Optimizacijske metode
6. Linearne naloge

Vladimir Batagelj

Univerza v Ljubljani
FMF, matematika

na vrhu

Kazalo

Linearne optimizacijske naloge

Linearna optimizacijska naloga ali *naloga linearnega programiranja* je naloga (Φ, P, Min), pri kateri je kriterijska funkcija linearna

$$P(x) = \sum_{j=1}^{n} c_j x_j$$

in je tudi množica dopustnih rešitev $\Phi \subseteq \mathbb{R}^n$ določena z linearnimi omejitvami (in zato konveksna):

$$\Phi = \{x : \begin{array}{ll} \sum_{j=1}^{n} a_{ij} x_j \geq b_i & i = 1, \ldots, k, \\ \sum_{j=1}^{n} a_{ij} x_j = b_i & i = k+1, \ldots, m, \\ x_j \geq 0, & j = 1, \ldots, t\} \end{array}$$

pri čemer so a_{ij}, b_i in c_j dana realna števila.

Tej obliki naloge linearnega programiranja pravimo *splošna*.

Posebne oblike nalog

Pogosto srečamo še naslednje tri posebne oblike:

- *osnovna*: $k = m, t = 0$;
- *standardna*: $k = m, t = n$;
- *kanonska*: $k = 0, t = n$.

Pokažimo, da za vsako nalogo v splošni obliki obstaja enakovredna naloga v vsaki izmed teh treh oblik.

Osnovna oblika: Neenakost $x_j \geq 0$ je le poseben primer neenakosti $\sum_{j=1}^{n} a_{ij}x_j \geq b_i$; enakost $\sum_{j=1}^{n} a_{ij}x_j = b_i$ pa je enakovredna neenakostima

$$\sum_{j=1}^{n} a_{ij}x_j \geq b_i \quad \text{in} \quad -\sum_{j=1}^{n} a_{ij}x_j \geq -b_i$$

…Posebne oblike nalog

Standardna oblika: Pokazati moramo, da se lahko omejimo na nenegativne spremenljivke. Spremenljivko $x \in \mathbb{R}$ lahko nadomestimo z razliko dveh nenegativnih spremenljivk

$$x = y - z, \qquad y, z \in \mathbb{R}_0^+$$

Pogoj oblike $x \leq 0$ pa uženemo z zameno $x = -y,\ y \in \mathbb{R}_0^+$. To zameno je smiselno opraviti že pri pretvarjanju v osnovno obliko.

Kanonska oblika: Neenakosti lahko pretvorimo v enakosti z uvedbo nenegativnih *dopolnitvenih* spremenljivk. Neenakost $\sum_{j=1}^n a_{ij}x_j \geq b_i$ je tako enakovredna enakosti

$$\sum_{j=1}^n a_{ij}x_j - s = b_i, \quad s \in \mathbb{R}_0^+$$

Primer pretvorb

$$P(x_1, x_2, x_3) = 2x_1 + 3x_2 + 4x_3$$

$$\begin{aligned} 4x_1 - 5x_2 - \ \ x_3 &\leq 1 \\ 3x_1 - 4x_2 + 2x_3 &= 6 \\ x_1 &\geq 0 \\ x_2 &\leq 0 \end{aligned}$$

osnovna

$$P(x_1, x_2', x_3) = 2x_1 - 3x_2' + 4x_3$$

$$\begin{aligned} -4x_1 - 5x_2' + \ \ x_3 &\geq -1 \\ 3x_1 + 4x_2' + 2x_3 &\geq \ \ 6 \\ -3x_1 - 4x_2' - 2x_3 &\geq -6 \\ x_1 &\geq \ \ 0 \\ x_2' &\geq \ \ 0 \end{aligned}$$

...Primer pretvorb

standardna

$$P(x_1, x_2', y_3, z_3) = 2x_1 - 3x_2' + 4y_3 - 4z_3$$

$$\begin{aligned} -4x_1 - 5x_2' + \ \ y_3 - \ \ z_3 &\geq -1 \\ 3x_1 + 4x_2' + 2y_3 - 2z_3 &\geq \ \ 6 \\ -3x_1 - 4x_2' - 2y_3 + 2z_3 &\geq -6 \\ x_1 &\geq \ \ 0 \\ x_2' &\geq \ \ 0 \\ y_3 &\geq \ \ 0 \\ z_3 &\geq \ \ 0 \end{aligned}$$

kanonska

$$P(x_1, x_2', y_3, z_3, s) = 2x_1 - 3x_2' + 4y_3 - 4z_3$$

$$\begin{aligned} 4x_1 + 5x_2' - \ \ y_3 + \ \ z_3 + s &= 1 \\ 3x_1 + 4x_2' + 2y_3 - 2z_3 \qquad &= 6 \\ x_1 &\geq 0 \\ x_2' &\geq 0 \\ y_3 &\geq 0 \\ z_3 &\geq 0 \\ s &\geq 0 \end{aligned}$$

...Posebne oblike nalog

V nadaljnjem bomo privzeli, da je naloga $LP = (\Phi, P, \mathrm{Min})$ podana v kanonski obliki

$$P(x) = cx \quad \text{in} \quad \Phi = \{x : Ax = b, x \geq 0\}$$

pri čemer so $x \in \mathbb{R}^{n\times 1}$, $c \in \mathbb{R}^{1\times n}$, $A \in \mathbb{R}^{m\times n}$ in $b \in \mathbb{R}^{m\times 1}$.

Predpostavimo lahko, da je $\mathrm{rank}(A) = m$. Če ni, so enačbe ali odvisne in jih lahko nekaj odvržemo, ali pa so protislovne in je $\Phi = \emptyset$.

Geometrija linearnega programiranja

Množica Φ je konveksna. Pravimo ji *konveksni politop*. Omejenemu konveksnemu politopu pravimo tudi *konveksni polieder*. *Oglišče* politopa je točka, ki je ni mogoče izraziti kot konveksno kombinacijo drugih njegovih točk. Vsako točko konveksnega poliedra je mogoče izraziti kot konveksno kombinacijo njegovih oglišč.

Vektor $y \geq 0$ je *žarek*, če zanj velja

$$\forall x \in \Phi \forall \lambda \geq 0 : x + \lambda y \in \Phi$$

Iz pogoja izhaja, da je $y \geq 0$ žarek natanko takrat, ko je $Ay = 0$.

Množica $\Psi \subseteq \Omega$ je *stožec*, če zadošča pogoju $\forall x \in \Psi \forall \lambda \geq 0 : \lambda x \in \Psi$.

Množica vseh žarkov $\{y : Ay = 0, y \geq 0\}$ je stožec.

Žarek je *skrajni* (ektremni), če ga ni mogoče zapisati kot pozitivno linearno kombinacijo od njega različnih žarkov.

... Geometrija linearnega programiranja

Vsako točko konveksnega politopa x je mogoče izraziti kot vsoto konveksne kombinacije njegovih oglišč $\{x_i\}$ in nenegativne kombinacije njegovih skrajnih žarkov $\{y_j\}$

$$x = \sum_{i=1}^{k} \alpha_i x_i + \sum_{j=1}^{l} \beta_j y_j$$

$$\alpha_i \geq 0, i \in 1..k, \quad \beta_j \geq 0, j \in 1..l, \quad \sum_{i=1}^{k} \alpha_i = 1$$

Poglejmo sedaj kriterijsko funkcijo

$$P(x) = cx = \sum_{i=1}^{k} \alpha_i c x_i + \sum_{j=1}^{l} \beta_j c y_j$$

Če je kak člen $cy_j < 0$, je $\min(\Phi, P) = -\infty$, saj lahko ustrezni β_j pošljemo proti ∞, ostale β pa postavimo na 0.

. . . Geometrija linearnega programiranja

Če pa so vsi $cy_j > 0$, bo prispevek te vsote najmanjši, ko so vsi $\beta_j = 0$. Prva vsota bo tedaj najmanjša, ko pri enem izmed členov cx_i z najmanjšo vrednostjo postavimo $\alpha_i = 1$, vsi ostali α pa so 0. Torej je $\min(\Phi, P) = cx_i$, kar pomeni, da

Če je naloga linearnega programiranja rešljiva, je vsaj eno oglišče pripadajočega politopa minimalna rešitev; če jih je več, so minimalne rešitve tudi vse njihove konveksne kombinacije.

Posamezno oglišče politopa je določeno z m neodvisnimi enačbami iz $Ax = b$. Ker iz n enačb lahko sestavimo $\binom{n}{m}$ sistemov, je lahko največ toliko različnih oglišč.

Torej moramo za rešitev naloge linearnega programiranja pregledati le končno množico oglišč pripadajočega politopa. Pravzaprav, niti te ne v celoti. Naloga linearnega programiranja je namreč konveksna. Zato je zanjo lokalna optimizacija točna metoda.

…Geometrija linearnega programiranja

Kako je v tem primeru določena relacija sosednosti?

Kot smo že rekli, je vsako oglišče določeno z m enačbami. Če eno odvzamemo, ostalih $m - 1$ enačb določa premico – nosilko roba politopa; če tem dodamo novo enačbo, dobimo novo točko na tej premici – možno sosednje oglišče. To je osnova *simpleksne metode*.

Pri razdelavi te metode moramo odgovoriti še na vprašanja:

- ali je naloga rešljiva?
- ali je dano oglišče minimalna rešitev?
- če nismo v minimumu, kako izbrati sosednje oglišče z manjšo vrednostjo kriterijske funkcije?

Teoretične osnove metode simpleksov

Nesingularno kvadratno podmatriko B, reda m, matrike A imenujemo *baza*. Ker je $\mathrm{rank}(A) = m$, obstaja vsaj ena baza.

Preuredimo stolpce tako, da bo $A = [B, N]$. Podobno preuredimo vektor $x = \begin{bmatrix} x_B \\ x_N \end{bmatrix}$, kjer so x_B bazične spremenljivke in x_N nebazične spremenljivke. Enačba $Ax = b$ dobi tedaj obliko $Bx_B + Nx_N = b$.

Rešitev x je *bazična*, če velja $x_N = 0$. Tedaj je $Bx_B = b$ in, ker je B nesingularna, $x_B = B^{-1}b$.

Bazična rešitev je *dopustna*, če je $x_B \geq 0$; tedaj tudi bazi B pravimo dopustna. Dopustnim bazičnim rešitvam ustrezajo oglišča politopa naloge. Bazična rešitev je *izrojena*, če vektor x_B vsebuje ničelne člene.

. . . Teoretične osnove metode simpleksov

Označimo $\overline{b} = B^{-1}b$ in $\overline{N} = B^{-1}N = [\overline{a}_{ij}]$. Za prevedbo sistema

$$[B, N]x = b$$

v *bazični* sistem

$$[I, \overline{N}]x = \overline{b}$$

uporabimo Gaussov postopek.

Razmeroma preprosto je prevesti dani bazični sistem v nov bazični sistem, ki ga dobimo s premeno bazične spremenljivke x_r z nebazično spremenljivko x_s, pri čemer je $\overline{a}_{rs} \neq 0$. Predelamo stolpec a_s v e_r in seveda pri tem pokvarimo stolpec a_r. Nato ju premenjamo.

. . . Teoretične osnove metode simpleksov

Za $i = 1, \ldots, m,\ i \neq r$

$$x_i + \frac{-\overline{a}_{is}}{\overline{a}_{rs}} x_r + \sum_{\substack{j=m+1 \\ j \neq s}}^{n} (\overline{a}_{ij} - \frac{\overline{a}_{is}}{\overline{a}_{rs}} \overline{a}_{rj}) x_j = \overline{b}_i - \frac{\overline{a}_{is}}{\overline{a}_{rs}} \overline{b}_r$$

in

$$\frac{1}{\overline{a}_{rs}} x_r + \sum_{\substack{j=m+1 \\ j \neq s}}^{n} \frac{\overline{a}_{rj}}{\overline{a}_{rs}} x_j + x_s = \frac{\overline{b}_r}{\overline{a}_{rs}}$$

Pripadajoča bazična rešitev je tedaj, ker je $x_N = 0$, za $i = 1, \ldots, m,\ i \neq r$

$$x_i = \overline{b}_i - \frac{\overline{a}_{is}}{\overline{a}_{rs}} \overline{b}_r \quad \text{in} \quad x_s = \frac{\overline{b}_r}{\overline{a}_{rs}}$$

Izreki: minimalnost

IZREK 1 *Dopustna bazična rešitev* $x^* = (x_B^*, x_N^*) = (B^{-1}b, 0)$ *naloge linearnega programiranja je minimalna, če velja*

$$\overline{c}_N \equiv c_N - c_B B^{-1} N \geq 0$$

Dokaz: Vzemimo poljubno rešitev $x = (x_B, x_N)$. Tedaj iz $Ax = b$ izhaja $x_B = B^{-1}b - B^{-1}Nx_N$. Upoštevajmo to v

$$\begin{aligned} P(x) &= [c_B, c_N](x_B, x_N) = c_B x_B + c_N x_N = \\ &= c_B B^{-1} b + (c_N - c_B B^{-1} N) x_N = c_B B^{-1} b + \overline{c}_N x_N \end{aligned}$$

Če je x^* dopustna bazična rešitev, je $x_N^* = 0$. Zato je $P(x^*) = c_B B^{-1} b$ in, če je $\overline{c}_N \geq 0$, je $\forall x \in \Phi : P(x) \geq P(x^*)$. $\square$

Izreki: neomejenost

Izrek nam omogoča z vpeljavo *popravljenih cen*

$$\bar{c}_j = c_j - c_B B^{-1} a_j, \quad j = 1, \ldots, n$$

ugotavljati optimalnost rešitve. Za $i = 1, \ldots, m$ je $\bar{c}_i = 0$.

IZREK 2 *Če za dopustni bazični sistem obstaja nebazična spremenljivka x_s, za katero je $\bar{c}_s < 0$ in $\bar{a}_{is} \leq 0$, $i = 1, \ldots, m$, tedaj je* $\min P(x) = -\infty$.

Dokaz: Ker so spremenljivke v x_N neodvisne, lahko postavimo vse razen x_s na 0. Tedaj iz $[I, \overline{N}]x = \bar{b}$ izhaja $x_i = \bar{b}_i - \bar{a}_{is} x_s, \ i = 1, \ldots, m$.

Če so $\bar{a}_{is} \leq 0$, lahko x_s poljubno povečamo ne da bi s tem porušili dopustnost rešitve x. No, v tem primeru je

$$P(x) = c_B B^{-1} b + \bar{c}_N x_N = c_B B^{-1} b + \bar{c}_s x_s$$

kar gre proti $-\infty$, ko gre x_s proti $+\infty$. □

Izreki: boljši sosed

IZREK 3 *Če v neizrojenem dopustnem bazičnem sistemu obstaja nebazična spremenljivka x_s, za katero je $\overline{c}_s < 0$ in za nek indeks i velja $\overline{a}_{is} > 0$, potem lahko določimo novo dopustno bazično rešitev, ki ima strogo manjšo vrednost kriterijske funkcije.*

Dokaz: V enačbah

$$x_i = \overline{b}_i - \overline{a}_{is} x_s, \quad i = 1, \ldots, m$$

povečamo x_s kar se da, tako da je rešitev x še dopustna, to je $x \geq 0$. To storimo tako, da postavimo $x_s = \vartheta_s$, kjer je

$$\vartheta_s = \min_{\overline{a}_{is} > 0} \frac{\overline{b}_i}{\overline{a}_{is}} = \frac{\overline{b}_r}{\overline{a}_{rs}}$$

Ker je rešitev neizrojena, je $\overline{b}_r > 0$ in zato tudi $\vartheta_s > 0$. Torej se vrednost $P(x)$ spremeni za $\overline{c}_s \vartheta_s < 0$.

. . . Izreki: boljši sosed

Tako dobimo novo bazično rešitev saj velja

$$\begin{aligned}
\overline{x}_i &= \overline{b}_i - \overline{a}_{is}\frac{\overline{b}_r}{\overline{a}_{rs}}, \quad i = 1, \dots, m, \quad i \neq r \\
\overline{x}_r &= 0 \\
\overline{x}_s &= \frac{\overline{b}_r}{\overline{a}_{rs}} = \vartheta_s \\
\overline{x}_j &= 0, \quad j = m+1, \dots, n, \quad j \neq s
\end{aligned}$$

kar predstavlja premeno nebazične spremenljivke x_s z bazično spremenljivko x_r. $\square$

Simpleksni algoritem

Dokazani izreki so osnova *simpleksnega algoritma* za reševanje nalog linearnega programiranja:

> določi začetni dopustni bazični sistem;
> **loop**
> $\overline{b} := B^{-1}b$; $u := c_B B^{-1}$; $\overline{c}_N := c_N - uN$;
> **if** $\overline{c} \geq 0$ **then** x je minimalna rešitev; **exit endif**;
> $s :=$ argmin$_j \overline{c}_j$; $\overline{a}_s := B^{-1} a_s$;
> **if** $\overline{a}_s \leq 0$ **then** min $:= -\infty$; **exit endif**;
> $r =$ argmin$_{i:\overline{a}_{is}>0} \dfrac{\overline{b}_i}{\overline{a}_{is}}$; $\vartheta_s = \dfrac{\overline{b}_r}{\overline{a}_{rs}}$;
> opravi premeno spremenljivk x_s in x_r – nova baza B
> **endloop**

...Simpleksni algoritem

Če je $\overline{b} > 0$, se na vsakem koraku vrednost kriterijske funkcije zmanjša – premaknemo se v novo, še neobiskano točko. Če je $\overline{b} \geq 0$, se lahko zgodi, da je $\vartheta = 0$ – premaknemo se v točko z isto vrednostjo kriterijske funkcije. Algoritem lahko zaide v cikel. Temu se lahko izognemo na več načinov. Eden izmed njih je uporaba *leksikografskega pravila* (Bland 1977):

- izberi $\overline{c}_s < 0$ z najmanjšim indeksom;
- med i, ki minimizirajo ϑ, izberi najmanjšega.

Dvofazna metoda

Kako dobimo začetni bazični sistem? Če je sistem v kanonski obliki, lahko predpostavimo, da je $b \geq 0$ (po potrebi kako enakost pomnožimo z -1).

Osnovni nalogi (Φ, P, Min): $P(x) = cx$, $\Phi = \{x : Ax = b, x \geq 0\}$ priredimo nalogo $(\Gamma, Q, \mathrm{Min})$ takole: naj bo $w \in \mathbb{R}^{m\times 1}$, tedaj je

$$Q(x, w) = \sum_{i=1}^{m} w_i \quad \text{in} \quad \Gamma = \{(x, w) : Ax + Iw = b,\ x, w \geq 0\}$$

Pokažimo, da ima osnovna naloga (Φ, P, Min) dopustno rešitev natanko takrat, ko je $\min(\Gamma, Q) = 0$.

Naj bo $(x^*, w^*) \in \mathrm{Min}(\Gamma, Q)$ in $\min(\Gamma, Q) = 0$. Tedaj je $w^* = 0$ in dalje $x^* \geq 0,\ Ax^* = b$. Torej je res $x^* \in \Phi$.

V obratno smer pa sklepamo takole. Naj bo $x^* \in \Phi$. Tedaj je $(x^*, 0) \in \Gamma$ in $Q(x^*, 0) = 0$. Ker je $Q(x, w) \geq 0$, je potemtakem res $\min(\Gamma, Q) = 0$.

No, za nalogo $(\Gamma, Q, \mathrm{Min})$ je $(0, b)$ vselej dopustna bazična rešitev.

Metoda velikega M

Nalogi (Φ, P, Min): $P(x) = cx$, $\Phi = \{x : Ax = b, x \geq 0\}$, $b \geq 0$ priredimo enakovredno nalogo linearnega programiranja $(\Gamma, Q, \mathrm{Min})$:

$$\begin{aligned} Q(x,w) &= cx + M\sum_{i=1}^{m} w_i \\ \Gamma &= \{(x,w) : Ax + Iw = b,\ x, w \geq 0\} \end{aligned}$$

kjer je M neko zelo veliko pozitivno število. Brez težav se lahko prepričamo, da velja:

- $x \in \Phi \Leftrightarrow (x, 0) \in \Gamma$
- $(0, b) \in \Gamma$
- $(x^*, w^*) \in \mathrm{Min}(\Gamma, Q) \quad \Leftrightarrow \quad t^* = 0 \wedge x^* \in \mathrm{Min}(\Phi, P)$

Simpleksna tabela

Naloga linearnega programiranja v kanonski obliki (Φ, P, Min):

$$P(x) = cx, \quad \Phi = \{x : Ax = b, x \geq 0\}$$

je enakovredna nalogi (Ψ, Q, Min):

$$Q(x, z) = z, \quad \Psi = \{(x, z) : cx - z = 0, Ax = b, x \geq 0\}$$

ki jo lahko uredimo v *simpleksno* tabelo

$$
\begin{array}{c|ccc|cccc|c}
 & x_1 & & x_m & x_{m+1} & & x_n & z & \\
\hline
L_0 & c_1 & \cdots & c_m & c_{m+1} & \cdots & c_n & -1 & 0 \\
\hline
L_1 & a_{11} & \cdots & a_{1m} & a_{1,m+1} & \cdots & a_{1n} & 0 & b_1 \\
\vdots & \vdots & B & \vdots & \vdots & N & \vdots & 0 & \vdots \\
L_m & a_{m1} & \cdots & a_{mm} & a_{m,m+1} & \cdots & a_{mn} & 0 & b_m \\
\hline
\end{array}
$$

. . . Simpleksna tabela

Pomnožimo tabelo z matriko $\begin{bmatrix} 1 & -c_B B^{-1} \\ 0 & B^{-1} \end{bmatrix}$

	x_1	x_n	z	
L_0	0	$\overline{c} = c_N - c_B B^{-1} N$	-1	$-c_B B^{-1} b = -P(x)$
L_1, $\vdots$, L_m	I	$\overline{N} = B^{-1} N$	0	$\overline{b} = B^{-1} b$

Iz tabele vidimo:

- bazično rešitev kar preberemo: ker je zanjo $x_N = 0$, je $x_B = \overline{b}$. Če je baza dopustna je $\overline{b} \geq 0$.
- v prvi vrstici so nad nebazičnimi spremenljivkami popravljene cene $\overline{c}$.
- negativno vrednost kriterijske funkcije za dano bazično rešitev preberemo v desnem zgornjem vogalu tabele.

. . . Simpleksna tabela

Za prvo vrstico velja enakost

$$\overline{c}_N x_N - z = -c_B B^{-1} b$$

in ker je za bazično rešitev $x_N = 0$, je

$$-z = -c_B B^{-1} = -Q(x, z) = -P(x)$$

Opomba:
Nekateri avtorji uporabljajo tabelo, ki temelji na enačbi $z - cx = 0$. Tedaj v desnem zgornjem vogalu dobimo pravo vrednost kriterijske funkcije $P(x)$; zato pa cene spremenijo predznak in ustrezno moramo prilagoditi tudi simpleksni algoritem.

Zgled

(Φ, P, Max)

$$P(x_1, x_2) = x_1 + 2x_2$$

$$\begin{aligned} -3x_1 + 2x_2 &\le 2 \\ -x_1 + 2x_2 &\le 4 \\ x_1 + x_2 &\le 5 \\ x_1 &\ge 0 \\ x_2 &\ge 0 \end{aligned}$$

$(\Phi, P, \mathrm{Max}) \approx (\Phi, -P, \mathrm{Min})$

kanonska (Ψ, Q, Min)

$$P(x_1, x_2, s_1, s_2, s_3, z) = z$$

$$\begin{aligned} -x_1 - 2x_2 - z &= 0 \\ -3x_1 + 2x_2 + s_1 &= 2 \\ -x_1 + 2x_2 + s_2 &= 4 \\ x_1 + x_2 + s_3 &= 5 \\ x_i \ge 0, &\quad i=1,2 \\ s_j \ge 0, &\quad j=1,2,3 \end{aligned}$$

. . . Zgled

Omejitveni stolpec je nenegativen, $b \geq 0$. Zato sestavljajo dopolnitvene spremenljivke s_1, s_2, s_3 bazo. V simpleksni tabeli bomo bazične spremenljivke označevali s puščicami.

	x_1	x_2	s_1	s_2	s_3	z	
L_0	-1	-2	0	0	0	-1	0
L_1	-3	$\boxed{2}$	1	0	0	0	2
L_2	-1	2	0	1	0	0	4
L_3	1	1	0	0	1	0	5
			$\uparrow$	$\uparrow$	$\uparrow$		

Ker nebazični c-ji niso vsi nenegativni, baza ni minimalna. Izberemo najmanjši $\overline{c}_i$. To je $\overline{c}(x_2) = -2$, kar da $x_s = x_2$. Ker pripadajoči stolpec $\overline{a}_2$, vsebuje pozitivne člene, pregledamo kvociente $\frac{\overline{b}_i}{\overline{a}_{i2}} \rightarrow \frac{2}{2}, \frac{4}{2}, \frac{5}{1}$ in izberemo najmanjšega. Torej je $x_r = s_1$.

...Zgled

Opravimo premeno bazične spremenljivke s_1 z nebazično spremenljivko x_2.

$$
\begin{array}{c|cccccc|c|l}
 & x_1 & x_2 & s_1 & s_2 & s_3 & z & & \\
\hline
L_0' & -4 & 0 & 1 & 0 & 0 & -1 & 2 & L_0 + L_1 \\
\hline
L_1' & -\frac{3}{2} & 1 & \frac{1}{2} & 0 & 0 & 0 & 1 & \frac{1}{2}L_1 \\
L_2' & \boxed{2} & 0 & -1 & 1 & 0 & 0 & 2 & L_2 - L_1 \\
L_3' & \frac{5}{2} & 0 & -\frac{1}{2} & 0 & 1 & 0 & 4 & L_3 - L_1' \\
\hline
 & & \uparrow & & \uparrow & \uparrow & & &
\end{array}
$$

Ker je $\overline{c}(x_1) = -4 < 0$, je $x_s = x_1$ in pripadajoči kvocienti $\frac{\overline{b}_i}{\overline{a}_{i1}} \to \_\_,\ \frac{2}{2},\ \frac{8}{5}$. Torej je $x_r = s_2$.

...Zgled

Opravimo premeno bazične spremenljivke s_2 z nebazično spremenljivko x_1.

$$
\begin{array}{c|cccccc|c|l}
 & x_1 & x_2 & s_1 & s_2 & s_3 & z & & \\
\hline
L_0'' & 0 & 0 & -1 & 2 & 0 & -1 & 6 & L_0' + 2L_2' \\
\hline
L_1'' & 0 & 1 & -\frac{1}{4} & \frac{3}{4} & 0 & 0 & \frac{5}{2} & L_1' + \frac{3}{4}L_2' \\
L_2'' & 1 & 0 & -\frac{1}{2} & \frac{1}{2} & 0 & 0 & 1 & \frac{1}{2}L_2' \\
L_3'' & 0 & 0 & \boxed{\frac{3}{4}} & -\frac{5}{4} & 1 & 0 & \frac{3}{2} & L_3' - \frac{5}{4}L_2' \\
\hline
 & \uparrow & \uparrow & & & \uparrow & & &
\end{array}
$$

Ker je $\overline{c}(s_1) = -1 < 0$, je $x_s = s_1$ in pripadajoči kvocienti $\frac{\overline{b}_i}{\overline{a}_{i1}} \rightarrow$ __, __, 2. Torej je $x_r = s_3$.

...Zgled

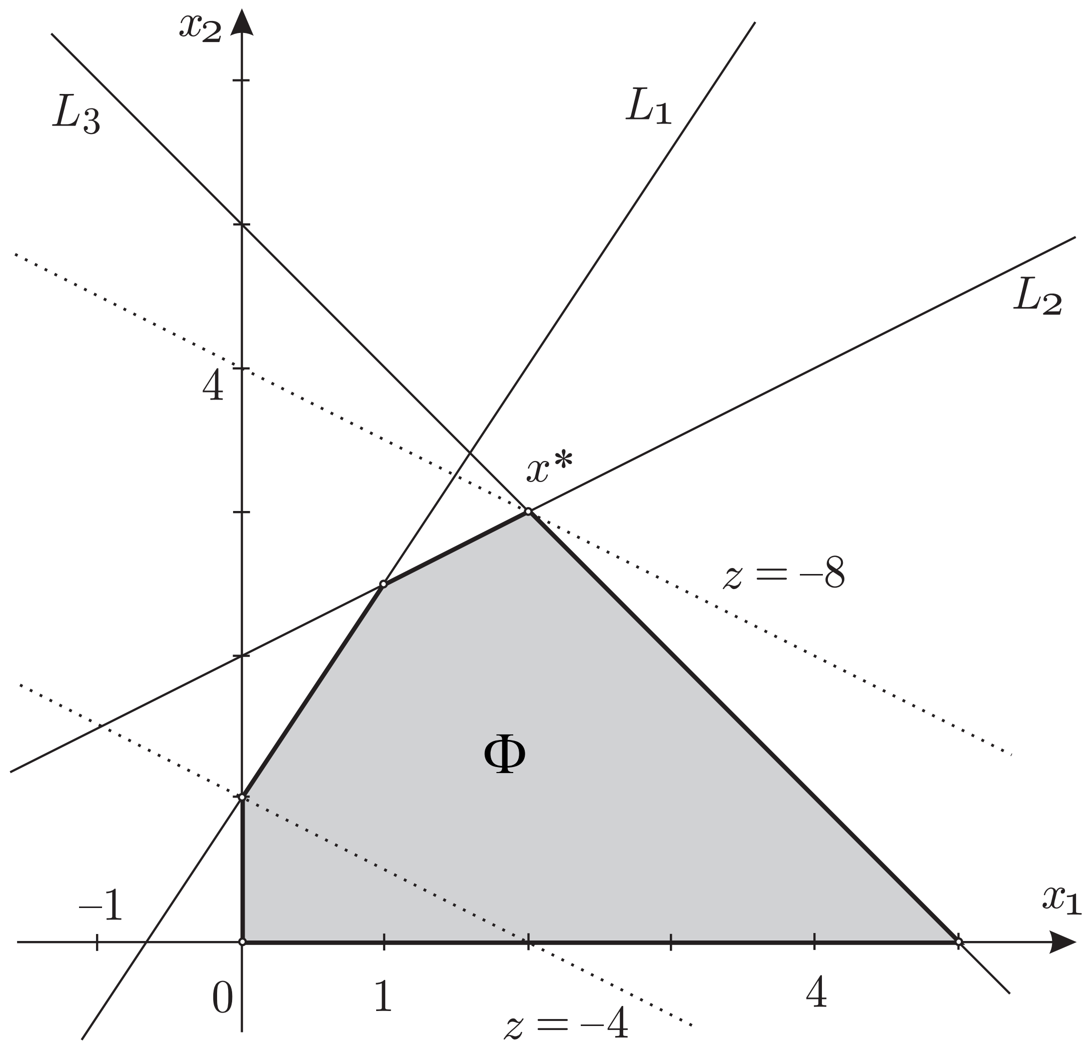
Opravimo premeno bazične spremenljivke s_3 z nebazično spremenljivko s_1.

$$
\begin{array}{c|cccccc|c|l}
 & x_1 & x_2 & s_1 & s_2 & s_3 & z & & \\
\hline
L_0''' & 0 & 0 & 0 & \frac{1}{3} & \frac{4}{3} & -1 & 8 & L_0'' + \frac{4}{3}L_3'' \\
\hline
L_1''' & 0 & 1 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 3 & L_1'' + \frac{1}{3}L_3'' \\
L_2''' & 1 & 0 & 0 & -\frac{1}{3} & \frac{2}{3} & 0 & 2 & L_2'' + \frac{2}{3}L_3'' \\
L_3''' & 0 & 0 & 1 & -\frac{5}{3} & \frac{4}{3} & 0 & 2 & \frac{4}{3}L_3'' \\
\hline
 & \uparrow & \uparrow & \uparrow & & & & &
\end{array}
$$

Ker je $\bar{c}_N \geq 0$ smo prišli v minimum

$$x^* = (2, 3, 2, 0, 0, -8) \quad \text{in} \quad P(x^*) = -8$$

. . . Zgled – slika



Dualnost v linearnem programiranju

Vzemimo nalogo linearnega programiranja $LP = (\Phi, P, \mathrm{Min})$ v standardni obliki

$$\Phi = \{x : Ax \geq b,\ x \geq 0\}$$

kjer je $x \in \mathbb{R}^{n\times 1}, b \in \mathbb{R}^{m\times 1}, c \in \mathbb{R}^{1\times n}$ in $A \in \mathbb{R}^{m\times n}$.

Njena Lagrangeova funkcija je

$$L(x,u) = cx + u(b - Ax) = (c - uA)x + ub, \quad u \geq 0$$

Določimo še prirejeno nalogo (Ψ, Q, Max)

$$Q(u) = \inf_{x\in(\mathbb{R}_0^+)^n} L(x,u) = \begin{cases} ub & c - uA \geq 0 \\ -\infty & \text{sicer} \end{cases}$$

Torej je

$$\Psi = \{u \in \mathbb{R}^{1\times n} : A^T u^T \leq c^T,\ u \geq 0\} \quad \text{in} \quad Q(u) = ub$$

Prirejana naloga je tudi v standardni obliki.

Izrek o dualnost

IZREK 4 *Prirejena naloga je rešljiva (dualna) natanko takrat, ko je rešljiva osnovna naloga. Če je ena naloga neomejena, je druga protislovna; ali pa sta obe nalogi protislovni.*

V primeru, ko sta nalogi rešljivi velja, kakor vemo, za $x^* \in \mathrm{Min}(\Phi, P)$ in $u^* \in \mathrm{Max}(\Psi, Q)$

$$P(x^*) = L(x^*, u^*) = Q(u^*)$$

oziroma $cx^* = cx^* + u^*(b - Ax^*) = (c - u^*A)x^* + u^*b = u^*b$ kar je enakovredno enakostima

$$u^*(Ax^* - b) = 0 \quad \text{in} \quad (c - u^*A)x^* = 0$$

Od tu tudi vidimo, da omejitvi oblike enakosti/neenakosti ustreza pozitivna/ničelna dualna spremenljivka, saj velja $\sum_i u_i^*(A_i x^* - b_i) = 0 \Leftrightarrow \forall i : u_i^*(A_i x^* - b_i) = 0 \Leftrightarrow \forall i : (u_i^* = 0) \vee (A_i x^* = b_i)$.

Dualnost – Zgled

Rešimo z uporabo dualnosti nalogo $LP = (\Phi, P, \mathrm{Min})$, kjer je

$$\Phi = \{(x, y, z) \in \mathbb{R}^3 : 4x - 5y - z \leq 1,\ 3x - 4y + 2z = 6,\ x \geq 0,\ y \leq 0\}$$

$$P(x, y, z) = 2x + 3y + 4z$$

Z zameno $y \to -y'$ jo najprej preuredimo v enakovredno nalogo v osnovni obliki $LP' = (\Phi', P', \mathrm{Min})$

$$\Phi' = \{(x, y', z) \in \mathbb{R}^3 : -4x - 5y' + z \geq -1,\ 3x + 4y' + 2z = 6,\ x \geq 0,\ y' \geq 0\}$$

$$P'(x, y', z) = 2x - 3y' + 4z$$

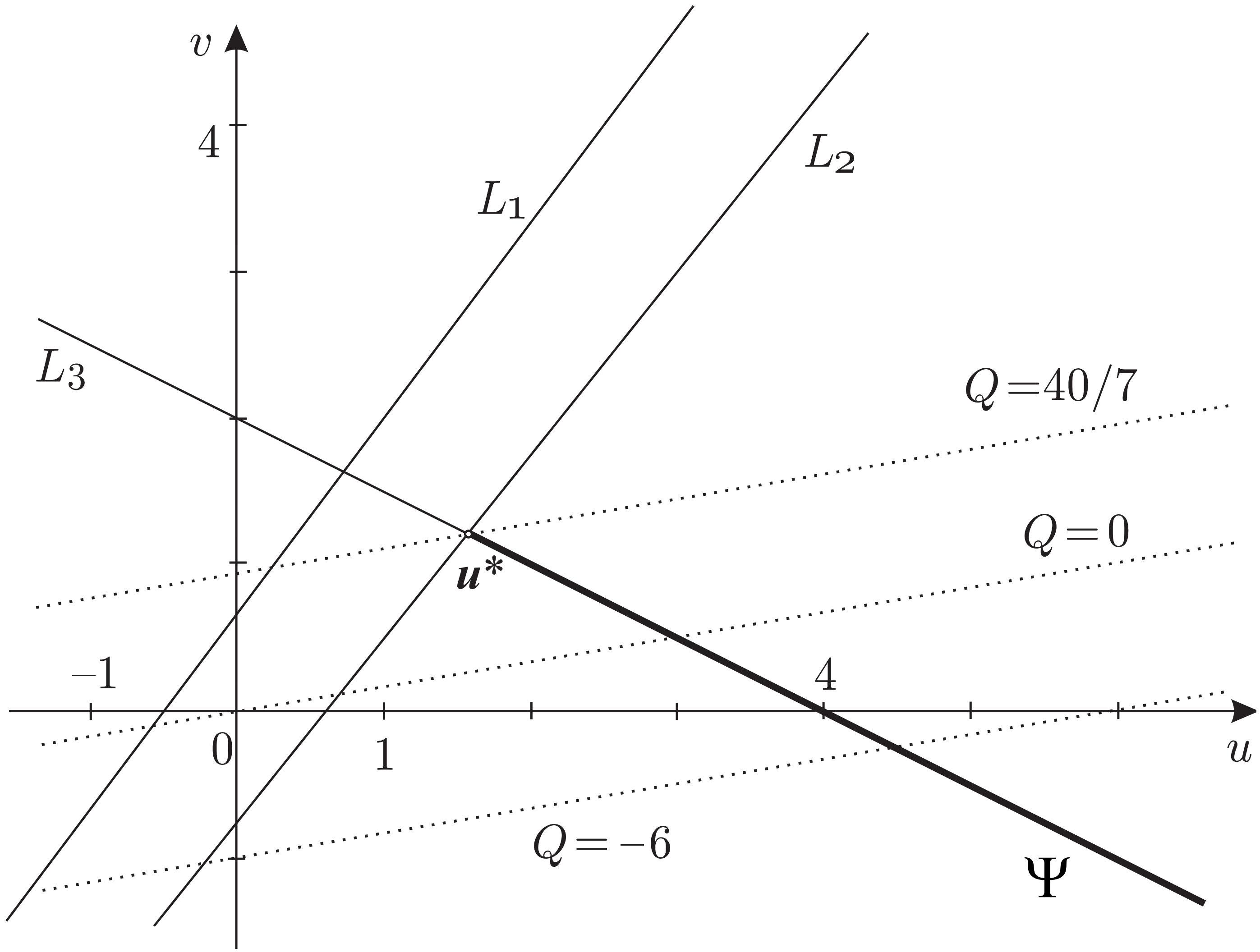
Prirejena naloga $LP^* = (\Psi, Q, \mathrm{Max})$ je določena z

$$\Psi = \{(u, v) \in \mathbb{R}^2 : -4u + 3v \leq 2,\ -5u + 4v \leq -3,\ u + 2v = 4,\ u \geq 0\}$$

$$Q(u, v) = -u + 6v$$

To nalogo lahko rešimo tudi grafično (glej sliko).

Zgled – Slika



. . . Dualnost – Zgled

Rešitev naloge je presečišče premice prve in tretje omejitve:

$$-5u + 4v = -3 \quad \text{in} \quad u + 2v = 4$$

Od koder dobimo

$$u^* = \frac{11}{7}, \quad v^* = \frac{17}{14} \quad \text{in} \quad Q^* = \frac{40}{7}$$

Določimo še rešitev osnovne naloge. Iz $(\mathbf{c} - \mathbf{u}^* A)\mathbf{x}^* = 0$ izhaja $x^* = 0$ in y'^*, z^* sta rešitvi sistema enačb

$$-5y' + z = -1 \quad \text{in} \quad 4y' + 2z = 6$$

Torej je rešitev osnovne naloge enaka

$$\mathbf{x}^* = (0, -\frac{4}{7}, \frac{13}{7})$$

. . . Dualnost

Izkaže se tudi: Če je naloga LP v osnovni obliki je njej prirejena naloga v kanonski obliki.

Prehode med obema nalogama opisuje tabela

	Min		Max	
	≥ 0	$\leftrightarrow$	$\leq$	
sprem.	≤ 0	$\leftrightarrow$	$\geq$	omejitve
	$\in \mathbb{R}$	$\leftrightarrow$	$=$	
	$\geq$	$\leftrightarrow$	≥ 0	
omejitve	$\leq$	$\leftrightarrow$	≤ 0	sprem.
	$=$	$\leftrightarrow$	$\in \mathbb{R}$	

Opombe

Obstajajo naloge linearnega programiranja (Klee-Minty), ki za izbrano izvedbo simpleksnega postopka zahtevajo pregled eksponentno veliko oglišč, preden postopek dospe do optimalne rešitve.

Zato je pomenil velik napredek *elipsoidni algoritem*, ki ga je leta 1979 objavil ruski matematik Kačjan (Khachian) in je zanj pokazal, da vselej zahteva le polinomsko število korakov. Za Kačjanov algoritem ni bilo učinkovite izvedbe, ki bi se lahko pri vsakodnevnih nalogah kosala s simpleksnim algoritmom.

Stvari so se spremenile leta 1984, ko je Narendra Karmakar, matematik iz AT&T Bell Laboratories, objavil drug polinomski algoritem. Tega so pri AT&T vgradili v sistem KORBX, ki omogoča v sprejemljivem času (ure) reševati tudi naloge obsežnosti čez 100000. Kasneje še boljši: npr. Bertsekas-ov RELAXT-3. Programi: CPLEX, LOQO, Mosek, OSL, PCx.

Linearno programiranje je tudi "vezna" metoda med zvezno in diskretno optimizacijo.