Linearno programiranje (LP)

Zapisali smo ***linearni program (LP) v standardni obliki***:

Podatki: realna matrika A velikosti m x n, realni vektor b dolžine m, realni vektor c dolžine n

Iščemo: max $c^T x$
pri pogojih: $A x <= b, x >= 0$

Tu je seveda x realni vektor dolžine n.

Če je $n <= 2$, lahko LP rešimo grafično. Naredili smo tri primere, od katerih je eden imel natanko eno optimalno rešitev, eden je imel neskončno optimalnih rešitev, eden pa je bil neomejen.

2.1 Primeri problemov, ki jih lahko zapišemo kot LP

Ogledali smo si nekaj problemov, ki jih lahko zapišemo kot linearne programe:

1. Proizvodni problem
2. Prehrambeni problem
3. Načrtovanje dela

Na primeru smo si ogledali reševanje LP s **simpleksno metodo**.

V ***osnovni varianti simpleksne metode*** predpostavimo, da je vektor b nenegativen. S pomočjo *dopolnilnih spremenljivk* sistem neenačb spremenimo v enačbe in jih zapišemo v obliki *slovarja*, kjer vsako *bazno spremenljivko* (pa tudi ciljno funkcijo oziroma *funkcional*) izrazimo z *nebaznimi spremenljivkami*. Slovar določa *bazno dopustno rešitev*, v kateri imajo vse nebazne spremenljivke vrednost 0, vrednosti baznih spremenljivk pa lahko preberemo iz slovarja. Na začetku za bazne spremenljivke vzamemo dopolnilne spremenljivke, prvotne spremenljivke pa so nebazne.

Na vsakem koraku potem skušamo povečati vrednost ciljne funkcije s spremembo slovarja:
- ena od nebaznih spremenljivk vstopi v bazo,
- ena od baznih spremenljivk zapusti bazo.

Za *vstopajočo spremenljivko* lahko izberemo katerokoli spremenljivko, ki ima v ciljni funkciji pozitiven koeficient. (Pogosto vzamemo tisto z največjim koeficientom.) Vrednosti vstopajoče spremenljivke običajno ne moremo poljubno povečati, saj nas pri tem omejujejo pogoji nenegativnosti baznih spremenljivk. Za *izstopajočo spremenljivko* vzamemo tisto, ki to povečanje najmočneje omejuje. Enačbo, ki v starem slovarju izraža izstopajočo spremenljivko, imenujemo *pivotna vrstica*. Iz nje izrazimo vstopajočo spremenljivko, ki jo nato v vseh drugih vrsticah slovarja in tudi v ciljni funkciji nadomestimo z dobljenim izrazom. Tako smo dobili nov slovar, ki mu ustreza nova bazna dopustna rešitev.

To ponavljamo, dokler v ciljni funkciji nobena spremenljivka nima več pozitivnega koeficienta. Bazna dopustna rešitev, ki ustreza zadnjemu slovarju, je optimalna.

Ugotovili smo, da so pri reševanju LP po simpleksni metodi vsi slovarji med sabo ekvivalentni.

Pri reševanju linearnega programa s simpleksno metodo lahko nastopijo trije problemi, ki jih še nismo razrešili:

a) *izbira začetne bazne dopustne rešitve* (kaj če desna stran začetnega sistema neenačb vsebuje tudi negativne komponente?),
b) *izbira izstopajoče spremenljivke* (kaj če nobena od baznih spremenljivk ne omejuje povečanja vrednosti vstopajoče spremenljivke?),
c) *končnost postopka* (kaj če se postopek nikoli ne konča?).

b) Neomejenost rešitve. Če ne moremo izbrati izstopajoče spremenljivke, ker nobena bazna spremenljivka ne postavlja zgornjih mej za vrednost vstopajoče spremenljivke, lahko le-ta zavzame poljubno nenegativno vrednost, kar pomeni, da ima LP dopustne rešitve s poljubno veliko vrednostjo. Tak LP je **neomejen**.

c) Končnost postopka. Spoznali smo *izrojene* ali *degenerirane* bazne dopustne rešitve, pri katerih imajo tudi nekatere bazne spremenljivke vrednost nič. Kadar pivotna vrstica pripada bazni spremenljivki z vrednostjo 0, se vrednost funkcionala ne poveča. Obstajajo primeri podatkov, pri katerih se simpleksna metoda ne konča. Pokazali smo, da se to lahko zgodi le, če pride do cikličnega ponavljanja slovarjev. To pa lahko preprečimo z ***Blandovim pravilom*** ali ***pravilom najmanjšega indeksa***, ki pravi, da se v primeru, ko imamo pri izbiri vstopajoče in/ali izstopajoče spremenljivke na razpolago več možnosti, vedno odločimo za spremenljivko z najmanjšim indeksom.

Linearni program v splošni obliki smo definirali kot optimizacijsko nalogo, pri kateri iščemo ekstrem linearnega funkcionala pri linearnih pogojih, ki so lahko neenačbe (<= ali >=) ali enačbe, in pokazali, kako ga lahko prevedemo na enakovreden program v standardni obliki.

a) Začetna bazna dopustna rešitev. Če so na desni strani začetnih neenačb tudi negativna števila, običajna začetna bazna rešitev, kjer za bazo vzamemo vse dopolnilne spremenljivke, ni dopustna. Zato v takem primeru uporabimo ***dvofazno simpleksno metodo***, kjer v I. fazi poiščemo začetni dopustni slovar, v II. fazi pa potem nadaljujemo z običajnim simpleksnim postopkom. V I. fazi rešujemo pomožni LP, ki ga dobimo iz prvotnega tako, da desnim stranem neenačb prištejemo *umetno spremenljivko* x_0 in sestavimo slovar, za funkcional pa vzamemo $w = -x_0$. Pokazali smo, da ima prvotni LP dopustno rešitev natanko tedaj, ko je optimalna vrednost pomožnega LP enaka 0.

Če se pri reševanju I. faze poleg običajne simpleksne metode držimo še dodatnih ***pravil o ravnanju v I. fazi*** (*1. na začetku v bazo vključimo spremenljivko x_0, iz nje pa odstranimo spremenljivko z najmanjšim prostim členom na desni; 2. umetno spremenljivko odstranimo iz baze, brž ko je mogoče*; *3. brž ko je vrednost trenutne bazne dopustne rešitve enaka 0, končamo*), na koncu I. faze nastopita le dve možnosti:

1. opt. vrednost I. faze je različna od 0: v tem primeru je prvotni LP nedopusten,
2. opt. vrednost I. faze je enaka 0, umetna spremenljivka x_0 pa ni v bazi: v slovarju vse člene, ki vsebujejo spremenljivko x_0, izpustimo, funkcional pod črto pa nadomestimo s prvotnim funkcionalom $z = c^T x$, ki ga s pomočjo slovarja izrazimo s samimi nebaznimi spremenljivkami. Dobljeni slovar nam da začetno bazno dopustno rešitev za prvotni problem in predstavlja začetni slovar II. faze.

V II. fazi z običajno simpleksno metodo poiščemo optimalno rešitev prvotnega LP (ali pa ugotovimo, da je le-ta neomejen).

Podali smo povzetek dvofazne metode in ugotovili, da iz nje izhaja ***osnovni izrek linearnega programiranja***, ki pravi: *Za LP v standardni obliki veljajo naslednje trditve:*

1. Je bodisi nedopusten bodisi neomejen bodisi ima optimalno rešitev.
2. Če ima dopustno rešitev, ima tudi bazno dopustno rešitev.
3. Če ima optimalno rešitev, ima tudi bazno optimalno rešitev.

Na vajah ste na primeru pogledali tudi, kako poiščemo vse optimalne rešitve linearnega programa.

Obravnavali smo časovno zahtevnost simpleksnega algoritma (*n* neznank, *m* neenačb):

- vsak korak metode je polinomski, reda velikosti $m*n$.
- število korakov v splošnem ni polinomsko (za večino standardnih pravil za izbiro vstopajoče spremenljivke so znani primeri, kjer metoda simpleksov porabi eksponentno število korakov)
- v praksi (za "tipične" podatke, ki nastopijo pri realnih problemih) je metoda hitra, število korakov je (zelo približno) reda velikosti $m \log n$.

Omenili smo še dve metodi za reševanje linearnih programov

- elipsoidna metoda, Khachyan, 1979 (polinomska časovna zahtevnost za LP, a v praksi prepočasna)
- metode notranjih točk, Karmakar, 1984 (praktično uporabne, v polinomskem času poiščejo poljubno natančen približek za optimalno rešitev LP).

2.3. Dualnost v linearnem programiranju

Za linearni program P v standardni obliki:

iščemo: max $c^T x$
pri pogojih: $A x \le b$, $x \ge 0$

smo definirali njegov ***dualni program*** P'

iščemo: min $b^T y$
pri pogojih: $A^T y \ge c$, $y \ge 0$.

Če dualni program P' prepišemo v standardni obliki, je njegov dual (P')' enakovreden prvotnemu programu P.

Šibki izrek o dualnosti: *Vrednost dopustne rešitve prvotnega linearnega programa P je kvečjemu manjša od vrednosti katerekoli dopustne rešitve dualnega linearnega programa P'.*

Krepki izrek o dualnosti: *Če ima prvotni linearni program P optimalno rešitev, jo ima tudi dualni linearni program P', njuni vrednosti pa sta enaki.*

Pokazali smo, da od devetih možnih kombinacij lastnosti (je nedopusten, je neomejen, ima optimalno rešitev) prvotnega programa *P* in dualnega programa *P'* v resnici lahko nastopijo le štiri:

1. *P* in *P'* sta nedopustna,
2. *P* je nedopusten, *P'* pa neomejen,
3. *P* je neomejen, *P'* pa nedopusten,
4. *P* in *P'* imata optimalno rešitev.

Dokazali smo izrek o krepki dualnisti. Iz dokaza se vidi, da lahko optimalno rešitev dualnega programa preprosto preberemo iz koeficientov zadnjega funkcionala pri reševanju prvotnega programa z dvofazno simpleksno metodo.

Ogledali smo si uporabe dualnega programa oziroma dualne rešitve:

1. za pospešitev reševanja LP v primeru, ko je število pogojev večje od števila neznank,
2. za dokazovanje optimalnosti rešitve,
3. analiza rešitve (za koliko se spremeni optimalna vrednost pri majhnih spremembah omejitev).

Dokazali smo ***izrek o dualnem dopolnjevanju***, ki karakterizira optimalnost para dopustnih rešitev x oziroma y prvotnega oziroma dualnega programa s pomočjo sistema linearnih enačb, ki mu morata zadoščati x in y.

Vprašanje: Ali se podjetju splača dokupiti i-to surovino?

Odgovor: DA, če velja:
- prvotni program ima vsaj eno neizrojeno optimalno bazno rešitev,
- $y_i^* > r_i$, kjer je y_* dualna optimalna rešitev, r_i pa razlika med tržno ceno in vkalkulirano ceno enote i-te surovine,
- dokupljena količina i-te surovine je dovolj majhna, da se y_i^* še ne spremeni.