Lineārā programmēšana III

1. Simpleksalgoritma apraksts

Iepriekšējā lekcijā tika nodemonstrēta simpleksalgoritma darbība uz piemēriem. Šajā lekcijā tiks dots šī algoritma formāls apraksts.

Tiks pieņemts, ka lineārā programma ir pārveidota uz šādu standartformu:

Maksimizēt $c_1 x_1 + c_2 x_2 + ... + c_n x_n$
pie šādiem nosacījumiem:
$a_{11} x_1 + a_{12} x_2 + ... + a_{1n} x_n = b_1$,
...
$a_{k1} x_1 + a_{k2} x_2 + ... + a_{kn} x_n = b_k$,
$x_1 \geq 0, x_2 \geq 0, ..., x_n \geq 0$.

Šo standartformu var aprakstīt ar šādu tabulu:

a_{11}	a_{12}	...	a_{1n}	b_1
...	...	...	...	...
a_{k1}	a_{k2}	...	a_{kn}	b_k
c_1	c_2	...	c_n	

Simpleksa algoritms darbojas, pārvietojoties no viena pieļaujamā apgabala stūra uz citu stūri, līdz maksimizējamās funkcijas vērtību vairs nav iespējams palielināt. Ja pieļaujamā apgabala punkts ir stūris, tad vismaz n-k no mainīgajiem $x_1, x_2, ..., x_n$ ir vienādi ar 0. (n dimensiju telpā stūris ir punkts, kur krustojas n no plaknēm, kas ierobežo pieļaujamo apgabalu. n-k no tām ir jābūt $x_i = 0$ plaknēm, jo mums ir tikai k cita veida plakņu.)

Ja mums ir stūris, kurā mainīgie $x_1, x_2, ..., x_{n-k}$ ir vienādi ar 0, tad (kā tika izstāstīts iepriekšējā lekcijā) mēs varam tabulu aizstāt ar ekvivalentu tabulu, kas izskatās šādi:

a_{11}	...	$a_{1, n-k}$	1	0	...	0	b_1
a_{21}	...	$a_{2, n-k}$	0	1	...	0	b_2
...	...	...	...	...	...	...	...
a_{k1}	...	$a_{k, n-k}$	0	0	...	1	b_k
c_1	...	c_{n-k}	0	0	...	0	

Simpleksalgoritma soļi tagad ir šādi:

1. Lai pārbaudītu, vai kādā blakus stūrī nav lielāka mērķfunkcijas vērtība, mēs pārbaudām, vai kāds no $c_1, c_2, ..., c_{n-k}$ nav lielāks par 0. Ja $c_1, c_2, ..., c_{n-k}$ visi ir mazāki vai vienādi par 0, tad sasniegts maksimums.
2. Citādi mēs izvēlamies c_i, kas lielāks par 0.
3. Priekš katra j = 1, ..., k, kuram $a_{ji} \neq 0$, aprēķinam $d_j = b_j / a_{ji}$. Ja visi d_j ir negatīvi, tad mērķfunkcijas vērtība var būt neierobežoti liela un maksimums tai neeksistē.

4. Citādi atrodam j, kuram d_j ir vismazākais starp visiem pozitīvajiem d_j.
5. Apmaina tabulā i-to kolonnu vietām ar (n - k + j) - to kolonnu, iegūstot tabulu, kas izskatās šādi:

a_{11}	...	0	...	$a_{1,\,n-k}$	1	...	$a_{1,\,i}$	...	0	b_1
...	...	...	...	...	...	...	...	...	...	...
a_{j1}	...	1	...	$a_{j,\,n-k}$	0	...	$a_{j,\,i}$	...	0	b_j
...	...	...	...	...	...	...	...	...	...	...
a_{k1}	...	0	...	$a_{k,\,n-k}$	0	...	$a_{k,\,i}$	...	1	b_k
c_1	...	0	...	c_{n-k}	0	...	c_i	...	0	

6. Aizstāj šo tabulu ar ekvivalentu tabulu iepriekšējā standartformā:
   a. Izdala visus skaitļus j-tajā rindā ar $a_{j,\,i}$, lai šīs rindas krustojumā ar (n-k+j)-to kolonnu būtu 1.
   b. Priekš katra l, $l \neq j$, atņem no l-tās rindas, j-to rindu, kas pareizināta ar $a_{l,\,i}$. Tādejādi tiek panākts, ka (n-k+j)-tās kolonnas l-tajā rūtiņā ir 0.
   c. No pēdējās rindas atņem j-to rindu, kas pareizinātu ar c_i, lai pēdējā rindā (n-k+j)-ajā rūtiņā būtu 0.
7. Atkārto soļus 1-6, līdz tiek sasniegts maksimums vai noskaidrots, ka mērķfunkcija nav ierobežota.

2. Duālā lineārā programma

2.1. Duālās LP definīcija

Pieņemsim, ka mums ir lineāra programma (kuru mēs tālāk sauksim par primāro LP):

Maksimizēt $c_1 x_1 + c_2 x_2 + ... + c_n x_n$
pie šādiem nosacījumiem:
$a_{11} x_1 + a_{12} x_2 + ... + a_{1n} x_n \geq b_1$,
...
$a_{k1} x_1 + a_{k2} x_2 + ... + a_{kn} x_n = b_k$,
$x_1 \geq 0$, $x_2 \geq 0$, ...

Šoreiz, šī programma var nebūt standartformā:

- Nosacījumiem $a_{i1} x_1 + a_{i2} x_2 + ... + a_{in} x_n$? b_i, jautājuma zīmes vietā var būt jebkura no $\geq$, $\leq$, = zīmēm.
- Attiecībā uz mainīgajiem x_i, mums var būt nosacījumi $x_i \geq 0$, $x_i \leq 0$, vai vispār nebūt nosacījuma attiecībā uz x_i.

Tad duālā programma ir lineārā programma, kurā jāminimizē izteiksme

$$b_1 y_1 + b_2 y_2 + ... + b_k y_k,$$

pie nosacījumiem

$$a_{11} y_1 + a_{21} y_2 + ... + a_{k1} y_k \; ? \; c_k,$$

kur simbols jautājuma zīmes vietā tiek noteikts šādi:

- Ja primārajā LP bija nosacījums $x_i \geq 0$, tad jautājuma zīmes vietā ir $\geq$.
- Ja primārajā LP bija nosacījums $x_i \leq 0$, tad jautājuma zīmes vietā ir $\leq$.
- Ja primārajā LP nebija nosacījuma attiecībā uz x_i, tad jautājuma zīmes vietā ir =.

Attiecībā uz mainīgajiem $y_1, y_2, ..., y_k$, nosacījumi ir atkarīgi no tā, kāda zīme bija primārās LP nosacījumā $a_{i1} x_1 + a_{i2} x_2 + ... + a_{in} x_n$? b_i:

- Ja ? vietā bija $\geq$, tad mums tagad ir nosacījums $y_i \leq 0$.
- Ja ? vietā bija $\leq$, tad mums tagad ir nosacījums $y_i \geq 0$.
- Ja ? vietā bija =, tad mums tagad nav nosacījuma attiecībā uz y_i.

2.2 Piemērs un duālās LP interpretācija

Ja primārā lineārā programma ir

Maksimizēt $5x_1 + 16x_2$ pie nosacījumiem
$x_1 + x_2 \leq 1$,
$2 x_1 + 7 x_2 \leq 9$,
$x_1 \geq 0, x_2 \geq 0$,

tad duālā programma ir:

Minimizēt $y_1 + 9y_2$ pie nosacījumiem
$y_1 + 2 y_2 \geq 5$,
$y_1 + 7 y_2 \geq 16$,
$y_1 \geq 0, y_2 \geq 0$.

Duālo programmu var interpretēt šādi: katrs duālās programmas atrisinājums dod novērtējumu no augšas priekš primārās programmas atrisinājuma. Piemēram, ja mums ir duālās programmas atrisinājums $y_1 = y_2 = 2$, tad no duālās programmas nosacījumiem seko, ka

$$5x_1 + 16x_2 \leq 2 (x_1 + x_2) + 2 (2 x_1 + 7 x_2).$$

Apvienojot to ar primārās programmas nosacījumiem, mēs iegūstam, ka

$$5x_1 + 16x_2 \leq 2 \cdot 1 + 2 \cdot 9 = 20,$$

tas ir, primārās LP mērķfunkcija jebkurā punktā ir mazāka par duālās programmas mērķfunkciju (arī jebkurā punktā, jo augstāk minētajā spriedumā $y_1 = y_2 = 2$ var aizstāt ar jebkuru citu punkti, kur izpildās visi duālās programmas nosacījumi).

Dualitātes teorēma.

1. Ja primārajai LP eksistē maksimums, tad duālajai LP arī eksistē atrisinājums un primārās LP maksimums sakrīt ar duālās LP minimumu.

2. Ja primārajai LP neeksistē atrisinājums (nosacījumi ir pretrunīgi), tad duālajai LP mērķfunkcija var sasniegt patvaļīgi mazas vērtības.
3. Ja primārajai LP mērķfunkcija var sasniegt patvaļīgi lielas vērtības, tad duālajai LP atrisinājums neeksistē (nosacījumi ir pretrunīgi).

2.3 Primārās un duālās LP apvienošana

Ja mums ir primārā LP un duālā LP, mēs varam uzrakstīt jaunu LP, kas satur visus mainīgos (gan $x_1, x_2, ..., x_n$, gan $y_1, y_2, ..., y_k$), gan visus nosacījumus no abām programmām un pievienot tai vēl vienu nosacījumu:

$$c_1 x_1 + c_2 x_2 + ... + c_n x_n = b_1 y_1 + b_2 y_2 + ... + b_k y_k.$$

Tad vienīgais gadījums, kad izpildās visi nosacījumi ir, ja $x_1, x_2, ..., x_n$ sasniedz primārās LP maksimums, bet $y_1, y_2, ..., y_k$ sasniedz duālās LP minimumu.

Secinājums: Ja mums ir algoritms, kas prot patvaļīgai LP atrast vienu punktu, kas apmierina visus nosacījumus, tad šo algoritmu var izmantot arī maksimuma atrašanai.

3. Elipsoīda algoritms

Šo algoritmu izgudroja Hačijans (Khachiyan) 1979. gadā. Elipsoīda algoritms ir slavens kā pirmais lineārās programmēšanas algoritms, kuram tika pierādīts, ka tas atrod atrisinājumu polinomiālā laikā ($O(n^4L)$, kur n- dimensiju skaits, L – ar cik bitu precizitāti jāatrod atrisinājums).

Lai gan teorētiski darbības laiks ir polinomiāls, praksē algoritms ir lēns un netiek lietots. Tāpēc šajā kursā mēs ierobežosimies ar īsu šī algoritma aprakstu.

2.3 nodaļā aprakstītās redukcijas dēļ mums pietiek ar algoritmu, kas atrod punktu, kur izpildās visi nosacījumi. To dara šādi:

1. Izrēķinam elipsoīdu E_0, kas noteikti ietver LP pieļaujamo apgabalu.
2. Līdz tiek sasniegta vajadzīgā precizitāte:
   a. Ņem iepriekšējā elipsoīda E_i centru c_i.
   b. Ja c_i neapmierina visus LP nosacījumus, tad atrod nosacījumu a_k, kas tiek pārkāpts visvairāk.
   c. Ar plakni, kas sastāv no visiem punktiem, kur nosacījuma a_k izteiksmei ir vienāda vērtība c (kur c ir pa vidu starp vērtību punktā c_i un pieļaujamajām izteiksmes vērtībām) pārdala telpu divās daļās. Ar R_1 apzīmējam daļu, kur nonāk c_i un ar R_2 apzīmējam daļu, kur nonāk pieļaujamais apgabals.
   d. Uzkonstruē jaunu elipsoīdu E_{i+1}, tā lai izpildītos
   E_i 🖫 R_2 **U** E_{i+1}.

Var pierādīt, ka E_{i+1} var konstruēt tā, lai