

# Linearno programiranje (LP)

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

*Podatki*: realna matrika  $A$  velikosti  $m \times n$ , realni vektor  $b$  dolžine  $m$ , realni vektor  $c$  dolžine  $n$

*Iščemo*:  $\max \langle c, x \rangle$

pri pogojih:  $Ax \leq b, x \geq 0$

Tu je seveda  $x$  realni vektor dolžine  $n$ .

Če je  $n \leq 2$ , lahko LP rešimo grafično.

Na zgledu proizvodnega problema kmeta (kmet se odloča, koliko hektarov zemlje bo namenil posameznim poljščinam, da bo njegov čisti dobiček največji) smo si ogledali reševanje LP z **metodo simpleksov** ali **simpleksno metodo**.

V osnovni varianti 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 namensko funkcijo oziroma *funktional*) 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. 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 najbolj 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 funkcionalu 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.

Ogledali smo si tudi, kako s pomočjo zadnjega slovarja pri simpleksni metodi poiščemo vse optimalne rešitve linearnega programa.

Spoznali smo *izrojene* bazne dopustne rešitve, pri katerih imajo tudi nekatere bazne spremenljivke vrednost nič. Ugotovili smo, da se lahko pri reševanju linearnega programa s simpleksno metodo pojavijo naslednja vprašanja:

a) *končnost postopka* (ali se lahko primeri, da se postopek reševanja po simpleksni metodi nikoli ne konča?),

b) *izbira izstopajoče spremenljivke* (kaj če nobena od baznih spremenljivk ne omejuje

povečanja vrednosti vstopajoče spremenljivke?),

c) *izbira začetne bazne dopustne rešitve* (kako poiskati začetno bazno dopustno rešitev, če desna stran začetnega sistema neenačb vsebuje tudi negativne komponente?).

Oglejmo si odgovore na ta tri vprašanja.

a) **Končnost postopka.** Simpleksni postopek lahko teče v neskončnost, 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, vselej odločimo za spremenljivko z najmanjšim indeksom.

b) **Izbira izstopajoče spremenljivke.** Če ne moremo izbrati izstopajoče spremenljivke, ker nobena bazna spremenljivka ne omejuje povečanja vrednosti vstopajoče spremenljivke, lahko nastavimo njeno vrednost poljubno visoko. Od tod sledi, da obstajajo dopustne rešitve s poljubno veliko vrednostjo, saj ima vstopajoča spremenljivka v funkcionalu pozitiven koeficient. Ker iščemo maksimum ciljne funkcije, je LP v tem primeru **neomejen**.

c) **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  $-x_0$ . Na 1. koraku I. faze v bazo vključimo spremenljivko  $x_0$ , iz nje pa odstranimo spremenljivko z najmanjšo vrednostjo. Nato nadaljujemo po običajni simpleksni metodi. Pokazali smo, da ima prvotni LP dopustno rešitev natanko tedaj, ko je optimalna vrednost pomožnega LP enaka 0.

**Dogovor o ravnanju v I. fazi:**

1. Umetno spremenljivko  $x_0$  odstranimo iz baze, brž ko je to mogoče.
2. Če vrednost funkcionala postane enaka 0, končamo.

Če se držimo tega dogovora, ob koncu I. faze lahko nastopi le ena od naslednjih dveh možnosti:

a) *Optimalna vrednost I. faze je različna od 0:* V tem primeru je prvotni LP **nedopusten**.

b) *Optimalna vrednost I. faze je enaka 0 in umetna spremenljivka  $x_0$  ni v bazi:* Začnemo z II. fazo reševanja. Vse člene s spremenljivko  $x_0$  odstranimo iz slovarja in funkcional nadomestimo s prvotnim funkcionalom, v katerem vse bazne spremenljivke s pomočjo slovarja izrazimo z nebaznimi. Tako dobimo začetni dopustni slovar za prvotni LP. Nadaljujemo po običajni simpleksni metodi.

**Linearni program v splošni obliki** smo definirali kot optimizacijsko nalogo, pri kateri iščemo ekstrem linearnega funkcionala pri linearnih pogojih, ki so lahko neenačbe ( $\leq$  ali  $\geq$ ) ali enačbe, in pokazali, kako ga lahko prevedemo na ekvivalenten program v standardni obliki.

Podali smo povzetek dvofazne metode in ugotovili, da iz nje izhaja **osnovni izrek linearnega programiranja**, ki pravi:

1. če ima LP dopustno rešitev, ima tudi bazno dopustno rešitev,

2. če ima LP optimalno rešitev, ima tudi bazno optimalno rešitev,
3. LP je bodisi nedopusten bodisi neomejen bodisi ima optimalno rešitev.

Pri linearnem programu se torej ne more zgoditi, da bi bil dopusten in omejen, pa ne bi imel optimalne rešitve.

## Dualnost v linearnem programiranju

Linearnemu programu  $\Pi$  v standardni obliki, kjer

$$\text{iščemo: } \max \langle c, x \rangle$$

$$\text{pri pogojih: } Ax \leq b, x \geq 0,$$

smo priredili **dualni linearni program**  $\Pi'$ , kjer

$$\text{iščemo: } \min \langle b, y \rangle$$

$$\text{pri pogojih: } Ay \geq c, y \geq 0,$$

in pokazali, da je dual dualnega programa enakovreden prvotnemu programu.

Dokazali smo **šibki izrek o dualnosti**: Za vsako dopustno rešitev  $x$  programa  $\Pi$  in vsako dopustno rešitev  $y$  dualnega programa  $\Pi'$  je  $\langle c, x \rangle \leq \langle b, y \rangle$ .

**Posledica**: Če je  $x$  dopustna rešitev programa  $\Pi$ ,  $y$  dopustna rešitev dualnega programa  $\Pi'$  in je  $\langle c, x \rangle = \langle b, y \rangle$ , potem je  $x$  optimalna rešitev programa  $\Pi$  in  $y$  optimalna rešitev programa  $\Pi'$ .

Dokazali smo **kreпки izrek o dualnosti**: Če ima prvotni program optimalno rešitev, jo ima tudi dualni program, njuni vrednosti pa sta enaki. Pri dokazu smo ugotovili, da lahko optimalno rešitev dualnega programa preprosto preberemo iz koeficientov zadnjega funkcionala pri reševanju prvotnega programa z dvofazno simpleksno metodo.

Pokazali smo, da od devetih možnih kombinacij lastnosti (je nedopusten, je neomejen, ima optimalno rešitev) prvotnega in dualnega linearnega programa v resnici lahko nastopijo le štiri: bodisi sta oba programa nedopustna bodisi je eden od njiju nedopusten, drugi pa neomejen, bodisi ima vsak od njiju optimalno rešitev.

Omenili smo nekaj uporab dualnega linearnega programa oziroma dualne rešitve:

1. za dokazovanje optimalnosti rešitve,
2. za pospešitev reševanja LP v primeru, ko je število neenačb večje od števila spremenljivk,
3. za analizo vpliva sprememb desne strani na optimalno vrednost LP (ekonomski pomen dualnih spremenljivk).

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  $y$ .

Zapisali smo **dual LP** v splošni obliki.