Linearno programiranje

Uvod

Mnogi problemi mogu biti formulisani kao maksimizacija ili minimizacija neke ciljne funkcije za date ograničene resurse i uzajamna ograničenja. Ako možemo definisati ciljnu funkciju kao linearnu funkciju određenih promenljivih i ako možemo zadati ograničenja u resursima kao jednakosti ili nejednakosti ovih promenljivih, onda dobijamo **problem linearnog programiranja**. Linearni programi se javljaju u raznim praktičnim primenama. Razmatranje linearnog programiranja ćemo započeti na primeru političkih izbora.

Politički problem

Pretpostavimo da ste Vi političar koji pokušava da pobedi na izborima. Vaš okrug ima tri različite vrste oblasti: gradske, prigradske i seoske. Ove oblasti imaju redom 100 000, 200 000 i 50 000 registrovanih glasača. Da bi ste ostvarili uspeh, želite da osvojite većinu glasova u svakoj od ove tri oblasti. Vi ste časni i nikada ne biste podržali politiku u koju ne verujete. Međutim, jasno Vam je da određena pitanja mogu imati veće efekte u nekim mestima. Vaši primarni ciljevi su da izgradite više puteva, kontrola naoružanja, subvencioniranje farmi i takse na gorivo namenjene poboljšanju javnog prevoza. U skladu sa istraživanjima Vašeg tima možete proceniti koliko možete glasova osvojiti ili izgubiti u svakoj populaciji ukoliko utrošite 1000 EUR na promociju svakog od pitanja. Ove informacije su date u Tabeli ###.

Tabela ###. Efekti politike na glasače.

Politika	Grad	Prigrad	Selo
izgradnja puteva	–2	5	3
kontrola naoružanja	8	2	–5
subvencionisanje farmi	0	0	10
takse na gorivo	10	0	–2

U ovoj tabeli svako polje označava broj glasova (u hiljadama) u gradskoj, prigradskoj ili seoskoj oblasti koji može biti osvojen ukoliko se na promociju određenog pitanja uloži 1000 EUR. Negativne vrednosti označavaju da će glasovi biti izgubljeni. Vaš zadatak je da odredite minimalnu količinu novca koju morate da potrošite da biste osvojili najmanje 50 000 gradskih, 100 000 prigradskih i 25 000 seoskih glasova.

Metodom probe i greške je moguće doći do strategije koja će doneti traženi broj glasova, ali ova strategija ne mora biti najjeftinija moguća. Na primer, možete nameniti 20 000 EUR za promociju izgradnje puteva, 0 EUR za kontrolu naoružanja, 4000 EUR za subvencionisanje farmi i 9000 EUR za takse na gorivo. U ovom slučaju ćete osvojiti 20(-2) + 0(8) + 4(0) + 9(10) = 50 hiljada gradskih glasova 20(5) + 0(2) + 4(0) + 9(0) = 100 hiljada prigradskih glasova i 20(3) + 0(–5) + 4(10) + 9(–2) = 82 hiljade seoskih glasova. Na ovaj način biste osvojili tačno željeni broj glasova u gradskoj i prigradskoj oblasti, ali i više nego što je potrebno u seoskoj oblasti (čak ste osvojili više glasova nego što ima glasača!). Za osvajanje ovih glasova biste utrošili 20 + 0 + 4 + 9 = 33 hiljade evra na reklamu.

Prirodno, možete se zapitati da li je ova strategija bila najbolja moguća. Da li ste mogli da ostvarite svoje ciljeve i da pri tome potrošite manje novca na reklamu. Dodatna analiza metodom probe i greške može pomoći u dobijanju odgovora, ali Vi želite sistematičnu metodu koja bi dala odgovor na ova pitanja. Da bismo to postigli formulisaćemo pitanje matematički. Uvedimo 4 promenljive:

- x_1 je broj hiljada evra potrošenih na promociju izgradnje puteva,
- x_2 je broj hiljada evra potrošenih na promociju kontrole naoružanja
- x_3 je broj hiljada evra potrošenih na promociju subvencija za farme
- x_4 je broj hiljada evra potrošenih na promociju taksi na gorivo

Zahtev da se osvoji najmanje 50 000 gradskih glasova možemo zapisati kao

$$-2x_1 + 8x_2 + 0x_3 + 10x_4 \geq 50 \tag{1}$$

Slično, zahtevi za osvajanjem najmanje 100 000 prigradskih glasova i 25 000 seoskih glasova mogu se zapisati kao

$$5x_1 + 2x_2 + 0x_3 + 0x_4 \geq 100 \tag{2}$$

i

$$3x_1 - 5x_2 + 10x_3 - 2x_4 \geq 25 \tag{3}$$

Bilo koji skup promenljivih x_1, x_2, x_3 i x_4 tako da budu zadovoljene nejednakosti (1)-(3) čini strategiju koja će osvojiti dovoljan broj glasova u svakoj od oblasti. Da bismo troškove sveli na minimum, želimo da minimizujemo količinu novca utrošenog na reklamiranje. To zapravo znači da želimo da minimizujemo izraz

$$x_1 + x_2 + x_3 + x_4 \tag{4}$$

Iako je negativna reklama čest slučaj u političkim kampanjama, negativni troškovi na reklamu nisu mogući. Iz tog razloga zahtevamo da

$$x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0 \text{ i } x_4 \geq 0 \tag{5}$$

Spajanjem nejednakosti (1)-(3) i (5) sa ciljem da se minimizuje (4), dobijamo ono što nazivamo **linearni program**. Ovaj problem zapisujemo kao

minimizovati

$$x_1 + x_2 + x_3 + x_4 \tag{6}$$

pri ograničenjima

$$-2x_1 + 8x_2 + 0x_3 + 10x_4 \geq 50 \tag{7}$$

$$5x_1 + 2x_2 + 0x_3 + 0x_4 \geq 100 \tag{8}$$

$$3x_1 - 5x_2 + 10x_3 - 2x_4 \geq 25 \tag{9}$$

$$x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0 \text{ i } x_4 \geq 0 \tag{10}$$

Rešenje ovog linearnog programa će dati optimalnu strategiju za političara.

Opšti linearni programi

U opštem slučaju linearnog programiranja želimo da optimizujemo linearnu funkciju ograničenu linearnim nejednakostima. Za dati skup realnih brojeva a_1, a_2,..., a_n i skup promenljivih x_1, x_2,..., x_n, **linearna funkcija** f nad ovim promenljivama je definisana kao

$$f(x_1, x_2, ..., x_n) = a_1x_1 + a_2x_2 + ... + a_nx_n = \sum_{j=1}^{n} a_j x_j$$

Ako je b realan broj i f linearna funkcija, onda je jednačina

$$f(x_1, x_2, ..., x_n) = b$$

linearna jednačina, a nejednakosti

$$f(x_1, x_2, ..., x_n) \leq b$$

i

$f\left(x_1, x_2, ..., x_n\right) \geq b$

linearne nejednakosti. Termin **linearna ograničenja** koristimo kao zajednički naziv za linearne jednačine i linearne nejednakosti. U linearnom programiranju nisu dozvoljene striktne nejednakosti. Formalno, **problem linearnog programiranja** je problem minimizacije ili maksimizacije linearne funkcije ograničene linearnim ograničenjima. Ako želimo da vršimo minimizaciju, onda linearni program nazivamo **minimizacionim linearnim programom**, a ukoliko želimo da vršimo maksimizaciju, onda linearni program nazivamo **maksimizacionim linearnim programom**.

U nastavku ovog poglavlja razmotrićemo neke od načina za predstavljanje problema linearnog programiranja. Nakon toga preoučićemo "simpleks" algoritam koji je i najstariji algoritam za rešavanje ove vrste problema.

Standardni i labav oblik

U ovoj sekciji su opisana dva oblika opisivanja problema linearnog programiranja: standardni i labav oblik. U standardnom obliku su sva ograničenja nejednakosti, dok su u labavom obliku ograničenja jednakosti.

Standardni oblik

U **standardnom obliku** je dato n realnih brojeva $c_1, c_2, ..., c_n$, m realnih brojeva $b_1, b_2, ..., b_m$ i mn realnih brojeva a_{ij} za $i = 1, 2, ..., m$ i $j = 1, 2, ..., n$. Želimo da odredimo n realnih brojeva $x_1, x_2, ..., x_n$ koji maksimizuju

$$\sum_{j=1}^{n} c_j x_j \tag{11}$$

pri ograničenjima

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad \text{za } i = 1, 2, ..., m \tag{12}$$

$$x_j \geq 0 \qquad \text{za } j = 1, 2, ..., n \tag{13}$$

Izraz (11) nazivamo **ciljnom funkcijom**, a $n + m$ nejednakosti u (12) i (13) **ograničenjima**. n ograničenja u (13) nazivamo **ograničenjima nenegativnosti**. Proizvoljan linearan program ne mora da ima ograničenja nenegativnosti, ali ih standardni oblik zahteva. Ponekad je pogodno predstaviti linearan program u kompaktnijem obliku. Ako formiramo $m \times n$ matricu $A = \left(a_{ij}\right)$, m-dimenzioni vektor $b = \left(b_i\right)$, n-dimenzioni vektor $c = \left(c_j\right)$ i n-dimenzioni vektor $x = \left(x_j\right)$, onda možemo zapisati linearni program definisan u (11)-(13) kao

maksimizovati

$$c^T x \tag{14}$$

pri ograničenjima

$$Ax \leq b \tag{15}$$

$$x \geq 0 \tag{16}$$

U izrazu (14) $c^T x$ je skalarni proizvod dva vektora. U izrazu (15) Ax je proizvod matrice i vektora, a u izrazu (16) $x \geq 0$ znači da svaki element vektora x mora biti nenegativan.

Uvedimo sada terminologiju za opisivanje rešenja linearnih programa. Skup promenljivih $\overline{x}$ koje zadovoljavaju zadata ograničenja nazivamo **mogućim rešenjem**, dok skup promenljivih $\overline{x}$ koji ne zadovoljava bar jedno ograničenje nazivamo **nemogućim rešenjem**. Za rešenje $\overline{x}$ kažemo da ima **ciljnu vrednost** $c^T \overline{x}$. Moguće rešenje $\overline{x}$ čija je ciljna vrednost maksimalna među svim mogućim rešenjima je

optimalno rešenje, a njegova ciljna vrednost $c^T \bar{x}$ je **optimalna ciljna vrednost**. Ako linearni program nema mogućih rešenja, onda kažemo da je linearan program **nemoguć**. U suprotnom kažemo da je **moguć**. Ukoliko linearan program ima neka moguća rešenja, ali nema konačnu optimalnu ciljnu vrednost, za njega kažemo da je **neograničen**.

Transformacija linearnih programa u standardni oblik

Linearni program je uvek moguće transformisati u standardni oblik, bez obzira da li se radi o minimizaciji ili maksimizaciji linearne funkcije. Linearni program može da ne bude u standardnom obliku zbog nekog od sledeća četiri razloga:

1. U pitanju je minimizacija ciljne funkcije, a ne maksimizacija.
2. Postoje promenljive koje nemaju ograničenja nenegativnosti.
3. Mogu postojati **ograničenja jednakosti**, koja imaju znak jednako umesto znaka manje ili jednako.
4. Mogu postojati **ograničenja nejednakosti**, koja umesto znaka manje ili jednako imaju znak veće ili jednako.

Kada transformišemo jedan linearni program L u drugi linearni program L', želimo da optimalno rešenje programa L' daje optimalno rešenje za program L. U nastavku ćemo pokazati kako se može rešiti svaki od četiri navedena problema. Prilikom otklanjanja svakog problema, novodobijeni linearni program mora da bude ekvivalentan originalnom programu.

Da bismo transformisali minimizacioni linearni program L u ekvivalentan maksimizacioni linearni program L', jednostavno menjamo znak koeficijentima u ciljnoj funkciji. Pošto L i L' imaju isti skup mogućih rešenja i za svako moguće rešenje ciljna vrednost od L je negativna ciljna vrednost od L', ova dva linearna programa su ekvivalentna. Na primer, ako imamo linearan program

minimizovati $-2x_1 + 3x_2$

pri ograničenjima

$$x_1 + x_2 = 7$$
$$x_1 - 2x_2 \le 4$$
$$x_1 \ge 0$$

i ukoliko promenimo znak koeficijentima ciljne funkcije, dobijamo

maksimizovati $2x_1 - 3x_2$

pri ograničenjima

$$x_1 + x_2 = 7$$
$$x_1 - 2x_2 \le 4$$
$$x_1 \ge 0$$

Sada ćemo pokazati kako da transformišemo linearan program u kome neke promenljive nemaju ograničenja nenegativnosti u program u kome svaka promenljiva ima ograničenje nenegativnosti. Pretpostavimo da promenljiva x_j nema ograničenje nenegativnosti. Onda svako pojavljivanje promenljive x_j zamenjujemo sa $x'_j - x''_j$ i dodajemo ograničenja nenegativnosti $x'_j \ge 0$ i $x''_j \ge 0$. Tako, ukoliko ciljna funkcija ima član $c_j x_j$, on se zamenjuje članom $c_j x'_j - c_j x''_j$, a ako i-to ograničenje sadrži član $a_{ij} x_j$, on se zamenjuje članom $a_{ij} x'_j - a_{ij} x''_j$. Svako moguće rešenje $\bar{x}$ novog linearnog programa odgovara rešenju originalnog linearnog programa sa $\bar{x}_j = \bar{x}'_j - \bar{x}''_j$ i istom ciljnom funkcijom, tako da su ova dva rešenja ekvivalentna. Ovu šemu primenjujemo na svaku promenljivu koja nema ograničenje nenegativnosti, kako bismo dobili ekvivalentan linearan program u kome sve promenljive imaju ograničenje nenegativnosti.

Nastavljajući dalje sa primerom, želimo da svaka promenljiva ima ograničenje nenegativnosti. Promenljiva x_1 ima ovakvo ograničenje, ali promenljiva x_2 nema. Zato promenljivu x_2 zamenjujemo sa $x_2' - x_2''$, tako da dobijamo linearni program

maksimizovati $2x_1 - 3x_2' + 3x_2''$

pri ograničenjima

$$x_1 + x_2' - x_2'' = 7$$
$$x_1 - 2x_2' + 2x_2'' \le 4$$
$$x_1, x_2', x_2'' \ge 0$$

Dalje, ograničenja jednakosti transformišemo u ograničenja nejednakosti. Pretpostavimo da linearni program ima ograničenje jednakosti $f(x_1, x_2, ..., x_n) = b$. Pošto je $x = y$ ako i samo ako je i $x \ge y$ i $x \le y$, ovu ograničenje jednakosti zamenjujemo parom ograničenja nejednakosti $f(x_1, x_2, ..., x_n) \le b$ i $f(x_1, x_2, ..., x_n) \ge b$. Ako ovu proceduru ponovimo za sva ograničenja jednakosti, dobićemo linearan program u kome su sva ograničenja nejednakosti.

Konačno možemo transformisati ograničenja "veće ili jednako" u ograničenja "manje ili jednako", množenjem ovih ograničenja sa -1. To znači da je svaka nejednakost oblika

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

jednaka

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

tako da jednostavnom zamenom svakog koeficijenta a_{ij} sa $-a_{ij}$ i svake vrednosti b_i sa $-b_i$, dobijamo ekvivalentno "manje ili jednako" ograničenje.

Završavajući naš primer, zamenjujemo ograničenja jednakosti sa dva ograničenja nejednakosti i dobijamo

maksimizovati $2x_1 - 3x_2' + 3x_2''$

pri ograničenjima

$$x_1 + x_2' - x_2'' \le 7$$
$$x_1 + x_2' - x_2'' \ge 7$$
$$x_1 - 2x_2' + 2x_2'' \le 4$$
$$x_1, x_2', x_2'' \ge 0$$

Konačno menjamo znak ograničenjima "veće ili jednako" i menjamo nazive promenljivih x_2' i x_2'' u x_2 i x_3, čime dobijamo standardni oblik

maksimizovati $2x_1 - 3x_2 + 3x_3$

pri ograničenjima

$$x_1 + x_2 - x_3 \le 7$$
$$-x_1 - x_2 + x_3 \le -7$$
$$x_1 - 2x_2 + 2x_3 \le 4$$
$$x_1, x_2, x_3 \ge 0$$

Transformacija linearnih programa u labav oblik

Da bi se efikasno rešio linearni program korišćenjem simpleks algoritma, linearni program je potrebno izraziti u takvom obliku u kome su neka od ograničenja ograničenja jednakosti. Preciznije rečeno, potrebno je transformisati program tako da jedina ograničenja nejednakosti budu ograničenja nenegativnosti, a sva ostala ograničenja budu ograničenja jednakosti. Neka su

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i \tag{17}$$

ograničenja nejednakosti. Uvešćemo novu promenljivu s i prepisati nejednakost (17) kao dva ograničenja

$$s = b_i - \sum_{j=1}^{n} a_{ij} x_j \tag{18}$$

$$s \ge 0 \tag{19}$$

Promenljivu s nazivamo **labavom promenljivom** zato što meri labavost, tj. razliku, između leve i desne strane nejednakosti (17). S obzirom da je nejednakost (17) tačna ako i samo ako su i jednačina (18) i nejednačina (19) tačne, ovu transformaciju možemo primeniti na bilo koje ograničenje nejednakosti linearnog programa i da pri tome dobijemo ekvivalentan linearni program u kome su jedina ograničenja nejednakosti zapravo ograničenja nenegativnosti. Kada transformišemo standardni oblik u labav oblik, koristićemo x_{n+i} umesto s, da bismo označili labavu promenljivu povezanu sa i-tom nejednakošću. Sada je i-to ograničenje

$$x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij} x_j \tag{20}$$

zajedno sa ograničenjem nenegativnosti $x_{n+i} \ge 0$.

Primenjujući ovu transformaciju na svako ograničenje linearnog programa u standardnom obliku, dobijamo linearni program u drugačijem obliku. Na primer, za linearni program opisan u tekstu o standardnom obliku uvodimo labave promenljive x_4, x_5 i x_6, tako da dobijamo

maksimizovati

$$2x_1 - 3x_2 + 3x_3 \tag{21}$$

pri ograničenjima

$$x_4 = 7 - x_1 - x_2 + x_3 \tag{22}$$

$$x_5 = -7 + x_1 + x_2 - x_3 \tag{23}$$

$$x_6 = 4 - x_1 + 2x_2 - 2x_3 \tag{24}$$

$$x_1, x_2, x_3, x_4, x_5, x_6 \ge 0 \tag{25}$$

U ovom linearnom programu su sva ograničenja ograničenja jednakosti osim ograničenja nenegativnosti. Svako ograničenje jednakosti je zapisano sa jednom promenljivom na levoj strani jednakosti, a sve ostale su na desnoj strani. Dalje, svaka jednakost ima isti skup promenljivih na desnoj strani i to su jedine promenljive koje se javljaju u ciljnoj funkciji. Promenljive na levoj strani jednakosti nazivamo **osnovnim promenljivama**, a promenljive na desnoj strani **sporednim promenljivama**.

Za linearne programe koji zadovoljavaju ove uslove, ponekad ćemo izostaviti reči "maksimizovati" i "pri ograničenjima" i izostaviti eksplicitno navođenje ograničenja nenegativnosti. Takođe ćemo koristiti promenljivu z da označimo vrednost ciljne funkcije. Dobijeni oblik nazivamo **labavi oblik**. Ukoliko linearni program dat u (21)-(25) napišemo u labavom obliku, dobijamo

$$z = 2x_1 - 3x_2 + 3x_3 \tag{26}$$

$$x_4 = 7 - x_1 - x_2 + x_3 \tag{27}$$

$$x_5 = -7 + x_1 + x_2 - x_3 \tag{28}$$

$$x_6 = 4 - x_1 + 2x_2 - 2x_3 \tag{29}$$

Kao i kod standardnog oblika, pogodnije je koristiti koncizniju notaciju za opisivanje labavog oblika. Kao što ćemo videti u narednoj sekciji, skupovi osnovnih i sporednih promenljivih će se menjati tokom izvršavanja simpleks algoritma. Sa N ćemo označiti skup sporednih promenljivih, a sa B skup osnosvnih promenljivih. Uvek će važiti da je $|N| = n$, $|B| = m$ i $N \cup B = \{1,2,\ldots,n+m\}$. Jednačine će biti indeksirane po članovima skupa B, a promenljive na desnoj strani će biti indeksirane po članovima skupa N. Kao i kod standardne forme, koristićemo b_i, c_j i a_{ij} da označimo konstantne članove i koeficijente. Koristićemo takođe i v da označimo opcioni konstantni član ciljne funkcije. Sada se labav oblik može predstaviti kao

$$z = v + \sum_{j \in N} c_j x_j \tag{30}$$

$$x_i = b_i - \sum_{j \in N} a_{ij} x_j \qquad \text{za } i \in B \tag{31}$$

pri čemu su sve promenljive nenegativne. Pošto u (31) oduzimamo sumu, to znači da su koeficijenti a_{ij} negativni u odnosu na one koji se pojavljuju u labavom obliku.

Na primer, u labavom obliku

$$z = 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3}$$

$$x_1 = 8 + \frac{x_3}{6} + \frac{x_5}{6} - \frac{x_6}{3}$$

$$x_2 = 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3}$$

$$x_4 = 18 - \frac{x_3}{2} + \frac{x_5}{2}$$

imamo da je $B = \{1,2,4\}$, $N = \{3,5,6\}$,

$$A = \begin{pmatrix} a_{13} & a_{15} & a_{16} \\ a_{23} & a_{25} & a_{26} \\ a_{43} & a_{45} & a_{46} \end{pmatrix} = \begin{pmatrix} -1/6 & -1/6 & 1/3 \\ 8/3 & 2/3 & -1/3 \\ 1/2 & -1/2 & 0 \end{pmatrix},$$

$$b = \begin{pmatrix} b_1 \\ b_2 \\ b_4 \end{pmatrix} = \begin{pmatrix} 8 \\ 4 \\ 18 \end{pmatrix},$$

$c = (c_3, c_5, c_6)^T = (-1/6 \quad -1/6 \quad -2/3)^T$, a $v = 28$.

Simpleks algoritam

Simpleks algoritam je klasičan metod za rešavanje linearnih programa. Za razliku od većine algoritama koje smo izučavali, njegovo vreme izvršavanja nije polinomijalno u najgorem slučaju. Međutim, ovaj algoritam pruža uvid u linearno programiranje i u praksi se često pokazao kao veoma brz.

U nastavku ćemo prikazati osnovnu ideju koja je u osnovi svake iteracije simpleks algoritma. Svakoj iteraciji će biti pridruženo "osnovno rešenje", koje se jednostavno dobija iz labavog oblika linearnog programa: sve sporedne promenljive se postave na 0, a vrednosti osnovnih promenljivih se izračunavaju iz ograničenja jednakosti. Algebarski, jedna iteracija transformiše jedan labavi oblik u ekvivalentni labavi oblik. Ciljna vrednost pridruženog osnovnog mogućeg rešenja neće biti manja od one iz prethodne iteracije (najčešće je veća). Da bismo postigli ovo povećanje ciljne vrednosti, biramo sporednu promenljivu tako da ako bismo povećali vrednost promenljive sa nule i ciljna vrednost bi se takođe povećala. Vrednost za koju možemo uvećati promenljivu je ograničena ostalim ograničenjima. Promenljivu uvećavamo sve dok neka osnovna promenljiva ne postane jednaka nuli. Zatim prepisujemo slabi oblik tako što zamenjujemo uloge te osnovne promenljive i odabrane sporedne promenljive. Na ovaj način algoritam prepisuje linearni program sve dok optimalno rešenje ne postane "očigledno".

Primer simpleks algoritma

Posmatrajmo sledeći linearni program u standardnom obliku:

maksimizovati

$$3x_1 + x_2 + 2x_3 \tag{32}$$

pri ograničenjima

$$x_1 + x_2 + 3x_3 \le 30 \tag{33}$$

$$2x_1 + 2x_2 + 5x_3 \le 24 \tag{34}$$

$$4x_1 + x_2 + 2x_3 \le 36 \tag{35}$$

$$x_1, x_2, x_3 \ge 0 \tag{36}$$

Da bismo koristili simpleks algoritam, moramo transformisati linearni program u labav oblik, kao što smo pokazali u prethodnoj sekciji. Imajući u vidu da svaka promenljiva ima odgovarajuće ograničenje nenegativnosti, za neko ograničenje jednakosti kažemo da je **tesno** za određen skup vrednosti njegovih sporednih promenljivih ukoliko ona daju vrednost osnovne promenljive jednaku 0. Slično, vrednosti sporednih promenljivih koje daju negativnu vrednost osnovne promenljive **narušavaju** ograničenje. Iz tog razloga, labave promenljive direktno čuvaju informaciju o tome koliko je svako ograničenje daleko od toga da bude tesno i na taj način nam pomažu u određivanju koliko možemo uvećati vrednosti sporednih promenljivih bez narušavanja bilo kog ograničenja.

Povezivanjem labavih promenljivih x_4, x_5 i x_6 sa nejednakostima (33)-(36) redom i transformisanjem linearnog programa u labav oblik, dobijamo

$$\begin{array}{ccccccccc} z & = & & & 3x_1 & + & x_2 & + & 2x_3 \\ x_4 & = & 30 & - & x_1 & - & x_2 & - & 3x_3 \\ x_5 & = & 24 & - & 2x_1 & - & 2x_2 & - & 5x_3 \\ x_6 & = & 36 & - & 4x_1 & - & x_2 & - & 2x_3 \end{array} \tag{37}$$

Sistem ograničenja (37) ima 3 jednačine i 6 promenljivih. Svaki skup vrednosti promenljivih x_1, x_2 i x_3 definišu vrednosti promenljivih x_4, x_5 i x_6, pa samim tim postoji beskonačno mnogo rešenja ovog sistema jednačina. Rešenje je moguće ukoliko su sve promenljive x_1, x_2, ..., x_6 nenegativne, tako da može biti i beskonačno mnogo mogućih rešenja. Mi ćemo se usresrediti na **osnovno rešenje**: postaviti sve (sporedne) promenljive na desnoj strani na 0 i izračunati vrednosti (osnovnih) promenljivih na levoj strani. U ovom primeru osnovno rešenje je $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_6) = (0,0,0,30,24,36)$, a njegova ciljna vrednost je $z = (3\cdot 0) + (1\cdot 0) + (2\cdot 0) = 0$. Primetimo da ovo osnovno rešenje postavlja $\bar{x}_i = b_i$ za svako $i \in B$. Jedna iteracija simpleks algoritma će prepisati skup jednačina i ciljnu funkciju, tako da postavi drugačiji skup promenljivih na desnu stranu. Tako će postojati drugačije osnovno rešenje za transformisan problem. Naglasimo da transformisanje ne menja linearni program ni na koji način; problem u jednoj iteraciji ima ima identičan skup mogućih rešenja kao i problem u prethodnoj iteraciji. Ovaj problem, međutim, ima drugačije osnovno rešenje od onog u prethodnoj iteraciji.

Ako je osnovno rešenje takođe i moguće, onda ga nazivamo **osnovnim mogućim rešenjem**. Tokom izvršavanja simpleks algoritma, osnovno rešenje će skoro uvek biti osnovno moguće rešenje.

Naš cilj u svakoj iteraciji je da preformulišemo linearni program tako da osnovno rešenje ima veću ciljnu vrednost. Biramo sporednu promenljivu x_e čiji je koeficijent u ciljnoj funkciji pozitivan i uvećavamo vrednost x_e koliko god je moguće, a da pri tome ne narušimo neko od ograničenja. Promenljiva x_e postaje osnovna, a neka druga promenljiva x_l postaje sporedna. Vrednosti ostalih osnovnih promenljivih i ciljne funkcije se mogu takođe promeniti.

Da bismo nastavili sa primerom, razmotrimo uvećavanje vrednosti promenljive x_1. Sa povećanjem x_1, vrednosti promenljivih x_4, x_5 i x_6 se smanjuju. S obzirom da imamo ograničenja nenegativnosti za svaku promenljivu, ne smemo dozvoliti da bilo koja od njih postane negativna. Ukoliko x_1 povećamo preko 30, onda x_4 postaje negativno, dok x_5 i x_6 postaju negativni ako se x_1 poveća preko 12 i 9 redom. Treće ograničenje je najtesnije i ono ograničava koliko se može uvećati x_1. Zbog toga ćemo zameniti uloge promenljivama x_1 i x_6. Rešavamo poslednju jednačinu po x_1 i dobijamo

$$x_1 = 9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{4} \tag{38}$$

Da bismo prepisali ostale jednačine tako da x_6 bude na desnoj strani, zamenjujemo x_1 korišćenjem jednačine (38) i dobijamo

$$\begin{array}{rcrcccccc}
z & = & 27 & + & \frac{x_2}{4} & + & \frac{x_3}{2} & - & \frac{3x_6}{4} \\
x_1 & = & 9 & - & \frac{x_2}{4} & - & \frac{x_3}{2} & - & \frac{x_6}{4} \\
x_4 & = & 21 & - & \frac{3x_2}{4} & - & \frac{5x_3}{2} & + & \frac{x_6}{4} \\
x_5 & = & 6 & - & \frac{3x_2}{2} & - & 4x_3 & + & \frac{x_6}{2}
\end{array} \tag{39}$$

Ovu operaciju nazivamo **pivot**. Kao što je prikazano, pivot bira sporednu promenljivu x_e, koju nazivamo **ulaznom promenljivom**, i osnovnu promenljivu x_l, koju nazivamo **izlaznom promenljivom**, i zamenjuje njihove uloge.

Linearni program opisan u (39) je ekvivalentan linearnom programu opisanom u jednačinama (37). Operacije koje primenjujemo u simpleks algoritmu su prepisivanje jednačina tako da se promenljive prebacuju sa jedne strane na drugu i zamena jedne jednačine u drugoj. Prva operacija kreira ekvivalentan problem, a druga, pomoću elementarne linearne algebre, takođe kreira ekvivalentan problem.

Da bismo pokazali ovu ekvivalenciju, posmatrajmo početno osnovno rešenje $(0,0,0,30,24,36)$ koje zadovoljava nove jednačine (39) i koje ima ciljnu vrednost $27 + \frac{1}{4}\cdot 0 + \frac{1}{2}\cdot 0 - \frac{3}{4}\cdot 36 = 0$. Osnovno rešenje pridruženo novom linearnom programu postavlja sporedne promenljive na 0 i jednako je $(9,0,0,21,6,0)$ sa ciljnom vrednošću $z = 27$. Jednostavnom aritmetikom se može pokazati da ovo rešenje takođe zadovoljava jednačine (37) i kada se zameni u toj ciljnoj funkciji ima ciljnu vrednost $(3\cdot 9) + (1\cdot 0) + (2\cdot 0) = 27$.

Nastavljajući primer, želimo da pronađemo novu promenljivu čiju vrednost hoćemo da uvećavamo. Ne želimo da uvećavamo x_6, s obzirom da se pri povećanju njene vrednosti ciljna vrednost umanjuje. Možemo pokušati da uvećavamo ili x_2 ili x_3; izaberimo x_3. Koliko možemo da uvećamo x_3 bez narušavanja ograničenja? Prvo ograničenja ograničava na 18, drugo na 42/5, a treće na 3/2. Treće ograničenje je ponovo najtesnije, tako da x_3 prelazi na levu stranu, a x_5 na desnu. Ponovo ovu novu jednačinu zamenjujemo u ostalim jednačinama i dobijamo novi, ekvivalentni sistem

$$\begin{array}{rcrcrcrcr} z & = & \frac{111}{4} & + & \frac{x_2}{16} & - & \frac{x_5}{8} & - & \frac{11x_6}{16} \\ x_1 & = & \frac{33}{4} & - & \frac{x_2}{16} & + & \frac{x_5}{8} & - & \frac{5x_6}{16} \\ x_3 & = & \frac{3}{2} & - & \frac{3x_2}{8} & - & \frac{x_5}{4} & + & \frac{x_6}{8} \\ x_4 & = & \frac{69}{4} & + & \frac{3x_2}{16} & + & \frac{5x_5}{8} & - & \frac{x_6}{16} \end{array} \qquad (40)$$

Ovaj sistem ima pridruženo osnovno rešenje $(33/4, 0, 3/2, 69/4, 0, 0)$, sa ciljnom vrednošću 111/4. Sada je jedini način da uvećamo ciljnu vrednost uvećanje promenljive x_2. Tri ograničenja daju gornje granice 132, 4 i ∞, redom. Promenljivu x_2 uvećavamo na 4 i ona postaje osnovna. Zatim rešavamo pretposlednju jednačinu i zamenjujemo je u ostalim jednačinama, tako da dobijamo

$$\begin{array}{rcrcrcrcr} z & = & 28 & - & \frac{x_3}{6} & - & \frac{x_5}{6} & - & \frac{2x_6}{3} \\ x_1 & = & 8 & + & \frac{x_3}{6} & + & \frac{x_5}{6} & - & \frac{x_6}{3} \\ x_2 & = & 4 & - & \frac{8x_3}{3} & - & \frac{2x_5}{3} & + & \frac{x_6}{3} \\ x_4 & = & 18 & - & \frac{x_3}{2} & + & \frac{x_5}{2} & & \end{array} \qquad (41)$$

U ovom trenutku svi koeficijenti u ciljnoj funkciji su negativni. Ovo se dešava samo u slučaju kada smo prepisali linearni program tako da je osnovno rešenje ujedno i optimalno rešenje. Tako je za ovaj problem rešenje $(8, 4, 0, 18, 0, 0)$, sa ciljnom vrednošću 28, optimalno rešenje. Sada se možemo vratiti na početni linearni problem u (32)-(36). Jedine promenljive u originalnom linearnom programu su x_1, x_2 i x_3, tako da je naše rešenje $x_1 = 8$, $x_2 = 4$ i $x_3 = 0$, sa ciljnom vrednošću $(3 \cdot 8) + (1 \cdot 4) + (2 \cdot 0) = 28$.

Pivotiranje

Sada ćemo formalizovati proceduru za pivotiranje. Procedura PIVOT uzima kao ulaz labav oblik, dat kao n-torka (N, B, A, b, c, v), indeks l izlazne promenljive x_l i indeks e ulazne promenljive x_e. Procedura vraća n-torku $(\hat{N}, \hat{B}, \hat{A}, \hat{b}, \hat{c}, \hat{v})$ koja opisuje novi labavi oblik.

PIVOT(N, B, A, b, c, v)

```
    // Racunanje koeficijenata jednacina za novu osnovnu promenljivu x_e.
    b̂_e = b_l / a_le
    for each j ∈ N − {e}
        â_ej = a_lj / a_le
    â_el = 1 / a_le
```

6 `// Racunanje koeficijenata za ostala ogranicenja.`
7 **for each** $i \in B - \{l\}$
8 $\quad \hat{b}_i = b_i - a_{ie}\hat{b}_e$
9 $\quad$ **for each** $j \in N - \{e\}$
10 $\quad\quad \hat{a}_{ij} = a_{ij} - a_{ie}\hat{a}_{ej}$
11 $\quad \hat{a}_{il} = -a_{ie}\hat{a}_{el}$
12 `// Racunanje ciljne funkcije.`
13 $\hat{v} = v + c_e\hat{b}_e$
14 **for each** $j \in N - \{e\}$
15 $\quad \hat{c}_j = c_j - c_e\hat{a}_{ej}$
16 $\hat{c}_l = -c_e\hat{a}_{el}$
17 `// Racunanje novog skupa osnovnih i sporednih promenljivih.`
18 $\hat{N} = N - \{e\} \cup \{l\}$
19 $\hat{B} = B - \{l\} \cup \{e\}$
20 **return** $\left(\hat{N}, \hat{B}, \hat{A}, \hat{b}, \hat{c}, \hat{v}\right)$

Procedura PIVOT funkcioniše na sledeći način. Linije 2-5 računaju koeficijente u novoj jednačini za x_e, prepisujući jednačinu koja ima x_l sa leve strane u jednačinu koja ima x_e sa leve strane. Linije 7-11 ažuriraju ostale jednačine tako što zamenjuju svako pojavljivanje promenljive x_e levom stranom ove nove jednačine. Linije 13-16 vrše istu zamenu u ciljnoj funkciji, a linije 18 i 19 ažuriraju skup sporednih i osnovnih promenljivih. Linija 20 vraća novi labav oblik. Prilikom pozivanja procedure PIVOT mora se voditi računa da a_{le} bude različito od nule, kako bi se izbeglo deljenje nulom u linijama 4 i 5.

Formalni simpleks algoritam

Sada smo spremni da formalizujemo simpleks algoritam prikazan u primeru. Pretpostavimo da imamo proceduru INITIALIZE_SIMPLEX(A, b, c) koja kao ulaz uzima linearni program u standardnom obliku, tj. matricu $A = \left(a_{ij}\right)$ dimenzija $m \times n$, m-dimenzioni vektor $b = \left(b_i\right)$ i n-dimenzioni vektor $c = \left(c_j\right)$. Ukoliko problem nije moguć, ova procedura vraća poruku da je program nemoguć i zatim se prekida izvršavanje. U suprotnom, procedura vraća labav oblik za koji je početno osnovno rešenje moguće.

Procedura SIMPLEX uzima kao ulaz linearni program u labavom obliku, a vraća n-dimenzioni vektor $\bar{x} = \left(\bar{x}_j\right)$ koji je optimalno rešenje opisanog linearnog problema.

`SIMPLEX`(A,b,c)
$(N,B,A,b,c,v) \leftarrow$ `INITIALIZE_SIMPLEX`(A,b,c)
while postoji indeks $j \in N$ za koji je $c_j > 0$
$\quad$ izaberi neki indeks $e \in N$ za koji je $c_e > 0$
$\quad$ **for each** $i \in B$
$\quad\quad$ **if** $a_{ie} > 0$ **then**
$\quad\quad\quad \Delta_i = b_i / a_{ie}$
$\quad\quad$ **else** $\Delta_i = \infty$
$\quad$ izaberi neki indeks $l \in B$ koji minimizuje Δ_l
$\quad$ **if** $\Delta_l = \infty$ **then**
$\quad\quad$ **return** “neogranicen”

11   **else** $(N,B,A,b,c,v) \leftarrow$ PIVOT(N,B,A,b,c,v,l,e)
12   **for** $i=1,n$
13   **if** $i \in B$ **then**
14             $\overline{x}_i = b_i$
15   **else** $\overline{x}_i = 0$
16   **return** $(\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$

Procedura SIMPLEX funkcioniše na sledeći način. U linije 1 se poziva procedura INITIALIZE_SIMPLEX(A,b,c), koja određuje da li je program nemoguć ili vraća labav oblik za koji je osnovno rešenje moguće. Osnovni deo algoritma je dat u **while** petlji u linijama 2-11. Ako su svi koeficijenti u ciljnoj funkciji negativni, onda se **while** petlja završava. U suprotnom, u liniji 3 biramo promenljivu x_e čiji je koeficijent u ciljnoj funkciji pozitivan za ulaznu promenljivu. Pošto imamo slobodu da izaberemo bilo koju takvu promenljivu za ulaznu, pretpostavićemo da koristimo neko deterministično pravilo. Zatim u linijama 4-8 ispitujemo svako ograničenje i uzimamo ono koje najtesnije ograničava vrednost za koju možemo uvećati x_e, a da pri tome ne narušimo ograničenja nenegativnosti. Osnovna promenljiva povezana sa ovim ograničenjem je x_l. Ukoliko nijedno ograničenje ne limitira vrednost za koju se može uvećati ulazna promenljiva, algoritam u liniji 10 vraća poruku da je problem "neograničen". U suprotnom linija 11 zamenjuje uloge ulaznoj i izlaznoj promenljivoj pozivanjem procedure PIVOT(N,B,A,b,c,v,l,e). Linije 12-15 izračunavaju rešenje za originalne promenljive $\overline{x}_1, \overline{x}_2, ..., \overline{x}_n$ postavljajući vrednosti svih sporednih promenljivih na 0, a svih osnovnih promenljivih $\overline{x}_i$ na b_i. Može se dokazati da je to rešenje optimalno rešenje linearnog programa. Konačno, linija 16 vraća izračunate vrednosti promenljivih linearnog programiranja.