdevuma G_1 pirmo iterācijas tabulu. Ja arī uzdevuma G_1 optimālais atrisinājums neapmierina visus uzdevuma G nosacījumus, tad pilnīgi analogi jāsastāda uzdevums G_2 utt. Šādu uzdevumu sastādīšana un atrisināšana jāatkārto tik reižu, cik nepieciešams, lai iegūtu uzdevuma G atrisinājumu vai arī lai pārliecinātos, ka atrisinājums neeksistē. Uzdevumam G nav atrisinājuma, ja neeksistē atrisinājums kādam no uzdevumiem G_0, G_1,

$G_2, \ldots$ Jāievēro, ka vienādojumam $\boldsymbol{a}^T\boldsymbol{x}=b$ nav atrisinājuma, ja b nav vesels skaitlis, bet visas vektora $\boldsymbol{a}^T$ koordinātes ir veseli skaitļi un visām vektora $\boldsymbol{x}$ koordinātēm arī jābūt veseliem skaitļiem.

Piemērs 12.3.1°. Maksimizēt $z=3x_1+3x_2+13x_3$ pie ierobežojumiem $-3x_1+6x_2+7x_3\leqslant 8$; $6x_1-3x_2+7x_3\leqslant 8$; $x_j\geqslant 0$ $(j=1, 2, 3)$, visiem x_j jābūt veseliem skaitļiem.

Atrisinājums. Apzīmējot doto uzdevumu ar G, risināsim tam atbilstošo lineārās programmēšanas uzdevumu G_0, kura pirmā simpleksa metodes iterācijas tabula ir šāda:

$+1$	-1	x_1	x_2	x_3	$(\mathrm{I}G_0)$
z	0	-3	-3	-13	0
x_4	8	-3	6	$\boxed{7}$	1
x_5	8	6	-3	7	2
σ	15	-1	-1	0	3

(12.3.3)

Ar $x_4\geqslant 0$ un $x_5\geqslant 0$ tabulā (12.3.3) apzīmēti papildu nezināmie, kas pēc uzdevuma G nosacījumiem var arī nebūt veseli skaitļi. Izpildot trīs iterācijas, iegūstam uzdevuma G_0 pēdējo iterācijas tabulu:

$+1$	-1	x_5	x_3	x_4	$(\mathrm{IV}G_0)$
z	16	1	1	1	0
x_2	8/3	1/9	7/3	2/9	1
x_1	8/3	2/9	7/3	1/9	2
σ	61/3	1/3	14/3	1/3	3

(12.3.4)

Tā kā uzdevuma G_0 optimālajā atrisinājumā $x_1=x_2=8/3$, kas nav veseli skaitļi, tad sastādīsim tabulas (12.3.4) 1. rindai atbilstošo Gomori šķēlumu un jaunas tabulas veidā formulēsim uzdevumu G_1. Šis uzdevums, tāpat kā uzdevums G_0, ir maksimizācijas uzdevums, kurš nolasāms tabulā (12.3.5):

$+1$	-1	x_5	x_3	x_4	$(\mathrm{I}G_1)$
z	16	1	1	1	0
x_2	8/3	1/9	7/3	2/9	1
x_1	8/3	2/9	7/3	1/9	2
g_1	$-2/3$	$-2/9$	$\boxed{-1/3}$	$-1/9$	3
σ	59/3	1/9	13/3	2/9	4

(12.3.5)

Tabulā (12.3.5) $g_1 \geqslant 0$ ir papildu nezināmais Gomori šķēlumā, ar kuru pagarināta uzdevuma G_0 ierobežojumu sistēma.

Ja lieto duālo simpleksa metodi, tabulai (12.3.5) seko tabulas (12.3.6), (12.3.7):

$+1$	-1	x_5	g_1	x_4	$(\mathrm{II}G_1)$
z	14	1/3	3	2/3	0
x_2	−2	−13/9	7	−5/9	1
x_1	−2	$\boxed{-4/3}$	7	−2/3	2
x_3	2	2/3	−3	1/3	3
σ	11	−25/9	13	−11/9	4

(12.3.6)

$+1$	-1	x_1	g_1	x_4	$(\mathrm{III}G_1)$
z	27/2	1/4	19/4	1/2	0
x_2	1/6	−13/12	−7/12	1/6	1
x_5	3/2	−3/4	−21/4	1/2	2
x_3	1	1/2	1/2	0	3
σ	91/6	−25/12	−19/12	1/6	4

(12.3.7)

Tabulā (12.3.7) nolasāmais uzdevuma G_1 optimālais atrisinājums arī neapmierina visus uzdevuma G ierobežojumus, jo $x_2 = 1/6$ ir daļskaitlis. Tādēļ jāsastāda un jāatrisina uzdevums G_2. Tabulas (12.3.7) brīvo locekļu kolonnā lielākais skaitlis ir 3/2, kas atrodas 2. rindā. Tādēļ jāsastāda šajā rindā nolasāmajam vienādojumam atbilstošais Gomori šķēlums. Pievienojot šo Gomori šķēlumu tabulai (12.3.7), iegūst uzdevuma G_2 formulējumu, kas nolasāms kā funkcijas z maksimizācijas uzdevums tabulā (12.3.8):

$+1$	-1	x_1	g_1	x_4	$(\mathrm{I}G_2)$
z	27/2	1/4	19/4	1/2	0
x_2	1/6	−13/12	−7/12	1/6	1
x_5	3/2	−3/4	−21/4	1/2	2
x_3	1	1/2	1/2	0	3
g_2	−1/2	−1/4	−3/4	$\boxed{-1/2}$	4
σ	44/3	−7/3	−7/3	−1/3	5

(12.3.8)

Ja lieto duālo simpleksa metodi, tabulai (12.3.8) seko uzdevuma G_2 optimālā atrisinājuma tabula (12.3.9):

$+1$	-1	x_1	g_1	g_2	(IIG_2)
z	13	0	4	1	0
x_2	0	$-7/6$	$-5/6$	$1/3$	1
x_5	1	-1	-6	1	2
x_3	1	$1/2$	$1/2$	0	3
x_4	1	$1/2$	$3/2$	-2	4
σ	15	$-13/6$	$-11/6$	$-2/3$	5

(12.3.9)

Uzdevuma G_2 optimālais plāns apmierina visus dotā uzdevuma G nosacījumus. Tādēļ tā atrisinājums ir arī uzdevuma G optimālais atrisinājums: $\max z=13$, ja $x_1=0$, $x_2=0$ un $x_3=1$.

Šī paragrāfa sākumā tika izteiktas kritiskas piezīmes par paņēmienu, kad vienkārši noapaļo uzdevuma G_0 optimālā atrisinājuma rezultātus, lai iegūtu veselos skaitļos izteiktu plānu. Cik aplama var būt tāda prakse, rāda tikko atrisinātā konkrētā uzdevuma rezultāti.

Ja noapaļotu tabulā (12.3.4) nolasāmā uzdevuma G_0 optimālā plāna skaitļus, tad iegūtu $x_1=3$, $x_2=3$; $x_3=0$, kam atbilstošā z vērtība ir 18. Tāds paņēmiens ir nederīgs tā vienkāršā iemesla dēļ, ka šie veselie plāna skaitļi neapmierina dotā uzdevuma ierobežojumus. Bez tam šie skaitļi rada maldīgu iespaidu, it kā dotās mērķa funkcijas vērtība varētu sasniegt skaitli 18. Noapaļojot uz leju, t. i., pieņemot $x_1=x_2=2$, rodas maldīgs priekšstats, ka maksimālā z vērtība ir tikai 12.

Noslēdzot šo paragrāfu, jāatzīmē, ka gadījumā, ja kādā no aprakstītā algoritma uzdevumiem G_k $(k=2, 3, 4, \ldots)$ pēdējā iterācijas tabulā starp bāzes mainīgajiem parādās skaitļi g_i, kuru indekss i mazāks par k, tad tās tabulas rindas, kurās šādi g_i atrodas, var vienkārši izsvītrot. Tādēļ nevienā no uzdevumiem G_k ierobežojumu rindu skaits nevar pārsniegt $n+1$, ja vektoram $\boldsymbol{x}$ ir n koordinātes.

12.4. §. Jēdziens par dinamiskās programmēšanas metodi. Ar nosaukumu dinamiskā programmēšana mēdz apzīmēt īpašu matemātiskās programmēšanas uzdevumu skaitliskās atrisināšanas metodi. Tādu nosaukumu metodei devis tās autors R. Bellmans un viņa līdzgaitnieki. Dinamiskās programmēšanas rašanās saistīta ar dažu daudzsoļu vadības procesu optimizācijas pētījumiem, tādēļ arī ieviesies no angļu valodas pārņemtais metodes nosaukums. Dinamiskā programmēšana kā matemātiskās programmēšanas uzdevumu skaitliskās atrisināšanas metode principiāli atšķiras no visām iepriekš minētajām metodēm, un tai ir universāls raksturs. Teorētiski ar šo metodi var aizstāt visas tās matemātiskās pogrammēšanas uzdevumu atrisināšanas metodes, kuras aplūkotas šīs grāmatas iepriekšējos paragrāfos un nodaļās, bez tam dinamiskās programmēšanas metode ir noderīga arī tādu uzdevumu risināšanai, kuros nav izpildīti iepriekšējos paragrāfos un nodaļās aplūkoto skaitlisko metožu pielietošanas iespēju nosacījumi. Vienu no tādiem uzdevumiem vispārīgā veidā var formulēt šādi.

Uzdevums 12.4.1°. Maksimizēt funkciju $z=f_1(x_1)+f_2(x_2)+ +\ldots+f_n(x_n)$ pie ierobežojumiem $a_1x_1+a_2x_2+\ldots+a_nx_n\leqslant b$, visi $x_j\geqslant 0$ $(j=1, 2, \ldots, n)$ ir veseli skaitļi, $a_j>0$.

Šādam uzdevumam ir vienkārša ekonomiskā interpretācija. Pieņemsim, ka mūsu rīcība ir atkarīga no kāda ražošanas resursa (zemes platības, darbaspēka, naudas līdzekļiem vai kāda cita resursa). Šo resursu var izmantot ar n dažādiem ražošanas paņēmieniem, kurus sauksim par tehnoloģijām. Resursa izmantošana ar kādu no tehnoloģijām dod zināmu ienākumu, kuru var novērtēt ar paša šī resursa vienībām vai arī ar pavisam citādām mērvienībām, pārvēršot, piemēram, visu ienākumu naudas izteiksmē. Ienākums ir atkarīgs kā no izmantotā resursa daudzuma, tā arī no pielietotās tehnoloģijas. Optimālās plānošanas uzdevums ir izmantot doto resursu tā, lai gūtu maksimālo ienākumu. Dinamiskās programmēšanas metodē pieņem, ka a) ienākumu $f_j(x_j)$, kuru iegūst no resursa daudzuma x_j izmantošanas pēc j-tās tehnoloģijas, var izmērīt ar kopīgu mērvienību visiem $j=1, 2, \ldots, n$; b) ienākums $f_j(x_j)$, kuru iegūst no j-tās tehnoloģijas pielietošanas, nav atkarīgs no tā, kādi resursu daudzumi izdalīti izmantošanai pēc citām iespējamām tehnoloģijām; c) kopīgo ienākumu no resursa izmantošanas pēc dažādām iespējamām tehnoloģijām var noteikt kā to ienākumu summu, kurus iegūst no atsevišķām tehnoloģijām. Šiem nosacījumiem un interpretācijai pilnā mērā atbilst uzdevuma 12.4.1° formulējums.

Ievērojot uzdevuma 12.4.1° ekonomisko interpretāciju, šo uzdevumu var aplūkot arī kā n soļu vadības procesu, kurā attiecīgo lēmumu izdalīt noteiktu resursa daudzumu x_j, kas izmantojams pēc j-tās tehnoloģijas, pieņem j-tajā solī. Uz šo faktu tad arī pamatojas dinamiskās programmēšanas metode. Vispirms risina viensoļa, tad divsoļu, trīssoļu uzdevumu utt., kamēr galu galā nonāk pie n soļu uzdevuma.

Lai tādu uzdevumu hierarhiju izveidotu, fiksē vienu no iespējamām x_n vērtībām un aprēķina a_nx_n.

No uzdevuma 12.4.1° formulējuma izriet, ka

$$\sum_{j=1}^{n-1} a_jx_j\leqslant b-a_nx_n \tag{12.4.1}$$

un

$$\max_{x_1, \ldots, x_{n-1}} \sum_{j=1}^{n-1} f_j(x_j)=\varphi_{n-1}(b-a_nx_n). \tag{12.4.2}$$

Maksimums izteiksmē (12.4.2) jāaprēķina pēc visiem tiem veselajiem nenegatīvajiem x_j $(j=1,2,\ldots,n-1)$, kuri apmierina nevienādību (12.4.1). Pieņemot, ka ir atrasta funkcijas $\varphi_{n-1}(b-a_n x_n)$ vērtība visiem pieļaujamiem x_n, t. i., x_n vērtībām 0, 1, 2, ..., $[b/a_n]$, var aprēķināt visiem x_n atbilstošās vērtības funkcijai

$$F_n(x_n)=f_n(x_n)+\varphi_{n-1}(b-a_n x_n). \qquad (12.4.3)$$

Lielākā no $F_n(x_n)$ vērtībām ir dotās mērķa funkcijas z maksimālā vērtība

$$z^*=\max_{x_n} F_n(x_n). \qquad (12.4.4)$$

Izteiksmē (12.4.3) nezināma ir funkcija $\varphi_{n-1}(b-a_n x_n)$. Lai to atrastu, ievērosim, ka jebkuram veselam pozitīvam skaitlim ξ ir spēkā vienādība

$$\varphi_{n-1}(\xi)=\max_{x_1,\ldots,x_{n-1}} \sum_{j=1}^{n-1} f_j(x_j), \qquad (12.4.5)$$

kur maksimums jāaprēķina pēc visiem veselajiem nenegatīvajiem skaitļiem x_j $(j=1,2,\ldots,n-1)$, kuri apmierina nevienādību

$$\sum_{j=1}^{n-1} a_j x_j \leqslant \xi. \qquad (12.4.6)$$

To ievērojot, iegūstam vienādību

$$\varphi_{n-1}(\xi)=\max_{x_{n-1}} [f_{n-1}(x_{n-1})+\varphi_{n-2}(\xi-a_{n-1}x_{n-1})], \qquad (12.4.7)$$

kur

$$\varphi_{n-2}(\eta)=\max_{x_1,\ldots,x_{n-2}} \sum_{j=1}^{n-2} f_j(x_j). \qquad (12.4.8)$$

Maksimums izteiksmē (12.4.8) jāaprēķina pēc visiem tiem veselajiem nenegatīvajiem x_j $(j=1,2,\ldots,n-2)$, kuri apmierina nevienādību

$$\sum_{j=1}^{n-2} a_j x_j \leqslant \eta. \qquad (12.4.9)$$

Lielumam x_{n-1} izteiksmē (12.4.7) var būt jebkura no vērtībām 0, 1, 2, ..., $[\xi/a_{n-1}]$. Tātad, ja zinām funkciju $\varphi_{n-2}(\eta)$, tad var aprēķināt funkciju $\varphi_{n-1}(\xi)$. Jāatzīmē, ka dažādām skaitļa ξ vērtībām, kurām nosaka atbilstošo funkcijas $\varphi_{n-1}(\xi)$ vērtību, jāatkārto maksimuma aprēķināšanas operācija izteiksmē (12.4.7).

Lai atrastu nezināmo funkciju $\varphi_{n-2}(\xi-a_{n-1}x_{n-1})$ izteiksmē

(12.4.7), tad tikko aprakstītais process jāturpina līdz pēdējam solim, kurā atrod funkciju

$$\varphi_1(\varrho) = \max_{x_1} f_1(x_1), \tag{12.4.10}$$

kur x_1 vērtības var būt $0, 1, \ldots, [\varrho/a_1]$.

Tātad, lai atrisinātu doto uzdevumu, jāsāk ar $\varphi_1(\varrho)$ aprēķināšanu. Pēc tam jāaprēķina $\varphi_2(\zeta), \ldots, \varphi_{n-2}(\eta)$, $\varphi_{n-1}(\xi)$ un beidzot z^*.

Dinamiskās programmēšanas metodes efektivitāti raksturo visu mērķa funkcijas maksimuma noteikšanai pieļaujamo plāna variantu skaita salīdzinājums ar saskaitīšanas darbību skaitu, kas jāizpilda, lai vajadzīgo maksimumu atrastu. Veselo nenegatīvo skaitļu kopā, kuras elementi apmierina uzdevuma 12.4.1° nosacījumus, ir pavisam C^b_{n+b-1} elementu. Ja $n=5$ un $b=20$, tad $C_{24}^4 = 10\,626$. Lietojot dinamiskās programmēšanas metodi, tāda uzdevuma 12.4.1° atrisināšanai, kurā vektoram $\boldsymbol{x}$ ir n koordinātes un ierobežotā resursa kopīgais daudzums ir b, vajadzīgo saskaitīšanas darbību skaitu var noteikt ar formulu $s = b\left[n + \frac{(n-1)(b+1)}{2}\right] + n$. Ja $n=5$ un $b=20$, iegūstam $s=945$. Kaut arī šis skaitlis ir ļoti liels, tas tomēr ir tikai nepilni 10% no tā skaitļošanas darba daudzuma, kas vajadzīgs, lai ar tiešu salīdzināšanu atrastu optimālo plāna variantu. Dinamiskās programmēšanas metodes efektivitāte kļūst vēl būtiskāka pie lieliem n, jo katrā solī visas neperspektīvās nezināmo kombinācijas var izslēgt no tālākās to attīstības.

Dinamiskās programmēšanas metodes pielietošanas ilustrācijai aplūkosim konkrētu diskrētās programmēšanas uzdevumu, kas pazīstams ar nosaukumu «ceļojošā tirgoņa uzdevums». Pieņemsim, ka ar 0, 1, 2, 3, 4 apzīmētas 5 vietas un ka attālumi starp šīm vietām doti tabulā (12.4.11):

	0	1	2	3	4
0	0	300	250	200	400
1	300	0	500	350	600
2	250	500	0	250	200
3	200	350	250	0	250
4	400	600	200	250	0

(12.4.11)

U z d e v u m s 12.4.2°. Izmantojot tabulā (12.4.11) doto informāciju, sastādīt visīsāko ceļojuma maršrutu, kas sākas un beidzas vietā 0, ja ceļotājam jāapmeklē katra no pārējām 4 vietām.

A t r i s i n ā j u m s. Ceļojuma maršrutā ir pavisam $n=5$ posmi. Principā tādu uzdevumu varētu atrisināt, sastādot visus iespējamos $5! = 120$ maršrutu variantus, bet tāds atrisināšanas paņēmiens pie liela n kļūst praktiski neizpildāms. Tādēļ dotā uzdevuma atrisināšanu var sākt ar mazāka posmu skaita kombināciju salīdzināšanu, izslēdzot neperspektīvās nezināmo kombinācijas no tālākās attīstības. Visu iespējamo trīsposmu kombināciju novērtējumi apkopoti tabulā (12.4.12):

Varianti	km		Varianti	km	
0 2 3 1	850	+	0 1 2 3	1050	+
0 3 2 1	950		0 2 1 3	1100	
0 2 4 1	1050	+	0 1 4 3	1150	+
0 4 2 1	1100		0 4 1 3	1350	
0 3 4 1	1050		0 2 4 3	700	+
0 4 3 1	1000	+	0 4 2 3	850	
0 1 3 2	900	+	0 1 2 4	1000	+
0 3 1 2	1050		0 2 1 4	1350	
0 1 4 2	1100	+	0 1 3 4	900	+
0 4 1 2	1500		0 3 1 4	1150	
0 3 4 2	650	+	0 2 3 4	750	
0 4 3 2	900		0 3 2 4	650	+

(12.4.12)

Tabulā (12.4.12) ar + atzīmēti tie varianti, kurus ir jēga turpināt. Varianti, kam tādas zīmes nav, ir neperspektīvi. Tie no turpmākās attīstīšanas izslēdzami, jo nav jēgas attīstīt, piemēram, maršrutu 0 2 1 3, kura garums 1100 km, ja maršruta 0 1 2 3 garums ir 1050 km un abi šie maršruti satur vienas un tās pašas vietas.

Attīstot un novērtējot tabulas (12.4.12) perspektīvos maršrutus, iegūstam tabulu (12.4.13):

Varianti	km		Varianti	km	
0 2 3 1 4	1450		0 1 2 3 4	1350	
0 2 4 1 3	1400		0 1 4 3 2	1400	
0 4 3 1 2	1500		0 2 4 3 1	1050	+
0 1 3 2 4	1100	+	0 1 2 4 3	1250	+
0 1 4 2 3	1350		0 1 3 4 2	1100	+
0 3 4 2 1	1150		0 3 2 4 1	1250	

(12.4.13)

Attīstot tabulā (12.4.13) ar + atzīmētos perspektīvos maršrutus, iegūstam tabulu (12.4.14):

Varianti	km
0 1 2 3 4 0	1500
0 2 4 3 1 0	1350
0 1 2 4 3 0	1450
0 1 3 4 2 0	1350

(12.4.14)

Atbilde. Optimālie ceļojuma maršruti ir divi. Pirmajā no tiem vietu secība ir 0 2 4 3 1 0, otrajā tā ir 0 1 3 4 2 0. Minimālais ceļa kopgarums katrā no šiem maršrutiem ir 1350 km.

13. nodaļa

SPĒĻU TEORIJAS ELEMENTI

13.1. §. Spēļu teorijas pamatjēdzieni. Pieņemsim, ka pastāv visiem zināmi *noteikumi* (likumi, tradīcijas, apstākļi), kas ir saistoši kāda *konflikta* dalībniekiem. Ar terminu «konflikts» sapratīsim parādību, kuras dalībniekiem ir dažādas intereses un iespēja izvēlēties dažādus darbības variantus atbilstoši savām interesēm. Matemātikas nozari, kura aplūko konfliktā esošo dalībnieku optimālo lēmumu pieņemšanas formālos modeļus, sauc par *spēļu teoriju.* Spēle šādā nozīmē ir konflikta matemātiskais modelis. Lai aprakstītu konfliktu spēles formā, tad jāzina, kas un kā piedalās spēlē, kādi ir iespējamie spēles iznākumi, kas un kādā veidā ir ieinteresēts spēles iznākumos. Konflikta dalībniekus sauc par *spēles koalīcijām;* koalīciju iespējamos darbības variantus — par to *tīrajām stratēģijām,* bet iespējamos konflikta iznākumus — par *situācijām.* Parasti ar terminu «situācija» saprot rezultātu, kāds rodas, kad katra koalīcija pielietojusi kādu no savām stratēģijām. Konflikta iznākumā ieinteresētos dalībniekus sauc par *interešu koalīcijām;* to intereses pastāv tādās vai citādās situācijas priekšrocībās, kuras bieži izteic ar skaitliskiem *laimestiem.* Dažādo spēļu klasifikācija ir atkarīga no pieminēto objektu konkretizācijas un savstarpējām sakarībām, kādas pastāv starp šiem objektiem.

Aplūkosim *divu koalīciju (spēlētāju) matricveida spēli ar summu 0.* Tā ir spēle, kura atbilst šādiem noteikumiem:

a) divi pretinieki (spēles koalīcijas) A un B var izmantot kāda mērķa sasniegšanai vairākas tīrās stratēģijas; A rīcībā esošo tīro stratēģiju skaits ir m, bet B rīcībā ir n stratēģijas;

b) katrs spēles dalībnieks pazīst visas iespējamās pretinieka stratēģijas, bet nezina, kuru no tām dotajā situācijā pretinieks izmantos;

c) ja A izmanto savu i-to stratēģiju, bet B savu j-to stratēģiju, tad A laimests ir p_{ij} *norēķina vienības,* kuras zaudē B, resp., B laimests ir $-p_{ij}$; abu pretinieku laimestu summa ir nulle, jo

$$p_{ij}+(-p_{ij})=0;$$

d) visi iespējamie laimesti p_{ij} $(i=1, 2, \ldots, m;\ j=1, 2, \ldots, n)$, kas sakārtoti m rindās un n kolonnās, veido norēķinu matricu

$$\boldsymbol{P}=\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ p_{m1} & p_{m2} & \cdots & p_{mn} \end{bmatrix}$$

Matricveida spēle ir pilnīgi definēta, ja dota norēķinu matrica. Tā, piemēram, ja dotā norēķinu matrica ir

$$\boldsymbol{P}=\begin{bmatrix} 1 & -1 & 3 & 0 \\ -1 & 2 & 0 & -1 \\ 3 & 0 & -1 & 1 \end{bmatrix} \qquad (13.1.1)$$

tad A rīcībā ir 3, bet B rīcībā 4 tīrās stratēģijas. Katrs matricas (13.1.1) elements ir norēķina vienību skaits, kuru iegūst A un zaudē B. Ja, piemēram, A lieto savu pirmo, bet B savu trešo tīro stratēģiju, tad laimesta lielums ir 3 norēķina vienības. Tās iegūst A un zaudē B. Ja A lieto savu pirmo, bet B otro tīro stratēģiju, tad laimesta lielums ir -1 norēķina vienība. Tas nozīmē, ka A zaudē, bet B iegūst vienu norēķina vienību.

Līdz šim tika pieņemts, ka aplūkotajā matricveida spēlē katra spēles koalīcija noteiktā situācijā izmantojusi vienu vienīgu reizi kādu no savām tīrajām stratēģijām.

Turpmāk aplūkosim t. s. *jaukto stratēģiju* jēdzienu, kas saistīts ar tīro stratēģiju lietošanas varbūtību sadalījumu. Ja ar x_i $(i=1, 2, \ldots, m)$ apzīmē varbūtību (biežumu), ar kuru spēles koalīcija A izvēlas savu i-to tīro stratēģiju, bet ar u_j $(j=1, 2, \ldots, n)$ — varbūtību, ar kuru B izvēlas savu j-to tīro stratēģiju, tad vektoru $\boldsymbol{x}=(x_1, x_2, \ldots, x_m)$ sauc par koalīcijas A, bet vektoru $\boldsymbol{u}=(u_1, u_2, \ldots, u_n)$ par koalīcijas B jaukto stratēģiju. Tā, piemēram, ja matricveida spēle ir definēta ar norēķinu matricu (13.1.1) un koalīcijas B jauktā stratēģija ir $\boldsymbol{u}=(0{,}1;\ 0{,}5;\ 0;\ 0{,}4)$, tad tas nozīmē, ka B izmanto savu pirmo tīro stratēģiju 10% gadījumu, otro 50%, trešo 0% un ceturto tīro stratēģiju 40% gadījumu. Jauktās stratēģijas acīm redzami apmierina šādus nosacījumus:

$$\begin{aligned} &\text{a) } \boldsymbol{x}\geqslant\boldsymbol{o}, \quad \text{b) } \boldsymbol{u}\geqslant\boldsymbol{o}, \\ &\text{c) } \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{e}^{\mathrm{T}}\boldsymbol{u}=1, \end{aligned} \qquad (13.1.2)$$

kur $\boldsymbol{e}^{\mathrm{T}}$ ir vektors $(1, 1, \ldots, 1)$, t. i., $\boldsymbol{e}=(1, 1, \ldots, 1)^{\mathrm{T}}$.

Ja matricveida spēle ir definēta ar norēķinu matricu $\boldsymbol{P}$ un koalīcijas A jauktā stratēģija ir $\boldsymbol{x}$, bet koalīcijas B jauktā stratēģija ir $\boldsymbol{u}$, tad koalīcijas A laimesta matemātisko cerību izteic skaitlis $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{u}$. Tā, piemēram, ja spēle ir definēta ar matricu (13.1.1), A jauktā stratēģija ir $\boldsymbol{x}=(0{,}4;\ 0;\ 0{,}6)$ un B jauktā stratēģija ir $\boldsymbol{u}=(0{,}1;\ 0{,}5;\ 0;\ 0{,}4)$, tad A laimesta matemātiskā cerība ir

$$(0{,}4;\ 0;\ 0{,}6)\begin{bmatrix} 1 & -1 & 3 & 0 \\ -1 & 2 & 0 & -1 \\ 3 & 0 & -1 & 1 \end{bmatrix}(0{,}1;\ 0{,}5;\ 0;\ 0{,}4)^{\mathrm{T}}=$$

$$=(2{,}2;\ -0{,}4;\ 0{,}6;\ 0{,}6)\,(0{,}1;\ 0{,}5;\ 0;\ 0{,}4)^{\mathrm{T}}=0{,}26.$$

Tas nozīmē, ka koalīcijas A laimests vienā spēlē šādā gadījumā vidēji ir 0,26 norēķina vienības.

13.2. §. Dažas matricveida spēļu pamatteorēmas. Lemma 1°. *Ja $\boldsymbol{P}=(p_{ij})_{(m\times n)}$ ir divu koalīciju A un B spēles norēķinu matrica, tad*

$$\min p_{ij}\leqslant\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{u}\leqslant\max p_{ij}, \qquad (13.2.1)$$

kur $\boldsymbol{x}$ un $\boldsymbol{u}$ attiecīgi ir A un B jauktās stratēģijas.

P i e r ā d ī j u m s. Atvietojot visus matricas $\boldsymbol{P}$ elementus ar $\min p_{ij}$, iegūstam matricu $\boldsymbol{P}'=(\min p_{ij})(\boldsymbol{e})_{(m\times n)}$ un pastāv nevienādība $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{u}\geqslant\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}'\boldsymbol{u}=\min p_{ij}$, jo $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}'\boldsymbol{u}=[\boldsymbol{x}^{\mathrm{T}}(\boldsymbol{e})\boldsymbol{u}]\min p_{ij}=\min p_{ij}$. Analogi, atvietojot visus matricas $\boldsymbol{P}$ elementus ar $\max p_{ij}$, iegūst matricu $\boldsymbol{P}''=(\max p_{ij})(\boldsymbol{e})_{(m\times n)}$ un pastāv nevienādība $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{u}\leqslant\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}''\boldsymbol{u}=\max p_{ij}$. Tādēļ $\min p_{ij}\leqslant\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{u}\leqslant\max p_{ij}$, ko arī vajadzēja pierādīt.

Iepriekšējā paragrāfā aplūkotajā skaitliskajā piemērā atrastā laimesta matemātiskā cerība 0,26, kā viegli pārliecināties, atbilst tikko pierādītajai lemmai 1°, jo $-1<0{,}26<3$.

D e f i n ī c i j a 1°. *Matricveida spēli, kas definēta ar norēķinu matricu $\boldsymbol{P}$, sauc par atrisināmu, ja eksistē tāds skaitlis v un vektori $\boldsymbol{x}_0$ un $\boldsymbol{u}_0$, ka visām iespējamām jauktajām stratēģijām $\boldsymbol{x}$ un $\boldsymbol{u}$ pastāv nevienādības*

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{u}_0\leqslant v\leqslant\boldsymbol{x}_0{}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{u}. \tag{13.2.2}$$

Skaitli v sauc par *spēles cenu*, bet vektorus $\boldsymbol{x}_0$ un $\boldsymbol{u}_0$ — par *optimālajām stratēģijām*.

Ja spēle ir atrisināma, tad, pielietojot stratēģiju $\boldsymbol{x}_0$, koalīcijas A laimesta summa nevar būt mazāka par spēles cenu v neatkarīgi no tā, kādu no savām jauktajām stratēģijām $\boldsymbol{u}$ pielieto koalīcija B. Analogi, pielietojot stratēģiju $\boldsymbol{u}_0$, koalīcijas B zaudējums nevar būt lielāks par spēles cenu v neatkarīgi no tā, kādu no savām jauktajām stratēģijām $\boldsymbol{x}$ pielieto koalīcija A.

L e m m a 2°. *Ja spēle ir atrisināma, tad tās cena ir*

$$v=\boldsymbol{x}_0{}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{u}_0. \tag{13.2.3}$$

P i e r ā d ī j u m s. Saskaņā ar definīciju 1° pastāv nevienādības $\boldsymbol{x}_0{}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{u}_0\leqslant v\leqslant\boldsymbol{x}_0{}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{u}_0$. No tā izriet, ka $v=\boldsymbol{x}_0{}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{u}_0$, ko arī vajadzēja pierādīt.

L e m m a 3°. *Ja spēle ir atrisināma, tad nevienādības* (13.2.2) *ir ekvivalentas ar nevienādību sistēmu*

$$\left.\begin{aligned} v\boldsymbol{e}-\boldsymbol{P}^{\mathrm{T}}\boldsymbol{x}_0 &\leqslant\boldsymbol{o},\\ v\boldsymbol{e}-\boldsymbol{u}_0{}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}} &\geqslant\boldsymbol{o}^{\mathrm{T}},\end{aligned}\right\} \tag{13.2.4}$$

kur $\boldsymbol{e}=(1,1,\ldots,1)^{\mathrm{T}}$.

P i e r ā d ī j u m s. Ievietojot nevienādību (13.2.2) labajā pusē $\boldsymbol{u}=\boldsymbol{e}_j$, kur $\boldsymbol{e}_j$ ir vienības vektors, kam j-tā koordināte ir 1, bet visas pārējās koordinātes vienādas ar 0, iegūst (ņemot $j=1,2,\ldots$ $\ldots,n$) nevienādību sistēmu

$$\left.\begin{aligned} v&\leqslant\boldsymbol{x}_0{}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{e}_1,\\ v&\leqslant\boldsymbol{x}_0{}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{e}_2,\\ &\cdots\cdots\\ v&\leqslant\boldsymbol{x}_0{}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{e}_n.\end{aligned}\right\} \tag{13.2.5}$$

Nevienādību sistēmu (13.2.5) var atvietot ar nevienādību

$$ve^{\mathrm{T}} \leqslant \boldsymbol{x}_0{}^{\mathrm{T}}\boldsymbol{P}$$

jeb

$$v\boldsymbol{e} - \boldsymbol{P}^{\mathrm{T}}\boldsymbol{x}_0 \leqslant \boldsymbol{o}. \qquad (13.2.6)$$

Izteiksme (13.2.6) ir pirmā nevienādība sistēmā (13.2.4). Pilnīgi analogi pierādāms, ka no nevienādībām (13.2.2) izriet arī nevienādība $ve^{\mathrm{T}} - \boldsymbol{u}_0{}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}} \geqslant \boldsymbol{o}^{\mathrm{T}}$.

Tagad spriedīsim otrādi. Pareizinot sistēmas (13.2.4) pirmo nevienādību no kreisās puses ar $\boldsymbol{u}^{\mathrm{T}}$ un otro nevienādību no labās puses ar $\boldsymbol{x}$, iegūstam nevienādību sistēmu

$$\left.\begin{aligned} v\boldsymbol{u}^{\mathrm{T}}\boldsymbol{e} - \boldsymbol{u}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{x}_0 &\leqslant \boldsymbol{u}^{\mathrm{T}}\boldsymbol{o}, \\ v\boldsymbol{e}^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{u}_0{}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{x} &\geqslant \boldsymbol{o}^{\mathrm{T}}\boldsymbol{x}. \end{aligned}\right\} \qquad (13.2.7)$$

Tā kā $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{e} = \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x} = 1$ un $\boldsymbol{u}^{\mathrm{T}}\boldsymbol{o} = \boldsymbol{o}^{\mathrm{T}}\boldsymbol{x} = 0$, tad no nevienādībām (13.2.7) izriet, ka

$$\left.\begin{aligned} v - \boldsymbol{u}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{x}_0 &\leqslant 0, \\ v - \boldsymbol{u}_0{}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{x} &\geqslant 0. \end{aligned}\right\} \qquad (13.2.8)$$

Apvienojot nevienādības (13.2.8), iegūstam

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{u}_0 \leqslant v \leqslant \boldsymbol{x}_0{}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{u}.$$

Tātad no nevienādībām (13.2.2) izriet nevienādības (13.2.4), un otrādi — no (13.2.4) izriet (13.2.2), t. i., šīs nevienādības ir ekvivalentas, ko arī vajadzēja pierādīt.

Definīcija 2°. *Divas spēles, kas definētas ar norēķinu matricām $\boldsymbol{P}$ un $\boldsymbol{Q}$, sauc par stratēģiski ekvivalentām, ja katra pirmās spēles optimālā stratēģija ir arī otrās spēles optimālā stratēģija — un otrādi.*

Lemma 3°. *Divas spēles, kas definētas ar norēķinu matricām $\boldsymbol{P}$ un $\boldsymbol{Q}$, ir stratēģiski ekvivalentas, ja $\boldsymbol{P} = \boldsymbol{Q} + \boldsymbol{Q}'$, kur $\boldsymbol{Q}' = k \cdot (\boldsymbol{e})_{(m \times n)}$ ir matrica, kuras katrs elements ir konstants skaitlis k.*

Pierādījums. Ievietojot izteiksmē (13.2.2) matricas $\boldsymbol{P}$ vietā matricu $\boldsymbol{Q} + \boldsymbol{Q}'$, iegūst

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{u}_0 + k[\boldsymbol{x}^{\mathrm{T}}(\boldsymbol{e})\boldsymbol{u}_0] \leqslant v \leqslant \boldsymbol{x}_0{}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{u} + k[\boldsymbol{x}_0{}^{\mathrm{T}}(\boldsymbol{e})\boldsymbol{u}_0]. \qquad (13.2.9)$$

Tā kā

$$\boldsymbol{x}^{\mathrm{T}}(\boldsymbol{e})\boldsymbol{u}_0 = \boldsymbol{x}_0{}^{\mathrm{T}}(\boldsymbol{e})\boldsymbol{u} = 1,$$

tad no (13.2.9) izriet, ka

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{u}_0 + k \leqslant v \leqslant \boldsymbol{x}_0{}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{u} + k$$

jeb

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{u}_0 \leqslant v' \leqslant \boldsymbol{x}_0{}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{u}_0,$$

kur $v' = v - k$. Lemma pierādīta.

No lemmas 3° pierādījuma redzams, ka *otrās spēles cena v' atšķiras no pirmās spēles cenas v par k norēķina vienībām.*

Teorēma 1°. *Katra matricveida spēle ir atrisināma*.*

Pierādījums. Saskaņā ar lemmu 3° varam pieņemt, ka spēles norēķinu matricas $\boldsymbol{P}=(p_{ij})_{(m\times n)}$ visi elementi ir nenegatīvi. Ja tas tā nebūtu, tad visiem šīs matricas elementiem vajadzētu pieskaitīt skaitli $k=|\max p_{ij}|\ (p_{ij}<0)$. Tādēļ pieņemsim, ka

$$\boldsymbol{P}\geqslant \boldsymbol{O} \tag{13.2.10}$$

un aplūkosim divus savstarpēji duālos lineārās programmēšanas uzdevumus:

$$\left.\begin{array}{l} f=y\rightarrow\max, \\ \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x}=1, \\ \boldsymbol{e}y-\boldsymbol{P}^{\mathrm{T}}\boldsymbol{x}\leqslant\boldsymbol{o}, \\ y\geqslant 0,\ \boldsymbol{x}\geqslant\boldsymbol{o}, \end{array}\right\} \tag{13.2.11}$$

un

$$\left.\begin{array}{l} w=z\rightarrow\min, \\ \boldsymbol{u}^{\mathrm{T}}\boldsymbol{e}=1, \\ z\boldsymbol{e}^{\mathrm{T}}-\boldsymbol{u}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}}\geqslant\boldsymbol{o}^{\mathrm{T}}, \\ z\geqslant 0;\ \boldsymbol{u}^{\mathrm{T}}\geqslant\boldsymbol{o}^{\mathrm{T}}. \end{array}\right\} \tag{13.2.12}$$

Tiešā uzdevuma (13.2.11) atbalsta plāns ir vektors $(y\mid\boldsymbol{x}^{\mathrm{T}})^{\mathrm{T}}=(0\mid\boldsymbol{e}_1^{\mathrm{T}})^{\mathrm{T}}$, bet duālā uzdevuma (13.2.12) atbalsta plāns ir vektors $(z\mid\boldsymbol{u}^{\mathrm{T}})=(\mu\mid\boldsymbol{e}_1^{\mathrm{T}})$, kur μ ir maksimālais elements matricas $\boldsymbol{P}$ pirmajā kolonnā. Tiešām, ja ņemam vērā (13.2.10), tad vektors $(0\mid\boldsymbol{e}_1^{\mathrm{T}})^{\mathrm{T}}$ apmierina visus uzdevuma (13.2.11) ierobežojumus un vektors $(\mu\mid\boldsymbol{e}_1^{\mathrm{T}})$ savukārt apmierina visus uzdevuma (13.2.12) ierobežojumus, jo

$$\left.\begin{array}{l} \boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}_1=1, \\ \boldsymbol{e}\cdot 0-\boldsymbol{P}^{\mathrm{T}}\boldsymbol{e}_1=-\boldsymbol{P}^{\mathrm{T}}\boldsymbol{e}_1\leqslant\boldsymbol{o}, \\ 0\equiv 0,\ \boldsymbol{e}_1\geqslant 0 \end{array}\right\}$$

un

$$\left.\begin{array}{l} \boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{e}=1, \\ \mu\boldsymbol{e}^{\mathrm{T}}-\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}}\geqslant\boldsymbol{o}^{\mathrm{T}}, \\ \mu\geqslant 0,\ \boldsymbol{e}_1^{\mathrm{T}}\geqslant\boldsymbol{o}^{\mathrm{T}}. \end{array}\right\}$$

* Šo teorēmu literatūrā sauc par Neimana teorēmu. Dž. fon Neimana un O. Morgenšterna 1944. gadā izdotajā grāmatā tika sistemātiski izstrādāta spēļu teorija kā līdzeklis matemātiskai pieejai ekonomiskajā konkurencē.

Uzdevumu (13.2.11), (13.2.12) vienlaicīgu atrisināšanu var sākt ar parastās simpleksa metodes iterāciju tabulu:

1	-1	y	x_1	x_2	$\cdots$	x_m	(I)
s	-1	0	-1	-1	$\cdots$	-1	M
f	0	-1	0	0	$\cdots$	0	0
$\bar{o}$	1	0	$\boxed{1}$	1	$\cdots$	1	1
u_1	0	1					2
u_2	0	1					3
.	.	.		$-\boldsymbol{P}^{\mathrm{T}}$			.
.	.	.					.
.	.	.					.
u_n	0	1					$n+1$
σ	σ_0	σ'	σ_1	σ_2	$\cdots$	σ_m	$n+2$

(13.2.13)

Kā redzams, mākslīgais bāzes mainīgais, kas tabulā (13.2.13) apzīmēts ar $\bar{o}$, izslēdzas jau pēc pirmā iterāciju soļa. Jau pēc viena iterāciju soļa arī sekundārā mērķa funkcija s sasniedz savu maksimumu, kas vienāds ar 0. Tādēļ jau otrajā iterācijas tabulā nolasāmi gan tiešā, gan arī duālā uzdevuma atbalsta plāni. Tabulas (13.2.13) $n+2$-ajā rindā ierakstītas kolonnu kontrolsummas.

Tā kā tiešajam un duālajam uzdevumam eksistē atbalsta plāni, tad saskaņā ar dualitātes teorēmām šādam uzdevumu pārim eksistē arī optimālie atrisinājumi. Apzīmēsim optimālā plāna vektoru tiešajam uzdevumam (13.2.11) ar $(y_0 \mid \boldsymbol{x}_0^{\mathrm{T}})^{\mathrm{T}}$ un duālajam uzdevumam (13.2.12) ar $(z_0 \mid \boldsymbol{u}_0)$. Šo uzdevumu optimālās mērķa funkciju vērtības tad ir $\max f = y_0$ un $\min w = z_0$. Šīs vērtības saskaņā ar pirmo dualitātes teorēmu ir vienādas. Apzīmēsim tās ar

$$v = y_0 = z_0. \tag{13.2.14}$$

Lai pierādītu, ka optimālo plānu vektoru bloki $\boldsymbol{x}_0$ un $\boldsymbol{u}_0^{\mathrm{T}}$ ir ar norēķinu matricu (13.2.10) definētās spēles optimālās stratēģijas, ievērosim, ka saskaņā ar uzdevumu (13.2.11), (13.2.12) ierobežojumiem pastāv vienādības

$$\left.\begin{aligned} \boldsymbol{e}^{\mathrm{T}}\boldsymbol{x}_0 &= 1, \\ \boldsymbol{u}_0^{\mathrm{T}}\boldsymbol{e} &= 1 \end{aligned}\right\} \tag{13.2.15}$$

un nevienādības

$$\left.\begin{aligned} v\boldsymbol{e} - \boldsymbol{P}^{\mathrm{T}}\boldsymbol{x}_0 &\leqslant \boldsymbol{o}, \\ v\boldsymbol{e}^{\mathrm{T}} - \boldsymbol{u}_0^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}} &\geqslant \boldsymbol{o}^{\mathrm{T}}. \end{aligned}\right\} \tag{13.2.16}$$

Nevienādības (13.2.16) un (13.2.4) ir identiskas. Vienādības (13.2.15) atbilst nosacījumiem (13.1.2). Tādēļ v ir ar matricu (13.2.10) definētās spēles cena, bet $\boldsymbol{x}_0$ ir koalīcijas A un $\boldsymbol{u}_0$ koalīcijas B optimālā stratēģija. Tā kā šādas stratēģijas un spēles cenu var vienmēr atrast, atrisinot lineārās programmēšanas uzdevumus (13.2.11), (13.2.12), tad jebkura spēle, kas definēta ar norēķinu matricu (13.2.10), ir atrisināma, ko arī vajadzēja pierādīt.

No teorēmas 1° pierādījuma izriet šāds secinājums. *Katras matricveida spēles cena ir noteikta viennozīmīgi, bet var būt tādas norēķinu matricas, ka vienai vai abām spēles koalīcijām eksistē vairākas optimālās stratēģijas.*

13.3. §. Skaitlisks piemērs. Dota spēle ar norēķinu matricu (13.1.1). Atrast spēles cenu un abu spēles koalīciju optimālās stratēģijas.

Atrisinājums. Tā kā matricai (13.1.1) ir arī daži negatīvi elementi, tad atrisināsim dotajai spēlei stratēģiski ekvivalentu spēli, kuras norēķinu matrica ir

$$\boldsymbol{P}' = \begin{bmatrix} 2 & 0 & 4 & 1 \\ 0 & 3 & 1 & 0 \\ 4 & 1 & 0 & 2 \end{bmatrix} \qquad (13.3.1)$$

Matrica $\boldsymbol{P}'$ iegūta no matricas $\boldsymbol{P}$, pieskaitot katram tās elementam skaitli 1. Tas nozīmē, ka ar matricu $\boldsymbol{P}'$ definētās spēles cena ir par vienu norēķina vienību lielāka nekā dotā uzdevuma cena. Apzīmēsim ar $\boldsymbol{x}^{\mathrm{T}} = (x_1, x_2, x_3)$ koalīcijas A un ar $\boldsymbol{u}^{\mathrm{T}} = (u_1, u_2, u_3, u_4)$ koalīcijas B jaukto stratēģiju. Saskaņā ar 13.2. § teorēmas 1° pierādījumu uzdevuma atrisinājumu iegūstam ar parasto simpleksa metodes algoritmu, kā parādīts tabulās (13.3.2)—(13.3.6).

1	-1	y	x_1	x_2	x_3	(I)
s	-1	0	-1	-1	-1	M
f	0	-1	0	0	0	0
$\bar{o}$	1	0	$\boxed{1}$	1	1	1
u_1	0	1	-2	0	-4	2
u_2	0	1	0	-3	-1	3
u_3	0	1	-4	-1	0	4
u_4	0	1	-1	0	-2	5
σ	-1	2	-8	-5	-8	6

(13.3.2)

1	-1	y	x_2	x_3	(II)
s	0	0	0	0	M
f	0	-1	0	0	0
x_1	1	0	1	1	1
u_1	2	1	2	-2	2
u_2	0	$\boxed{1}$	-3	-1	3
u_3	4	1	3	4	4
u_4	1	1	1	-1	5
σ	7	2	3	0	6

(13.3.3)

1	-1	u_2	x_2	x_3	(III)
f	0	1	-3	-1	0
x_1	1	0	1	1	1
u_2	2	-1	5	-1	2
y	0	1	-3	-1	3
u_3	4	-1	6	5	4
u_4	1	-1	$\boxed{4}$	0	5
σ	7	-2	9	2	6

(13.3.4)

1	-1	u_2	u_4	x_3	(IV)
f	0,75	0,25	0,75	-1	0
x_1	0,75	0,25	$-0{,}25$	1	1
u_1	0,75	0,25	$-1{,}25$	-1	2
y	0,75	0,25	0,75	-1	3
u_3	2,50	0,50	$-1{,}50$	$\boxed{5}$	4
x_2	0,25	$-0{,}25$	0,25	0	5
σ	4,75	0,25	$-2{,}25$	2	6

(13.3.5)

1	-1	u_2	u_4	u_3	(V)
f	1,25	0,35	0,45	0,20	0
x_1	0,25	0,15	0,05	$-0{,}20$	1
u_1	1,25	0,35	$-1{,}55$	0,20	2
y	1,25	0,35	0,45	0,20	3
x_3	0,50	0,10	$-0{,}30$	0,20	4
x_2	0,25	$-0{,}25$	0,25	0	5
σ	3,75	0,05	$-1{,}65$	$-0{,}40$	6

(13.3.6)

Tabulā (13.3.6) nolasāms tiešā lineārās programmēšanas uzdevuma optimālais atrisinājums $\max f = y = 1{,}25$, ja $x_1 = x_2 = 0{,}25$ un $x_3 = 0{,}50$. Šajā tabulā nolasāms arī atbilstošā duālā uzdevuma optimālais atrisinājums $\min w = \max f = 1{,}25$, ja $u_1 = 0$; $u_2 = 0{,}35$; $u_3 = 0{,}20$ un $u_4 = 0{,}45$. Tas nozīmē, ka dotajam uzdevumam stratēģiski ekvivalentās un ar norēķinu matricu (13.3.1) definētās spēles cena ir $v = 1{,}25$. Vienīgās optimālās stratēģijas šajā spēlē ir vektori

$$\left.\begin{aligned} \boldsymbol{x}_0 &= (0{,}25;\ 0{,}25;\ 0{,}50)^{\mathrm{T}}, \\ \boldsymbol{u}_0 &= (0;\ 0{,}35;\ 0{,}20;\ 0{,}45)^{\mathrm{T}}. \end{aligned}\right\} \tag{13.3.7}$$

Dotajā matricveida spēlē, kas definēta ar norēķinu matricu (13.1.1), vienīgās optimālās stratēģijas ir vektori (13.3.7), bet spēles cena ir $v = 1{,}25 - 1 = 0{,}25$.

13.4. §. Lineārā programmēšana kā spēļu teorijas speciālgadījums. Pieņemsim, ka matricveida spēle ir definēta ar antisimetrisku norēķinu matricu $\boldsymbol{P}$, t. i., ar tādu kvadrātisku matricu, kuras transponētā matrica $\boldsymbol{P}^{\mathrm{T}}$ ir vienāda ar $-\boldsymbol{P}$.

Definīcija 1°. *Matricveida spēli sauc par simetrisku, ja tā ir definēta ar antisimetrisku norēķinu matricu* $\boldsymbol{P} = -\boldsymbol{P}^{\mathrm{T}}$.

Teorēma 1°. *Ar norēķinu matricu* $\boldsymbol{P}$ *definētas simetriskas spēles optimālā stratēģija ir vektors* $\boldsymbol{x}_0$ *tad un tikai tad, ja*

$$\boldsymbol{P}\boldsymbol{x}_0 \leqslant \boldsymbol{o}. \tag{13.4.1}$$

Pierādījums. Ievietojot nevienādībās (13.2.4) $v = 0$, $\boldsymbol{u}_0 = \boldsymbol{x}_0$ un $-\boldsymbol{P}^{\mathrm{T}} = \boldsymbol{P}$, iegūstam nevienādību (13.4.1). Tā kā katra matricveida spēle ir atrisināma (13.2. § teorēma 1°), tad teorēma ir pierādīta.

No teorēmas 1° pierādījuma izriet šāds secinājums. *Simetriskas spēles cena* v *ir vienāda ar nulli un abu spēles koalīciju optimālās stratēģijas* $\boldsymbol{x}_0$ *un* $\boldsymbol{u}_0$ *ir vienādas.*

Pieņemsim, ka ir doti divi savstarpēji duāli lineārās programmēšanas uzdevumi

$$\left.\begin{array}{llll} & \text{I} & & \text{II} \\ \text{a)} & f = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \to \max, & \text{a)} & w = \boldsymbol{u}^{\mathrm{T}}\boldsymbol{b} \to \min, \\ \text{b)} & \boldsymbol{A}\boldsymbol{x} \leqslant \boldsymbol{b}, & \text{b)} & \boldsymbol{u}^{\mathrm{T}}\boldsymbol{A} \geqslant \boldsymbol{c}^{\mathrm{T}}, \\ \text{c)} & \boldsymbol{x} \geqslant \boldsymbol{o}, & \text{c)} & \boldsymbol{u}^{\mathrm{T}} \geqslant \boldsymbol{o}^{\mathrm{T}}. \end{array}\right\} \tag{13.4.2}$$

No šo uzdevumu konstantēm var izveidot antisimetrisku blokmatricu

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{o} & \boldsymbol{A} & -\boldsymbol{b} \\ -\boldsymbol{A}^{\mathrm{T}} & \boldsymbol{o} & \boldsymbol{c} \\ \boldsymbol{b}^{\mathrm{T}} & -\boldsymbol{c}^{\mathrm{T}} & 0 \end{bmatrix} \tag{13.4.3}$$

Pieņemot matricu (13.4.3) par simetriskas spēles norēķinu matricu, apzīmēsim šīs spēles optimālo stratēģiju ar 3 blokos sadalītu vektoru

$$\boldsymbol{x}_0 = (\boldsymbol{r}_0^{\mathrm{T}} \mid \boldsymbol{s}_0^{\mathrm{T}} \mid t_0)^{\mathrm{T}}. \tag{13.4.4}$$

Saskaņā ar teorēmu 1° vektoram (13.4.4) jāapmierina nevienādība

$$\begin{bmatrix} \boldsymbol{o} & \boldsymbol{A} & -\boldsymbol{b} \\ -\boldsymbol{A}^{\mathrm{T}} & \boldsymbol{o} & \boldsymbol{c} \\ \boldsymbol{b}^{\mathrm{T}} & -\boldsymbol{c}^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_0 \\ \boldsymbol{s}_0 \\ t_0 \end{bmatrix} \leqslant \boldsymbol{o}. \tag{13.4.5}$$

Nevienādība (13.4.5) ir identiska ar trīs nevienādību sistēmu

$$\left\{ \begin{array}{ll} \boldsymbol{A}\boldsymbol{s}_0 \leqslant \boldsymbol{b}t_0, & (13.4.6) \\ \boldsymbol{r}_0^{\mathrm{T}}\boldsymbol{A} \geqslant \boldsymbol{c}^{\mathrm{T}}t_0, & (13.4.7) \\ \boldsymbol{b}^{\mathrm{T}}\boldsymbol{r}_0 \leqslant \boldsymbol{c}^{\mathrm{T}}\boldsymbol{s}_0. & (13.4.8) \end{array} \right.$$

T e o r ē m a 2°. *Ja vektora (13.4.4) koordināte t_0 ir pozitīva, tad uzdevumiem (13.4.2) eksistē optimālie atrisinājumi.*

P i e r ā d ī j u m s. Ja $t_0 > 0$, tad, reizinot vektorus $\boldsymbol{s}_0$ un $\boldsymbol{r}_0$ ar $1/t_0$, iegūst vektorus

$$\frac{1}{t_0}\boldsymbol{s}_0 = \boldsymbol{y}_0 \geqslant \boldsymbol{o} \tag{13.4.9}$$

un

$$\frac{1}{t_0}\boldsymbol{r}_0 = \boldsymbol{z}_0 \geqslant \boldsymbol{o}. \tag{13.4.10}$$

Dalot nevienādības (13.4.6) un (13.4.7) ar t_0, iegūst $\boldsymbol{A}\boldsymbol{y}_0 \leqslant \boldsymbol{b}$ un $\boldsymbol{z}_0^{\mathrm{T}}\boldsymbol{A} \geqslant \boldsymbol{c}^{\mathrm{T}}$. Tas nozīmē, ka vektors $\boldsymbol{y}_0$ ir maksimizācijas, bet $\boldsymbol{z}_0^{\mathrm{T}}$ minimizācijas uzdevuma (13.4.2) atbalsta plāns. Dalot nevienādību (13.4.8) ar t_0, iegūst nevienādību

$$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{z}_0 \leqslant \boldsymbol{c}^{\mathrm{T}}\boldsymbol{y}_0, \tag{13.4.11}$$

bet saskaņā ar 5.2. § lemmu 1° jāizpildās nevienādībai

$$\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y}_0 \leqslant \boldsymbol{b}^{\mathrm{T}}\boldsymbol{z}_0. \tag{13.4.12}$$

No izteiksmēm (13.4.11), (13.4.12) izriet, ka $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y}_0 = \boldsymbol{b}^{\mathrm{T}}\boldsymbol{z}_0$. Tas nozīmē, ka $\boldsymbol{y}_0$ ir maksimizācijas, bet $\boldsymbol{z}_0$ minimizācijas uzdevuma (13.4.2) optimālais plāns. Tātad šiem uzdevumiem eksistē optimālais atrisinājums, ko arī vajadzēja pierādīt.

T e o r ē m a 3°. *Ja vektori (13.4.9), (13.4.10) ir uzdevumu (13.4.2) optimālie plāni, tad ar matricu (13.4.3) definētās simetriskās spēles optimālā stratēģija ir vektors (13.4.4).*

Pierādījums. Ja vektori y_0 un z_0 ir uzdevumu (13.4.2) optimālie plāni, tad pastāv sakarības

$$\left.\begin{array}{l} Ay_0 \leqslant b, \\ z_0^T A \geqslant c^T, \\ c^T y_0 = z_0^T b, \\ y_0 \geqslant o,\ z_0 \geqslant o \end{array}\right\}$$

vai

$$\left.\begin{array}{rrrl} oz_0 & +Ay_0 & -b & \leqslant o, \\ -A^T z_0 & +oy_0 & +c & \leqslant o, \\ b^T z_0 & -c^T y_0 & +0 & =0. \end{array}\right\} \qquad (13.4.13)$$

Tā kā vektoru y_0 un z_0 koordinātes ir nenegatīvas, tad eksistē pozitīvs skaitlis

$$t_0 = \frac{1}{e^T z_0 + e^T y_0 + 1} > 0$$

un izpildās sakarības $t_0 z_0 = r_0$, $t_0 y_0 = s_0$, $e^T r_0 + e^T s_0 + t_0 = 1$.

Reizinot izteiksmes (13.4.13) ar t_0, iegūst sakarību sistēmu

$$\left.\begin{array}{rrrl} or_0 & +As_0 & -bt_0 & \leqslant o, \\ -A^T r_0 & +os_0 & +ct_0 & \leqslant o, \\ b^T r_0 & -c^T s_0 & +0t_0 & = 0 \end{array}\right\}$$

jeb

$$\begin{bmatrix} o & A & -b \\ -A^T & o & c \\ b^T & -c^T & 0 \end{bmatrix} \begin{bmatrix} r_0 \\ s_0 \\ t_0 \end{bmatrix} \leqslant o. \qquad (13.4.14)$$

Saskaņā ar teorēmu 1° izteiksme (13.4.14) nozīmē, ka ar matricu (13.4.3) definētās simetriskās spēles optimālā stratēģija ir vektors $x_0 = (r_0^T \mid s_0^T \mid t_0)^T$, ko arī vajadzēja pierādīt.

Lineārās programmēšanas uzdevums tātad ir speciālgadījums divu koalīciju (spēlētāju) matricveida spēlei ar summu 0. Katram lineārās programmēšanas uzdevumam atbilst pilnīgi noteikta simetriskā spēle. Tā vienmēr ir atrisināma, un tās optimālo stratēģiju var izteikt formā (13.4.4). Lineārās programmēšanas uzdevumiem (13.4.2), kas atbilst ar matricu (13.4.3) definētai simetriskai spēlei, eksistē optimāls atrisinājums tikai tad, ja spēles optimālās stratēģijas koordināte t_0 ir pozitīva.

SATURS

Ievads 3

0.1. §. Matemātiskās programmēšanas uzdevumi 3
0.2. §. Īsas vēsturiskas ziņas 4

I daļa. Lineārās programmēšanas un tīklveida plānošanas elementi

1. nodaļa. Dažu lineārās programmēšanas piemēru ekonomiskā interpretācija un matemātiskais formulējums 5

1.1. §. Transporta uzdevums 5
1.2. §. Ierobežotu ražošanas resursu un ražošanas tehnoloģisko procesu izmantošanas optimizācija 9
1.3. §. Barības devu optimizācija 11
1.4. §. Ražošanas jaudu izmantošanas optimizācija 12

2. nodaļa. Lineārās programmēšanas vispārīgie jēdzieni . . . 13

2.1. §. Lineārās programmēšanas vispārīgais uzdevums un tā modifikācijas 13
2.2. §. Lineārās programmēšanas uzdevumu algebriskās transformācijas 14
2.3. §. Lineārās programmēšanas uzdevumu ģeometriskā interpretācija 17
2.4. §. Plāna, optimālā plāna un atbalsta plāna jēdzieni . . . 23

3. nodaļa. Dažas lineāru algebrisku vienādojumu sistēmu skaitliskās atrisināšanas metodes 24

3.1. §. Lineāru algebrisku vienādojumu sistēma 24
3.2. §. Modificētā Žordāna izslēgšanas metode 25
3.3. §. Lineāru vienādojumu sistēmas risināšanas gaitas kontrole 33
3.4. §. Gausa—Žordāna metode 34

4. nodaļa. Simpleksa metode lineārajā programmēšanā . . . 36

4.1. §. Uzdevuma normālforma 36
4.2. §. Simpleksa metodes parastais algoritms 40

4.3. §. Vispārīgā formā dota lineārās programmēšanas uzdevuma atrisināšana ar simpleksa metodi 45
4.4. §. Simpleksa metodes saīsināto iterāciju tabulu pieraksts 50
4.5. §. Simpleksa algoritma monotonitāte, singularitāte un galīgums 56

5. nodaļa. Dualitāte lineārajā programmēšanā 57

5.1. §. Duālie uzdevumi 57
5.2. §. Dualitātes teorēmas 66
5.3. §. Lineārās programmēšanas modeļu jutīguma analīze . . 70
5.4. §. Kantoroviča atrisinošo reizinātāju metode 76

6. nodaļa. Transporta uzdevuma atrisināšanas metodes . . . 81

6.1. §. Transporta uzdevuma ierobežojumu sistēmas un mērķa funkcijas īpašības 81
6.2. §. Atbalsta plāns transporta uzdevumā 87
6.3. §. Atbalsta plāna pakāpeniska uzlabošana ar sadalījumu metodi 91
6.4. §. Potenciālu metode un dualitāte 97
6.5. §. Aproksimācijas metode 101
6.6. §. Diferenciālo renšu metode 103

7. nodaļa. Tīklveida plānošanas elementi 109

7.1. §. Vispārīgie jēdzieni par grafu un tīklu 109
7.2. §. Sakārtota grafa virsotņu numerācija 111
7.3. §. Darbu plāna attēlošana ar tīkla grafiku 114
7.4. §. Tīklveida plāna izpildes dekretētais laiks, minimālais laiks un kritiskais ceļš 117
7.5. §. Tīklveida plāna parametri un to atrašanas algoritmi . . 119

8. nodaļa. Parametriskā un daļveida lineārā programmēšana . . 124

8.1. §. Matemātiskās programmēšanas uzdevums, kura lineārās mērķa funkcijas koeficienti ir viena parametra lineāras funkcijas 124
8.2. §. Matemātiskās programmēšanas uzdevums, kura lineārās ierobežojumu sistēmas brīvie locekļi ir viena parametra lineāras funkcijas 130
8.3. §. Daļveida lineārā programmēšana 131

II daļa. Lineārās programmēšanas teorijas papildinājumi

9. nodaļa. Lineārās programmēšanas uzdevumu plānu kopa . . 137

9.1. §. *n*-dimensiju vektoru telpas un izliektas punktu kopas jēdzieni 137
9.2. §. Matricu algebras elementi 142
9.3. §. Lineārās programmēšanas plānu kopas īpašības 144
9.4. §. Atbalsta plānu veidošana 149
9.5. §. Pirmās dualitātes teorēmas pierādījums 151

9.6. §. Singularitātes problēma 152
9.7. §. Transporta uzdevuma matricas rangs 153

10. nodaļa. Simpleksa metodes varianti 154

10.1. §. Duālā simpleksa metode 154
10.2. §. Modificētā simpleksa metode 156

III daļa. Nelineārās programmēšanas un spēļu teorijas elementi

11. nodaļa. Nelineārās programmēšanas vispārīgie jēdzieni . . 164

11.1. §. Dažādi matemātiskās programmēšanas uzdevumu tipi . . 164
11.2. §. Jēdziens par globālo un lokālo maksimumu un minimumu 166
11.3. §. Nelineārās programmēšanas uzdevuma ekonomiskās interpretācijas piemērs 167
11.4. §. Nelineārās programmēšanas uzdevumu ģeometriskās interpretācijas piemēri 168

12. nodaļa. Dažas nelineārās programmēšanas uzdevumu atrisināšanas metodes 170

12.1. §. Klasiskā nelineārās programmēšanas uzdevumu atrisināšanas metode 170
12.2. §. Jēdziens par gradienta metodi 171
12.3. §. Gomori metode diskrētajā programmēšanā 177
12.4. §. Jēdziens par dinamiskās programmēšanas metodi . . . 181

13. nodaļa. Spēļu teorijas elementi 186

13.1. §. Spēļu teorijas pamatjēdzieni 186
13.2. §. Dažas matricveida spēļu pamatteorēmas 187
13.3. §. Skaitlisks piemērs 192
13.4. §. Lineārā programmēšana kā spēļu teorijas speciālgadījums 194

А. Ф. Крастынь

МАТЕМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

Допущено Министерством высшего и среднего специального образования Латвийской ССР в качестве учебника для студентов экономических специальностей высших учебных заведений Латвийской ССР при изучении курса математического программирования

Издательство «Звайгзне»

Рига 1976

На латышском языке

Alberts Krastiņš

MATEMĀTISKĀ PROGRAMMĒŠANA

I izdevums

Vāku zīm. *J. Meijers*

Redaktors *A. Licis.* Māksl. redaktors *U. Gulbis.* Tehn. redaktore *K. Kazačenko.* Korektore *V. Bruņeniece.*

Nodota salikšanai 1975. g. 6. augustā. Parakstīta iespiešanai 1976. g. 15. jūlijā. Papīra formāts 60×90/16. Tipogr. papīrs Nr. 3. 12,5 fiz. iespiedl.; 12,5 uzsk. iespiedl.; 11,10 izdevn. l. Metiens 2000 eks. JT 05349. Maksā 52 kap. Izdevniecība «Zvaigzne» Rīgā, Gorkija ielā 105. Izdevn. Nr. 3443-TM-101. Iespiesta Latvijas PSR Ministru Padmes Valsts izdevniecību, poligrāfijas un grāmatu tirdzniecības lietu komitejas tipogrāfijā «Cīņa» Rīgā, Blaumaņa ielā 38/40. Pasūt. Nr. 2808-Ļ.

52 kap.