Operaciona istraživanja

Buda Bajić

30. mart 2020.

Sadržaj

Glava 1

Linearno programiranje

1.1 Motivacija

Posmatrajamo proizvodnju u jednoj fabrici. Fabrika proizvodi proizvode koje numerišemo od $1,2,\ldots,n$. Ovi proizvodi su proizvedeni od različitih sirovina. Pretpostavimo da fabrika raspolaže sa m ražličitih sirovina koje numerišemo sa $1,\ldots,m$. Zalihe sirovine i ćemo označiti sa $b_i, i=1,2,\ldots,m$. Svaka sirovina ima svoju tržišnu vrednost ρ_i. Dodatno, svaki proizvod je sačinjen od određene količine pojedinih sirovina. Neka je za proizvodnju proizvoda j potrebno $a_{i,j}$ sirovine i. Takođe, proizvod j se prodaje po tržišnoj ceni σ_j. U nastavku posmatramo dva optimizaciona problema.

1.1.1 Optimističan pogled proizvođača

Pretpostavimo da proizvođač odluči da proizvede x_j jedinica proizvoda j, $j=1,2,\ldots,n$. Prihod od prodaje jedne jedinice proizvoda j je σ_j. S druge strane, trošak za proizvodnju jedne jedinice proizvoda j iznosi $\sum_{i=1}^{m}\rho_i a_{i,j}$. Stoga, zarada od jedne jedinice proizvoda j je razlika između prihoda i troškova. Ako zaradu od prodaje jedne jedinice proizvoda j označimo sa c_j, onda

$$c_j = \sigma_j - \sum_{i=1}^{m}\rho_i a_{ij}, \quad j=1,2,\ldots,n.$$

Ukupan prihod proizvođača iznosi

$$\sum_{j=1}^{n} c_j x_j. \tag{1.1}$$

Cilj proizvođača je da maksimizira ukupan prihod. Ograničenja koja se javljaju su sledeća. Svaka količina proizvoda j, x_j mora biti nenegativna, tj.

$$x_j \geq 0, \quad \forall j=1,\ldots,n. \tag{1.2}$$

Dalje, ne može se proizvesti više proizvoda nego što ima sirovina na raspolaganju, tj. važi ograničenje

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i, \quad i = 1,2,\ldots,m. \tag{1.3}$$

Da sumiramo, proizvođač želi da utvrdi koliko jedinica x_j proizvoda j, $j = 1,2,\ldots,n$ treba da proizvede da bi ostvario maksimalan prihod (1.1) od prodaje svih proizvoda pod datim uslovima (1.2) i (1.3).

1.1.2 Pesimističan pogled magacionera

S druge strane, magacioner treba da odredi inventarsku vrednost w_i sirovine $i = 1,2,\ldots,m$ tako da vrednosti w_i ne budu manje od tražišnih vrednosti $\rho_i, i = 1,2,\ldots,m$ i da vrednost jedinice proizvoda j sa ugrađenim sirovinama vrednosti w_i ne bude manja od tržišne vrednosti jedinice proizvoda σ_j, $j = 1,2,\ldots,n$. Cilj magacionera je da minimizira ukupnu vrednost lagera (da učini da finansijski izveštaj izgleda što bolje)

$$\sum_{i=1}^{m} b_i w_i \tag{1.4}$$

pod pomenutim uslovima, tj. pod uslovom da

$$w_i \geq \rho_i, \quad i = 1,2,\ldots,m, \tag{1.5}$$

i

$$\sum_{i=1}^{m} w_i a_{ij} \geq \sigma_j, \quad j = 1,2,\ldots,n. \tag{1.6}$$

Minimiziranje (1.4) pod ograničenjima (1.5) i (1.6) je problem linearnog programiranja. On postaje za nijansu manje složen ako uvedemo smenu

$$y_i = w_i - \rho_i, \quad i = 1,2,\ldots,m.$$

Veličina y_i predstavlja dodatnu vrednost sirovine i koju bi kompanija zaradila preprodajući sirovinu i. U terminima ove promenljive, cilj magacionera je da minimizira

$$\sum_{i=1}^{m} b_i y_i$$

pod uslovima

$$y_i \geq 0, \quad i = 1,2,\ldots,m,$$

i

$$\sum_{i=1}^{m} y_i a_{ij} \geq c_j, \quad j = 1,2,\ldots,n.$$

Primetimo da smo izostavili $\sum_{i=1}^{m} b_i \rho_i$ iz funkcije cilja jer je to konstanta i neće uticati na pronalaženje optimalne vrednosti promenljivih (što je glavni cilj magacionera).

1.2 Problem linearnog programiranja

U prethodna dva primera, javljale su se promenljive čiju optimalnu vrednost je bio cilj odrediti. One se nazivaju *odlučujuće promenljive*. One se obično označavaju sa

$$x_j, \quad j = 1, 2, \ldots, n.$$

U linearnom programiranju, cilj je uvek da se maksimizira ili minimizira neka linearna funkcija odlučujućih promenljivih

$$\zeta = c_1x_1 + c_2x_2 + \cdots + c_nx_n.$$

Ova funkcija se zove *funkcija cilja*.

Pored funkcije cilja, prethodni primeri su dodatno imali razna ograničenja koja u opštem slučaju predstavljaju linearnu kombinaciju odlučujućih promenljivih:

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq b.$$

U nekim slučajevima ova ograničenja su bila oblika jednakosti ili $\geq$. Primetimo da u slučajevima kada su ograničenja data kao jednakost ili $\geq$, možemo ih lako pretvoriti u ograničenja $\leq$. Na primer, ograničenje oblika

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \geq b$$

se množenjem sa -1 pretvara u ograničenje oblika $\leq$, dok ograničenje

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b$$

se pretvara u dva ekvivalentna ograničenja sa nejednakostima

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq b$$

i

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n \geq b.$$

Stoga, ne postoji preferencija kako će ograničenja biti postavljena (sve dok god su linearna). Međutim sa matematičkog stanovišta postoji preferencija da ograničenja treba da budu zapisana u obliku $\leq$ kao i da sve odlučujuće promenljive treba da budu nenegativne. Stoga, problem linearnog programiranja koji ćemo mi posmatrati treba da bude formulisan na sledeći način:

$$\begin{array}{ccccccccc}
c_1x_1 & + & c_2x_2 & + & \ldots & + & c_nx_n & \to & \min \\
a_{11}x_1 & + & a_{12}x_2 & + & \ldots & + & a_{1n}x_n & \leq & b_1 \\
a_{21}x_1 & + & a_{22}x_2 & + & \ldots & + & a_{2n}x_n & \leq & b_2 \\
\vdots & & & & & & & & \\
a_{m1}x_1 & + & a_{m2}x_2 & + & \ldots & + & a_{mn}x_n & \leq & b_m \\
\multicolumn{9}{c}{x_1, x_2, \ldots, x_n \geq 0}
\end{array}$$

Ovako zapisan problem linearnog programiranja zovemo *standardnim*. Radi konzistentnosti, uvek će m označavati broj ograničenja a n broj promenljivih. Uređena n-torka $(x_1, x_2, \dots, x_n)$ se zove *dopustiva* ako zadovoljava sva ograničenja, a dodatno ako se u njoj dostiže željeni maksimum onda se zove *optimalna*.

Neki problemi su *nedopustivi*, kao na primer:

$$\begin{array}{cccccc}
5x_1 & + & 4x_2 & \to & \max \\
x_1 & + & x_2 & \leq & 2 \\
-2x_1 & - & 2x_2 & \leq & -9 \\
x_1, x_2 \geq 0 & & & &
\end{array}$$

Zaista, drugo ograničenje implicira da je $x_1 + x_2 \geq 4{,}5$, što je kontradiktorno prvom ograničenju.

Druga ekstremna situacija se javlja kada funkcija cilja nije ograničena - tada se problem naziva *neograničen*. Na primer:

$$\begin{array}{ccccc}
x_1 & - & 4x_2 & \to & \max \\
-2x_1 & + & x_2 & \leq & -1 \\
-x_1 & - & 2x_2 & \leq & -2 \\
\multicolumn{5}{c}{x_1, x_2 \geq 0.}
\end{array}$$

Ovde možemo uzeti da je x_2 jednako nuli a da je x_1 proizvoljno veliko. Dok god je x_1 veće od 2 rešenje je dopustivo, i kako se x_1 povećava tako raste i vrednost funkcije cilja.

1.3 Simpleks metoda

1.3.1 Primer

Prvo ćemo ilustrovati kako simpleks metoda radi na konkretnom primeru.

$$\begin{array}{ccccccc}
5x_1 & + & 4x_2 & + & 3x_3 & \to & \max \\
2x_1 & + & 3x_2 & + & x_3 & \leq & 5 \\
4x_1 & + & x_2 & + & 2x_3 & \leq & 11 \\
3x_1 & + & 4x_2 & + & 2x_3 & \leq & 8 \\
\multicolumn{7}{c}{x_1, x_2, x_3 \geq 0.}
\end{array}$$

Počinjemo da rešavamo tako što dodajemo *pomoćne promenljive*. Za svaku nejednakost $\leq$ uvodimo nove promenljive koje predstavljaju razliku između desne i leve strane nejednakosti. Na primer, za prvu nejednakost

$$2x_1 + 3x_2 + x_3 \leq 5,$$

uvodimo pomoćnu promenljivu w_1 koju definišemo kao

$$w_1 = 5 - 2x_1 - 3x_2 - x_3.$$

Jasno je da ovakva definicija promenljive w_1 zajedno sa uslovom nenegativnosti, tj. $w_1 \geq 0$, je ekvivalentna originalnom uslovu. Ponavljamo ovu proceduru za svako ograničenje $\leq$ da bismo dobili ekvivalentu reprezentaciju problema:

$$
\begin{array}{rcrcrcrcrcl}
\zeta & = & & & 5x_1 & + & 4x_2 & + & 3x_3 & \rightarrow & \max \\
w_1 & = & 5 & - & 2x_1 & - & 3x_2 & - & x_3 & & \\
w_2 & = & 11 & - & 4x_1 & - & x_2 & - & 2x_3 & & \\
w_3 & = & 8 & - & 3x_1 & - & 4x_2 & - & 2x_3 & & \\
& & \multicolumn{7}{c}{x_1, x_2, x_3, w_1, w_2, w_3 \geq 0.} & &
\end{array}
$$

Primetimo da smo uveli oznaku ζ za vrednost funkcije cilja $5x_1 + 4x_2 + 3x_3$.

Simpleks metoda je iterativni proces koji kreće sa uređenom 6-torkom $(x_1, x_2, \ldots, w_3)$ koja zadovoljava jednačine (dopustiva uređena 6-torka) i uslove nenegativnosti i onda traži novu bolju dopustivu uređenu 6-torku $(\bar{x}_1, \bar{x}_2, \ldots, \bar{w}_3)$ u kojoj funkcija cilja ima veću vrednost:

$$5\bar{x}_1 + 4\bar{x}_2 + 3\bar{x}_3 \geq 5x_1 + 4x_2 + 3x_3.$$

Ovaj proces se nastavlja dok god se ne stigne do uređene 6-torke za koju se vrednost funkcije cilja ne može više povećati. Takva uređena 6-torka je optimalno rešenje.

Da bismo započeli iterativni proces, treba nam polazna dopustiva uređena 6-torka $(x_1, x_2, \ldots, w_3)$. U našem primeru to je jednostavno. Proglasićemo originalne promenljive da budu jednake nuli i koristeći definišuće jednačine ćemo odrediti vrednosti pomoćnih promenljivih:

$$x_1 = 0, \quad x_2 = 0, \quad x_3 = 0, \quad w_1 = 5, \quad w_2 = 11, \quad w_3 = 8.$$

Funkcija cilja ima sada vrednost 0.

Mi se dalje pitamo da li se može povećati vrednost funkcije cilja. Kako je koeficijent u funkciji cilja uz x_1 pozitivan, ako povećamo vrednost promenljive x_1 sa nula na neku drugu pozitivnu vrednost povećaćemo i vrednost ζ. Ali menjajući vrednost promenljive x_1 promeniće se i vrednosti pomoćnih promenljivih. Moramo se osigurati da neka od pomoćnih promenljivih ne postane negativna. Kako su x_2 i x_3 trenutno jednaki nuli, imamo da je $w_1 = 5 - 2x_1$, tako da ako želimo da obezbedimo da w_1 ne bude negativna moramo ograničiti rast promenljive x_1 sa $5/2$. Slično, nenegativnost promenljive w_2 nameće uslov da $x_1 \leq 11/4$, dok nenegativnost promenljive w_3 nameće uslov da $x_1 \leq 8/3$. Kako svi ovi uslovi moraju biti zadovoljeni, vidimo da x_1 ne može biti veće od najmanje od ovih granica tj. $x_1 \leq 5/2$. Naše novo poboljšano rešenje je sada:

$$x_1 = \frac{5}{2}, \quad x_2 = 0, \quad x_3 = 0, \quad w_1 = 0, \quad w_2 = 1, \quad w_3 = \frac{1}{2}.$$

Prvi korak je bio jednostavan dok odgovor na pitanje kako da nastavimo nije toliko jednostavan. Ono što je učinilo prvi korak jednostavnim je to što some imali grupu promenljivih koje su inicijalno bile nule a ostale promenljive su bile izražene preko njih. Ovo je moguće postići i za novo rešenje. To se postiže tako što ćemo sistem jednačina napisati tako da x_1, w_2, w_3 i ζ budu izražene u funkciji od w_1, x_2 i x_3. Drugim rečima, uloge promenljivih x_1 i w_1 se moraju zameniti. Koristeći prvu jednačinu iz sistema jednačina dobijamo da je

$$x_1 = \frac{5}{2} - \frac{1}{2}w_1 - \frac{3}{2}x_2 - \frac{1}{2}x_3.$$

Jednačine za w_2, w_3 i ζ se moraju srediti tako da se x_1 ne pojavljuje na desnoj strani. Najjednostavniji način da se to postigne je pomoću ekvivalentnih transformacija (jedna jednačina se množi brojem i sabira se sa drugom jednačinom). Na primer, ako uzmemo jednačinu za w_2 i oduzmemo od nje jednačinu za w_1 pomnoženu sa 2 i prebacimo w_1 na desnu stranu dobićemo

$$w_2 = 1 + 2w_1 + 5x_2.$$

Primenjujući ekvivalentne transformacije za w_3 i ζ, dobijamo:

$$\begin{array}{ccccccccc}
\zeta & = & 12{,}5 & - & 2{,}5w_1 & - & 3{,}5x_2 & + & 0{,}5x_3 \\
x_1 & = & 2{,}5 & - & 0{,}5w_1 & - & 1{,}5x_2 & - & 0{,}5x_3 \\
w_2 & = & 1 & + & 2w_1 & + & 5x_2 & & \\
w_3 & = & 0{,}5 & + & 1{,}5w_1 & + & 0{,}5x_2 & - & 0{,}5x_3.
\end{array}$$

Primetimo da se trenutno rešenje može dobiti tako što se nezavisne promenljive proglase za nule a vrednost zavisnih promenljivih se pročita iz jednačina.

Sada vidimo da povećanje w_1 ili x_2 vodi ka smanjenju vrednosti funkcije cilja i da je x_3 jedina promenljiva sa pozitivnim koeficijentom u funkciji cilja čije povećanje vodi ka daljem povećanju vrednosti funkcije cilja. Opet moramo da utvrdimo koliko smemo maksimalno povećati ovu promenljivu a da pri tom ne narušimo uslov da su sve promenljive nenegativne. Ovog puta vidimo da jednačina za w_2 nije pogođena promenama promenljive x_3, dok jednačine za x_1 i w_3 nameću uslove da $x_3 \leq 5$ i $x_3 \leq 1$, redom. Poslednja nejednakost je strožija, tako da je novo rešenje:

$$x_1 = 2, \quad x_2 = 0, \quad x_3 = 1, \quad w_1 = 0, \quad w_2 = 1, \quad w_3 = 0.$$

Vrednost funkcije cilja je sad $\zeta = 13$. Opet ćemo pokušati da utvrdimo da li je moguće ponovo povećati vrednost funkcije cilja i ako jeste koliko. Stoga, moramo naše jednačine zapisati tako da ζ, x_1, w_2 i x_3 budu izražene u funkciji od w_1, x_2 i w_3. Rešavajući poslednju jednačinu za x_3 dobijamo

$$x_3 = 1 + 3w_1 + x_2 - 2w_3.$$

Takođe, primenjujući ekvivalentne transformacije možemo eliminisati x_3 iz ostalih jednačina. Ovo rezultuje sa:

$$\begin{array}{ccrcrcrcr}
\zeta & = & 13 & - & w_1 & - & 3x_2 & - & w_3 \\
x_1 & = & 2 & - & 2w_1 & - & 2x_2 & + & w_3 \\
w_2 & = & 1 & + & 2w_1 & + & 5x_2 & & \\
x_3 & = & 1 & + & 3w_1 & + & x_2 & - & 2w_3
\end{array}$$

Sada smo spremni da započenemo treću iteraciju. Prvi korak je da utvrdimo povećanje koje nezavisne promenljive će dovesti do daljeg povećanja funkcije cilja ζ. Ali ovog puta takva promenljiva ne postoji, jer sve promenljive u funkciji cilja imaju negativne koeficijente. Ova konstatacija je dokaz da smo pronašli optimalno rešenje $\zeta = 13$. Sistemi jednačina koje smo sretali u prethodnom primeru se zovu *rečnici*. Izuzev ζ, promenljive koje se nalaze na levoj strani jednakosti (zavisne promenljive) se zovu *bazične promenljive*. Promenljive na desnoj strani jednakosti (nezavisne promenljive) se zovu *nebazične promenljive*. Rešenja koja smo dobijali uzimajući da su nebazične promenljive jednake nuli se nazivaju bazična dopustiva rešenja.

1.3.2 Simpleks metoda

Posmatrajmo problem linearnog programiranja u standardnom obliku. Maksimiziramo funkciju cilja:

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

pod organičenjima

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad i = 1, 2, \dots, m$$

i

$$x_j \geq 0, \quad j = 1, 2, \dots, n.$$

Naš prvi zadatak je da uvedemo pomoćne promenljive i nazovemo funkciju cilja sa ζ:

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

i

$$w_i = b_i - \sum_{j=1}^{n} a_{ij} x_j, \quad i = 1, 2, \dots, m.$$

Kao što smo videli u prethodnom primeru, kako simpleks metoda napreduje, pomoćne promenljive menjaju ulogu sa originalnim promenljivama. Zgodno je uvesti notaciju u kojoj pomoćne promenljive su manje više isto tretirane kao i originalne promenljive. Jednostavno ćemo ih dodati na kraju liste x-promenljivih:

$$(x_1, \dots, x_n, w_1, \dots, w_m) = (x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m}).$$

Sa ovom notacijom možemo zapisati problem linearnog programiranja kao:

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

i

$$x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij} x_j, \quad i = 1, 2, \ldots, m.$$

To je početni rečnik. Kako simpleks metoda napreduje, ona prelazi sa jednog rečnika na drugi tražeći optimalno rešenje. Svaki rečnik ima m bazičnih promenljivih i n nebazičnih promenljivih. Neka je skup $\mathcal{B}$ podskup skupa indeksa $\{1, 2 \ldots, n+m\}$ koji odgovaraju bazičnim promenljivama, i neka je $\mathcal{N}$ skup indeksa koji odgovaraju nebazičnim promenljivama. Inicijalno, imamo da je $\mathcal{N} = \{1, 2, \ldots, n\}$ i $\mathcal{B} = \{n+1, n+2, \ldots, n+m\}$. Ovo se naravno menja kroz iteracije. Kako algoritam napreduje, trenutni rečnik izgleda:

$$\begin{array}{rcl} \zeta & = & \bar{\zeta} + \sum_{j \in \mathcal{N}} \bar{c}_j x_j \\ x_i & = & \bar{b}_i - \sum_{j \in \mathcal{N}} \bar{a}_{ij} x_j, \quad i \in \mathcal{B}. \end{array} \tag{1.7}$$

Primetimo da smo stavili crtice iznad koeficijenata kako bismo označili da se oni menjaju kako algoritam napreduje.

U svakoj iteraciji simpleks metode, tačno jedna nebazična promenljiva postaje bazična promenljiva. To smo videli i u našem primeru ali objasnimo sada i u opštem slučaju.

Promenljiva koja iz nebazične prelazi u bazičnu promenljivu se naziva *ulazna* promenljiva. Ona se bira sa ciljem da poveća vrednost ζ, i zbog toga se bira ona promenljiva čiji je koeficijent pozitivan: *izaberi k iz skupa* $\{j \in \mathcal{N} : \bar{c}_j > 0\}$. Ako je taj skup prazan onda je trenutno rešenje optimalno. Ako se skup sastoji od više takvih elemenata (što je najčešće slučaj), tada imamo izbor koji od elemenata da izaberemo. Postoji nekoliko kriterijuma za odabir koje ćemo diskutovati kasnije. Za sada je dovoljno da kažemo da se najčešće uzima indeks k čiji koeficijent je najveći.

Promenljiva koja napušta bazu tj. ona promenljiva koja od bazične postaje nebazična se naziva *izlazna* promenljiva. Ona se bira tako da se očuva nenegativnost trenutnih bazičnih promenljivih. Kada smo odlučili da je promenljiva x_k ulazna promenljiva, njena vrednost će se od nule povećati na neku pozitivnu vrednost. Njen porast će uticati na bazične promenljive:

$$x_i = \bar{b}_i - \bar{a}_{ik} x_k, \quad i \in \mathcal{B}.$$

Moramo osigurati da svaka od promenljivih ostane nenegativna, tj. moramo nametnuti uslov da je

$$\bar{b}_i - \bar{a}_{ik} x_k \geq 0, \quad i \in \mathcal{B}.$$

Od ovih izraza jedini koji mogu da postanu negativni sa porastom x_k su oni kod kojih je $\bar{a}_{ik}$ pozitivno, ostali ili ostaju nepromenjeni ili se povećavaju. Stoga našu pažnju možemo ograničiti na one i-ove za koje je $\bar{a}_{ik}$ pozitivno. Za takvo i, vrednost x_k za koje svaki izraz postaje nula je

$$x_k = \bar{b}_i / \bar{a}_{ik}.$$

Kako ne želimo da bilo koji od tih izraza postane negativan, moramo povećati x_k samo koliko je najmanja od pomenutih vrednosti:

$$x_k = \min_{i \in \mathcal{B}: \bar{a}_{ik} > 0} \bar{b}_i / \bar{a}_{ik}.$$

Konačno, pravilo za odabir izlazne promenljive je *izaberi l iz skupa* $\{i \in \mathcal{B} : \bar{a}_{ik} > 0$ *i* $\bar{b}_i/\bar{a}_{ik}$ *je minimalno*$\}$.

Jednom kada su izlazna i ulazna promenljiva odabrane, prelaz sa trenutnog rečnika na novi rečnik se odvija koristeći ekvivalentne transformacije. Ovaj korak prelaska sa jednog na drugi rečnik se naziva *pivotizacija*.

1.3.3 Simpleks metoda, tabelarni zapis - primer

Simpleks metoda se radi jednostavnosti najčešće u praksi zapisuje u obliku tabela. Vratimo se na naš primer i pogledajmo kako se na osnovu rečnika prave tabele i kako se vrše pivotizacije.

Podsetimo se najpre kako naš problem izgleda:

$$\begin{array}{ccccccc} 5x_1 & + & 4x_2 & + & 3x_3 & \to & \max \\ 2x_1 & + & 3x_2 & + & x_3 & \leq & 5 \\ 4x_1 & + & x_2 & + & 2x_3 & \leq & 11 \\ 3x_1 & + & 4x_2 & + & 2x_3 & \leq & 8 \\ & & & x_1, x_2, x_3 \geq 0. & & & \end{array}$$

Nakon uvođenja pomoćnih promenljivih imamo:

$$\begin{array}{ccccccccccccccc} \zeta & = & 5x_1 & + & 4x_2 & + & 3x_3 & & & & & & & \to & \max \\ & & 2x_1 & + & 3x_2 & + & x_3 & + & w_1 & & & & & = & 5 \\ & & 4x_1 & + & x_2 & + & 2x_3 & & & + & w_2 & & & = & 11 \\ & & 3x_1 & + & 4x_2 & + & 2x_3 & & & & & + & w_3 & = & 8 \\ & & & & & & x_1, x_2, x_3, w_1, w_2, w_3 \geq 0. & & & & & & & & \end{array}$$

Inicijalna tabela koja odgovara polaznom rečniku je data sa:

	x_1	x_2	x_3	w_1	w_2	w_3	
w_1	2	3	1	1	0	0	5
w_2	4	1	2	0	1	0	11
w_3	3	4	2	0	0	1	8
$-\zeta$	-5	-4	-3	0	0	0	0

i u njoj su upisani redom koeficijenti uz promenljive iz ograničenja i $-\zeta$.

U prvom koraku posmatramo koeficijente u poslednjoj vrsti (-5, -4, -3) i pošto smo sada zapisali $-\zeta$, onda **negativni** koeficijenti odgovaraju promenljivama čijim povećanjem će se ζ povećati. Najnegativniji koeficijent -5 sugeriše da nebažična promenljiva x_1 u prvom koraku ulazi u bazu. Da bismo odredili koja promenljiva izlazi iz baze, posmatramo strogo pozitivne koeficijente u koloni ispod promenljive x_1 (2, 4, 3) u pravimo količnike brojeva koji pišu u poslednjoj koloni (5, 11, 8) sa pomenutim brojevima (2, 4, 3). Dati količnici redom iznose $\frac{5}{2}$, $\frac{11}{4}$ i $\frac{8}{3}$. Najmanji od posmatranih količnika je $\frac{5}{2}$ i on daje gornje ograničenje za promenljivu x_1, tj. govori da promenljiva x_1 sme da poraste najviše $\frac{5}{2}$ a da pritom uslovi nenegativnosti svih promenljivih ne budu narušeni. Vrsta u kojoj se javlja ovaj količnik (prva vrsta) nam govori da će w_1 izaći iz baze i da će zameniti mesto sa x_1. Par promenljivih (x_1, w_1) se zove pivot.

Primetimo da ispod bazičnih pomenljivih (to su trenutno w_1, w_2 i w_3) u tabeli pišu sve nule sem jedne jedinice koja se baš nalazi u preseku vrste i kolone u kojoj se nalazi bazična promenljiva. Kada x_1 uđe u bazu da bismo postigli da ispod nje takođe pišu nule i jedino jedinica u prvoj vrsti, tj. da važi pomenuta osobina, treba da sprovedemo odgovarajuće ekvivalentne transformacije, tj. prva vrsta treba da se množi sa $\frac{1}{2}$, prva vrsta treba da se množi sa -2 i sabere sa drugom vrstom, sa $-\frac{3}{2}$ i sabere sa trećom vrstom, množi sa $\frac{5}{2}$ i sabere sa poslednjom vrstom. Tako dobijamo novu tabelu:

	x_1	x_2	x_3	w_1	w_2	w_3	
x_1	1	1,5	0,5	0,5	0	0	2,5
w_2	0	-5	0	-2	1	0	1
w_3	0	$-0{,}5$	0,5	$-1{,}5$	0	1	0,5
$-\zeta$	0	3,5	$-0{,}5$	2,5	0	0	12,5

Ponavljamo postupak. Posmatramo koeficijente u poslednjoj vrsti. Pošto nisu svi pozitivni (to bi bila optimalna tabela), gledamo koji koeficijent je najnegativniji. To je $-0{,}5$ i on se nalazi ispod promenljive x_3. Promenljiva x_3 ulazi u bazu. Dalje posmatramo količnike brojeva iz poslednje kolone i strogo pozitivnih koeficijenata koji se nalaze ispod promenljive x_3. Ti količnici iznose sada $\frac{2{,}5}{0{,}5} = 5$ i $\frac{0{,}5}{0{,}5} = 1$. Minimalni od tih količnika je 1 i on sugeriše da promenljiva w_3 izlazi iz baze. Vršimo pivotizaciju i dobijamo novu tabelu:

	x_1	x_2	x_3	w_1	w_2	w_3	
x_1	1	2	0	2	0	-1	2
w_2	0	-5	0	-2	1	0	1
x_3	0	-1	1	-3	0	2	1
$-\zeta$	0	3	0	1	0	1	13

Ova tabela je optimalna jer su u poslednjoj vrsti svi koeficijenti pozitivni. Iz nje čitamo optimalno rešenje: $x_1^* = 2$, $x_2^* = 0$ i $x_3^* = 1$ a optimalna vrednost funkcije cilja je $\zeta^* = 13$.

1.3.4 Inicijalizacija

U prethodnim primerima smo prezentovali simpleks metodu. Probleme koje smo posmatrali su bili takvi da su brojevi na desnoj strani nejednakosti bili nenegativni. To je obezbeđivalo da je polazni rečnik dopustiv. U ovom odeljku ćemo diskutovati šta se dešava kada to nije slučaj.

Problem linearnog programiranja koji smo posmatrali je

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

pod ograničenjima

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad i = 1,2,\ldots,m$$

i

$$x_j \geq 0, \quad j = 1,2,\ldots,n,$$

i za njega je inicijalni rečnik bio

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

i

$$w_i = b_i - \sum_{j=1}^{n} a_{ij} x_j, \quad i = 1,2,\ldots,m.$$

Rešenje vezano za ovaj rečnik se dobilo tako što smo svakom x_j dodelili vrednost nula a svakom w_i vrednost b_i. Ovo rešenje je dopustivo ako i samo ako su vrednosti b_i nenegativne. Ali šta ako to nije slučaj? Ovaj slučaj prevazilazimo uvođenjem *pomoćnog problema* za koji je

1. Dopustiv rečnik lako naći

2. Optimalni rečnik daje dopustiv rečnik originalnog problema.

Pomoćni problem je maksimizirati:

$$-x_0$$

pod organičenjima

$$\sum_{j=1}^{n} a_{ij} x_j - x_0 \leq b_i \quad i = 1,2,\ldots,m$$

i

$$x_j \geq 0, \quad j = 0,1,2,\ldots,n.$$

Jednostavno je naći dopustivo rešenje pomoćnog problema. Zaista, uzmimo da je $x_j = 0$ za $j = 1,2,\ldots,n$ i da je x_0 dovoljno veliko. Lako je videti da originalni problem ima dopustivo rešenje ako i samo ako pomoćni problem ima rešenje $x_0 = 0$. Drugim rečima, originalni problem ima dopustivo rešenje ako i samo ako je optimalno rešenje pomoćnog problema jednako nuli.

Iako pomoćni problem ima dopustiva rešenja, nismo još pokazali da ima dopustiv rečnik koji je lako dobiti. Najbolje je da na primeru ilustrujemo kako se dobija dopustiv rečnik:

$$\begin{array}{lrcrcr}
\text{maksimiziraj} & -2x_1 & - & x_2 & & \\
\text{pod uslovima} & -x_1 & + & x_2 & \leq & -1 \\
 & -x_1 & - & 2x_2 & \leq & -2 \\
 & & & x_2 & \leq & 1 \\
 & & & \multicolumn{3}{l}{x_1, x_2 \geq 0.}
\end{array}$$

Pomoćni problem je maksimizirati

$$\begin{array}{lrcrcrcr}
\text{maksimiziraj} & & & & - & x_0 & & \\
\text{pod uslovima} & -x_1 & + & x_2 & - & x_0 & \leq & -1 \\
 & -x_1 & - & 2x_2 & - & x_0 & \leq & -2 \\
 & & & x_2 & - & x_0 & \leq & 1 \\
 & & & \multicolumn{5}{l}{x_1, x_2, x_0 \geq 0.}
\end{array}$$

U sledećem koraku uvodimo pomoćne promenljive i zapisujemo inicijalni rečnik koji nije dopustiv:

$$\begin{array}{rcrcrcrcr}
\zeta & = & & & & & & - & x_0 \\
\hline
w_1 & = & -1 & + & x_1 & - & x_2 & + & x_0 \\
w_2 & = & -2 & + & x_1 & + & 2x_2 & + & x_0 \\
w_3 & = & 1 & & & - & x_2 & + & x_0
\end{array}$$

Ovaj rečnik je nedopustiv ali ga je lako transformisati u dopustiv rečnik. Sve što treba da uradimo je da x_0 uđe u bazu umesto najnedopustivije promenljive w_2:

$$\begin{array}{rcrcrcrcr}
\zeta & = & -2 & + & x_1 & + & 2x_2 & - & w_2 \\
\hline
w_1 & = & 1 & & & - & 3x_2 & + & w_2 \\
x_0 & = & 2 & - & x_1 & - & 2x_2 & + & w_2 \\
w_3 & = & 3 & - & x_1 & - & 3x_2 & + & w_2.
\end{array}$$

Primetimo da sada imamo dopustiv rečnik i da možemo kao i ranije da nastavimo da primenjujemo simpleks metodu. U prvom koraku uzimamo da x_2 uđe i w_1 izađe iz baze:

$$\begin{array}{rcrcrcrcr}
\zeta & = & -1{,}33 & + & x_1 & - & 0{,}67w_1 & - & 0{,}33w_2 \\
\hline
x_2 & = & 0{,}33 & & & - & 0{,}33w_1 & + & 0{,}33w_2 \\
x_0 & = & 1{,}33 & - & x_1 & + & 0{,}67w_1 & + & 0{,}33w_2 \\
w_3 & = & 2 & - & x_1 & + & w_1. & &
\end{array}$$

U drugom koraku uzimamo da x_1 ulazi u bazu a da x_0 izlazi iz baze:

$$\begin{array}{rcrcrcrcr}
\zeta & = & 0 & - & x_0 & & & & \\
\hline
x_2 & = & 0,33 & & & - & 0,33w_1 & + & 0,33w_2 \\
x_1 & = & 1,33 & - & x_0 & + & 0,67w_1 & + & 0,33w_2 \\
w_3 & = & 0,67 & + & x_0 & + & 0,33w_2 & - & 0,33w_2.
\end{array}$$

Dobijeni rečnik je otpimalan za pomoćni problem. Sada izostavljamo x_0 iz jednačina i vraćamo se na polazni problem:

$$\zeta = -2x_1 - x_2 = -3 - w_1 - w_2.$$

Stoga dopustiv rečnik za polazni problem je:

$$\begin{array}{rcrcrcr}
\zeta & = & -3 & - & w_1 & - & w_2 \\
\hline
x_2 & = & 0,33 & - & 0,33w_1 & + & 0,33w_2 \\
x_1 & = & 1,33 & + & 0,67w_1 & + & 0,33w_2 \\
w_3 & = & 0,67 & + & 0,33w_2 & - & 0,33w_2.
\end{array}$$

Ovaj rečnik je ujedno i optimalan jer su svi koeficijenti uz promenljive u funkciji cilja ζ negativni.

1.3.5 Neograničenost

Posmatrajmo primer

$$\begin{array}{rcrcrcr}
\zeta & = & 5 & + & x_3 & - & x_1 \\
\hline
x_2 & = & 5 & + & 2x_3 & - & 3x_1 \\
x_4 & = & 7 & & & - & 4x_1 \\
x_5 & = & & & & & x_1
\end{array}$$

Promenljiva koja bi trebala da uđe u bazu je promenljiva x_3 jer je koeficijent u funkciji cilja jedino uz nju pozitivan. Međutim, ako posmatramo količnike koji nam govore koja promenljiva treba da izađe iz baze, takva promenljiva ne postoji (svi koeficijenti u ograničenjima uz x_3 su negativni (-2) ili nule). To znači da povećanje promenljive x_3 nije ograničeno i ne dovodi do toga da neka od baznih promenljivih dobije vrednost nula i time postane nebazična promenljiva. Stoga je ovaj problem primer neograničenog problema gde funkcija cilja može da uzme proizvoljno veliku vrednost za dovoljno veliko x_3.

1.3.6 Grafičko rešavanje

Ako u problemu linearnog programiranja je broj promenljivih $n = 2$ ili $n = 3$, skup dopustivih vrednosti $\mathcal{S}$ se može nacrtati i lako se može videti šta je rešenje problema.

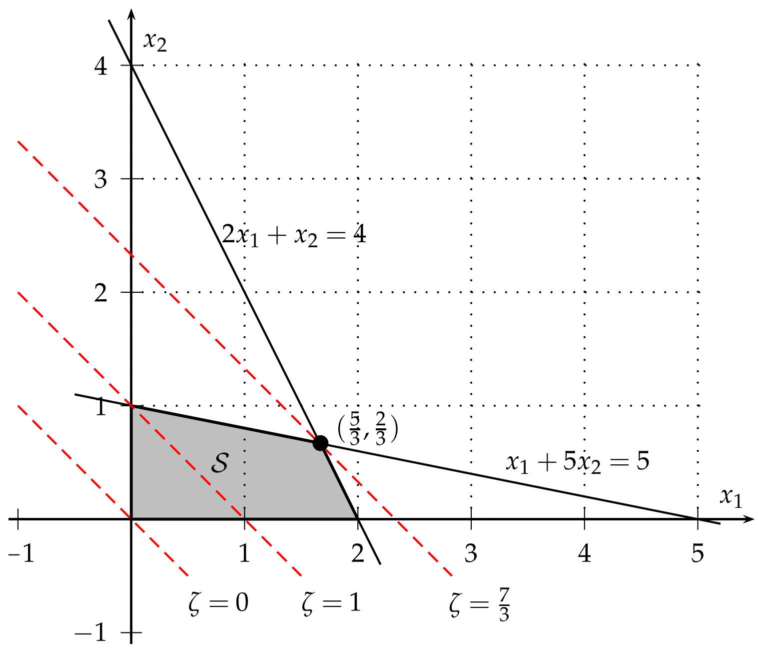
Na primer, za problem

$$\begin{array}{rcrcrcr}
\zeta & = & x_1 & + & x_2 & \to & \max \\
 & & x_1 & + & 5\,x_2 & \leq & 5 \\
 & & 2\,x_1 & + & x_2 & \leq & 4 \\
 & & \multicolumn{5}{c}{x_1 \geq 0 \quad x_2 \geq 0,}
\end{array}$$

optimalno rešenje očitavamo sa crteža (Slika **??**): $x_1^* = \dfrac{5}{3}$, $x_2^* = \dfrac{2}{3}$, $\zeta^* = \dfrac{7}{3}$.

Koordinate $(\frac{5}{3}, \frac{2}{3})$ dobijamo rešavanjem sistema

$$x_1 + 5x_2 = 5,\; 2x_1 + x_2 = 4.$$



Slika 1.1: Skup dopustivih rešenja zajedno sa izohipsama funkcije cilja.

Isprekidana linija predstavlja vrednosti funkcije $\zeta = x_1 + x_2$.

1.4 Degenerativnost

Kažemo da je rečnik degenerisan ako je $b_i = 0$ za neko $i \in \mathcal{B}$. Degenerisani rečnici mogu izazvati poteškoće u simpleks metodi ali i ne moraju. Na primer, rečnik:

$$\begin{array}{rcrcrcr}
\zeta & = & 5 & + & x_3 & - & x_1 \\
\hline
x_2 & = & 5 & + & 2x_3 & - & 3x_1 \\
x_4 & = & 7 & & & - & 4x_1 \\
x_5 & = & & & & & x_1
\end{array}$$

je degenerisan, ali je bilo jasno da je problem neograničen te stoga pivotizacije nisu ni potrebne. Dalje, da je koeficijent uz x_3 u jednačini za x_2 bio -2 umesto 2 tada bi simpleks metoda odabrala za izlaznu promenljivu x_2 i nikakvih poteškoća ne bi bilo.

Problem nastaje kada degenerisani rečnici proizvode degenerisane pivote. Kažemo da je pivot degenerisan ako je vrednost promenljive koja ulazi u bazu u novom rečniku jednaka nuli. Na primer, ovo je degenerisani rečnik:

$$\begin{array}{rcrcrcrcr} \zeta & = & 3 & - & 0{,}5x_1 & + & 2x_2 & - & 1{,}5w_1 \\ \hline x_3 & = & 1 & - & 0{,}5x_1 & & & - & 0{,}5w_1 \\ w_2 & = & & & x_1 & - & x_2 & + & w_1. \end{array}$$

Za ovaj rečnik ulazna promenljiva je x_2 a izlazna promenljiva je w_2. Kako svako povećanje x_2 implicira da promenljiva w_2 postane negativna, jasno je da x_2 ne može da postane pozitivna i mora ostati nula. U svakom slučaju može biti reklasifikovana kao bazična (do sada je bila nebazična). Pogledajmo šta je rezultat degenerisanog pivota:

$$\begin{array}{rcrcrcrcr} \zeta & = & 3 & + & 1{,}5x_1 & - & 2w_2 & + & 0{,}5w_1 \\ \hline x_3 & = & 1 & - & 0{,}5x_1 & & & - & 0{,}5w_1 \\ x_2 & = & & & x_1 & - & w_2 & + & w_1. \end{array}$$

Primetimo da $\bar{\zeta}$ i dalje ima istu vrednost kao i pre pivotizacije 3. Stoga degenerisani pivot nije doveo do porasta funkcije cilja. Dalje, vrednosti promenljivih se nisu promenile. I pre i posle pivotizacije one su:

$$(x_1, x_2, x_3, w_1, w_2) = (0,0,1,0,0).$$

Ali sada je rešenje predstavljeno pomoću novih bazičnih promenljivih i nadamo se da će novi pivot dovesti do povećanja funkcije cilja. Ulazna promenljiva za narednu iteraciju je x_1 a izlazna promenljiva je x_3 što nije više degenerisan pivot i vodi ka:

$$\begin{array}{rcrcrcrcr} \zeta & = & 6 & - & 3x_3 & - & 2w_2 & - & w_1 \\ \hline x_1 & = & 2 & - & 2x_3 & & & - & w_1 \\ x_2 & = & 2 & - & 2x_3 & - & w_2. & & \end{array}$$

Ovaj primer ilustruje šta se najčešće dešava kad se javi degenerisni pivot - neki od narednih pivota ne bude degenerisan i on nas odvede od degenerisanih rečnika. Međutim postoje primeri u kojima se javlja niz degenerisanih pivota koji eventualno dovede do pojave rečnika u kom se algoritam već našao ranije i na taj način se nikada ne stigne do optimalnog rešenja. Takva osobina se zove *kruženje* rečnika.

Nažalost kruženje rečnika se može pojaviti i kod najčešće korišćenog pravila pivotizacije u praksi:

- Izaberi ulaznu promenljivu za koju je koeficijent u funkciji cilja ζ pozitivan i najveći

- Kada se dve ili više promenljivih bore za to koja će biti izlazna, onda daj prednost promenljivama x i ako postoji izbor uzmi onu koja ima najmanji redni broj.

Primer kruženja rečnika:

$$
\begin{array}{rclrrrrrrrr}
\zeta & = & & & x_1 & - & 2x_2 & & & - & 2x_4 \\
\hline
w_1 & = & & - & 0{,}5x_1 & + & 3{,}5x_2 & + & 2x_3 & - & 4x_4 \\
w_2 & = & & - & 0{,}5x_1 & + & x_2 & + & 0{,}5x_3 & - & 0{,}5x_4 \\
w_3 & = & 1 & - & x_1 & & & & & &
\end{array}
$$

U prvoj pivotizaciji promenljiva x_1 ulazi u bazu dok promenljiva w_1 izlazi iz baze:

$$
\begin{array}{rclrrrrrrrr}
\zeta & = & & - & 2w_1 & + & 5x_2 & + & 4x_3 & - & 10x_4 \\
\hline
x_1 & = & & - & 2w_1 & + & 7x_2 & + & 4x_3 & - & 8x_4 \\
w_2 & = & & & w_1 & - & 2{,}5x_2 & - & 1{,}5x_3 & + & 3{,}5x_4 \\
w_3 & = & 1 & + & 2w_1 & - & 7x_2 & - & 4x_3 & + & 8x_4
\end{array}
$$

U drugoj iteraciji x_2 ulazi a w_2 izlazi iz baze:

$$
\begin{array}{rclrrrrrrrr}
\zeta & = & & & & - & 2w_2 & + & x_3 & - & 3x_4 \\
\hline
x_1 & = & & & 0{,}8w_1 & - & 2{,}8w_2 & - & 0{,}2x_3 & + & 1{,}8x_4 \\
x_2 & = & & & 0{,}4w_1 & - & 0{,}4w_2 & - & 0{,}6x_3 & + & 1{,}4x_4 \\
w_3 & = & 1 & - & 0{,}8w_1 & + & 2{,}8w_2 & + & 0{,}2x_3 & - & 1{,}8x_4
\end{array}
$$

U trećoj iteraciji x_3 ulazi i x_1 izlazi iz baze:

$$
\begin{array}{rclrrrrrrrr}
\zeta & = & & & 4w_1 & - & 16w_2 & - & 5x_1 & + & 6x_4 \\
\hline
x_3 & = & & & 4w_1 & - & 14w_2 & - & 5x_1 & + & 9x_4 \\
x_2 & = & & - & 2w_1 & + & 8w_2 & + & 3x_1 & - & 4x_4 \\
w_3 & = & 1 & & & & & - & x_1. & &
\end{array}
$$

Četvrta iteracija ima pivot (x_4, x_2):

$$
\begin{array}{rclrrrrrrrr}
\zeta & = & & & w_1 & - & 4w_2 & - & 0{,}5x_1 & - & 1{,}5x_2 \\
\hline
x_3 & = & & - & 0{,}5w_1 & + & 4w_2 & + & 1{,}75x_1 & - & 2{,}25x_2 \\
x_4 & = & & - & 0{,}5w_1 & + & 2w_2 & + & 0{,}75x_1 & - & 0{,}25x_2 \\
w_3 & = & 1 & & & & & - & x_1. & &
\end{array}
$$

U petoj iteraciji pivot je (w_1, x_3):

$$
\begin{array}{rclrrrrrrrr}
\zeta & = & & - & 2x_3 & + & 4w_2 & + & 3x_1 & - & 6x_2 \\
\hline
w_1 & = & & - & 2x_3 & + & 8w_2 & + & 3{,}5x_1 & - & 4{,}5x_2 \\
x_4 & = & & & x_3 & - & 2w_2 & - & x_1 & + & 2x_2 \\
w_3 & = & 1 & & & & & - & x_1. & &
\end{array}
$$

Poslednja, šesta iteracija ima pivot (w_2, x_4):

$$
\begin{array}{rclcrcrcrcr}
\zeta & = & & & & - & 2x_4 & + & 3x_1 & - & 2x_2 \\
\hline
w_1 & = & & & 2x_3 & - & 4x_4 & - & 0{,}5x_1 & + & 3{,}5x_2 \\
w_2 & = & & & 0{,}5x_3 & - & 0{,}5x_4 & - & 0{,}5x_1 & + & x_2 \\
w_3 & = & 1 & & & & & - & x_1. & &
\end{array}
$$

Primetimo da smo se posle niza pivotizacija vratili u početni rečnik i odavde pa na dalje simpleks metoda kruži kroz ovih šest rečnika i nikada ne napravi napredak ka optimalnom rešenju.

Primetimo da ukoliko simpleks metoda kruži kroz rečnike, onda pivoti koji se javljaju moraju biti degenerisani. Ovo se uočava na osnovu činjenice da se vrednost funkcije cilja ne povećava. Stoga sledi da svi pivoti u kružnim rečnicima moraju voditi ka istoj vrednosti funkcije cilja, tj. moraju biti degenerisani.

Kruženje u rečnicima se može pojaviti i važno je da se spreči. **Blandovo pravilo** pivotizacije sprečava se dođe do kruženja u rečnicima i ono glasi: Pri izboru promenljive koja će ući u bazu, birati promenljivu sa pozitivnim koeficijentom $\bar{c}_j$ koja ima najmanji redni broj. Za izbor promenljive koja će izaći iz baze među bazičnim promenljivama koje daju minimalni količnik $\bar{b}_i/\bar{a}_{ik}, \bar{a}_{ik} > 0$, izabrati promenljivu sa najmanjim rednim brojem.

1.5 Dualnost

U vezi sa svakim linearnim programom se uvodi novi linearni program koji se zove *dual*. Dual duala je originalni problem (koji se naziva primar). Stoga, linearni programi dolaze u parovima primar/dual. Ispostaviće se da svako dopustivo rešenje za bilo koji od ova dva problema daje gornju granicu optimalne vrednosti funkcije cilja onog drugog problema. Ove ideje su važne i predmet su ovog poglavlja.

1.5.1 Motivacija: pronalaženje gornjih granica

Počećemo sa primerom:

$$\begin{array}{lrcrcrcl}
\text{maksimiziraj} & 4x_1 & + & x_2 & + & 3x_3 & & \\
\text{pod uslovima} & x_1 & + & 4x_2 & & & \leq & 1 \\
 & 3x_1 & - & x_2 & + & x_3 & \leq & 3 \\
 & & & \multicolumn{3}{c}{x_1, x_2, x_3 \geq 0.} & &
\end{array}$$

Naše prvo zapažanje je da svako dopustivo rešenje daje donju granicu optimalne vrednosti funkcije cilja ζ^*. Na primer, uređena trojka $(x_1, x_2, x_3) = (1, 0, 0)$ nam govori da je $\zeta^* \geq 4$. Koristeći dopustivo rešenje $(x_1, x_2, x_3) = (0, 0, 3)$ vidimo da $\zeta^* \geq 9$. Koliko su dobre dobijene granice? Jesmo li blizu optimalnog rešenja? Da bismo dobili odgovor na ova pitanja treba nam gornja granica optimalne vrednosti funkcije cilja koju možemo dobiti na sledeći način. Pomnožimo prvu jednačinu sa 2 i drugu jednačinu sa 3 i saberimo ih:

$$\begin{array}{rrcrcrcr}
2 & (x_1 & + & 4x_2 & &) & \leq & (2)1 \\
+3 & (3x_1 & - & x_2 & + & x_3) & \leq & (3)3 \\
\hline
 & 11x_1 & + & 5x_2 & + & 3x_3 & \leq & 11.
\end{array}$$

Kako je svaka promenljiva nenegativna, možemo uporediti dobijenu sumu sa vrednošću funkcije cilja i primetiti da

$$4x_1 + x_2 + 3x_3 \leq 11x_1 + 5x_2 + 3x_3 \leq 11.$$

Stoga je $\zeta^* \leq 11$. Lokalizovali smo pretragu negde između 9 i 11. Ove vrednosti ostavljaju prostor u kom optimalno rešenje leži, ali su one bolje nego ništa. Dalje, one se mogu popraviti. Da bismo dobili bolju gornju granicu primenićemo istu tehniku tako što ćemo umesto konkretnih brojeva koristiti promenljive a zatim pokušati da nađemo vrednosti tih promenljivih koje nam daju najbolju gornju granicu. Pomnožićemo ograničenja sa nenegativnim brojevima y_1 i y_2, redom. Činjenica da su ove promenljive nenegativne garantuje da će se znaci nejednakosti očuvati. Dobijamo:

$$\begin{array}{rrcrcrcr}
y_1 & (x_1 & + & 4x_2 & &) & \leq & y_1 \\
+y_2 & (3x_1 & - & x_2 & + & x_3) & \leq & 3y_2 \\
\hline
 & (y_1 + 3y_2)x_1 & + & (4y_1 - y_2)x_2 & + & (y_2)x_3 & \leq & y_1 + 3y_2.
\end{array}$$

Ako nametnemo uslov da svaki koeficijent uz x_i bude bar koliko su odgovarajući koeficijenti u funkciji cilja,

$$\begin{array}{rcrcl} y_1 & + & 3y_2 & \geq & 4 \\ 4y_1 & - & y_2 & \geq & 1 \\ & & y_2 & \geq & 3, \end{array}$$

možemo uporediti vrednost funkcije cilja sa dobijenom sumom (i njenom gornjom granicom):

$$\begin{array}{rcl} \zeta & = & 4x_1 + x_2 + 3x_3 \\ & \leq & (y_1 + 3y_2)x_1 + (4y_1 - y_2)x_2 + (y_2)x_3 \\ & \leq & y_1 + 3y_2. \end{array}$$

Sada imamo gornju granicu $y_1 + 3y_2$ koju treba da minimiziramo da bismo bismo dobili najbolju gornju granicu. Stoga smo prirodno došli do novog optimizacionog problema:

$$\begin{array}{lrcrcl} \text{minimiziraj} & y_1 & + & 3y_2 & & \\ \text{pod uslovima} & y_1 & + & 3y_2 & \geq & 4 \\ & 4y_1 & - & y_2 & \geq & 1 \\ & & & y_2 & \geq & 3 \\ & & & y_1, y_2 \geq 0. & & \end{array}$$

Dobijeni problem se naziva dual i u narednom odeljku ćemo definisati dual u opštem slučaju.

1.5.2 Dual

Za dati problem linearnog programiranja u standardnom obliku:

$$\begin{array}{llcll} \text{maksimiziraj} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{pod uslovima} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & i = 1, 2, \ldots, m \\ & x_j & \geq & 0 & j = 1, 2, \ldots, n, \end{array}$$

dodeljeni dualni problem je:

$$\begin{array}{llcll} \text{minimiziraj} & \sum_{i=1}^{m} b_i y_i & & & \\ \text{pod uslovima} & \sum_{i=1}^{m} a_{ij} y_i & \geq & c_j & j = 1, 2, \ldots, n \\ & y_i & \geq & 0 & i = 1, 2, \ldots, m. \end{array}$$

Problem od kog smo pošli se naziva primar. Naš prvi cilj je da pokažemo da dual duala je primar. Da bismo to postigli moramo prvo dovesti dual na standardni oblik, tj., moramo minimizaciju prevesti u maksimizaciju i ograničenja napisati u obliku manje ili jednako. Kako je maksimizacija neke fukcije jednaka minimizaciji negirane te iste funkcije, dobijamo:

$$\min \sum_{i=1}^{m} b_i y_i = -\max\left(-\sum_{i=1}^{m} b_i y_i\right).$$

Da bismo obrnuli znak nejednakosti, pomnožićemo ih sa -1. Dobijeni standardni oblik dualnog problema glasi:

$$\begin{array}{lrcll}
-\text{maksimiziraj} & \sum_{i=1}^{m}(-b_i)y_i & & & \\
\text{pod uslovima} & \sum_{i=1}^{m}(-a_{ij})y_i & \leq & (-c_j) & j=1,2,\ldots,n \\
 & y_i & \geq & 0 & i=1,2,\ldots,m.
\end{array}$$

Napravimo sada njegov dual:

$$\begin{array}{lrcll}
-\text{minimiziraj} & \sum_{j=1}^{n}(-c_j)x_j & & & \\
\text{pod uslovima} & \sum_{j=1}^{n}(-a_{ij})x_j & \geq & (-b_i) & i=1,2,\ldots,m \\
 & x_j & \geq & 0 & j=1,2,\ldots,n,
\end{array}$$

što je očigledno isto što i problem od kog smo krenuli.

1.5.3 Teorema slabe dualnosti

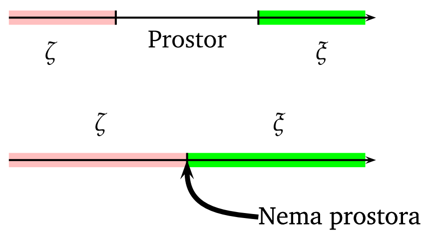
Kao što smo videli u našem primeru, dual nam daje gornju granicu optimalne vrednosti funkcije cilja. Ovo je poznat rezultat kao *Teorema slabe dualnosti*:

Teorema: Za dopustive vrednosti primara $(x_1,\ldots,x_n)$ i duala $(y_1,\ldots,y_m)$ važi

$$\zeta = \sum_{j=1}^{n} c_j\, x_j \leq \sum_{i=1}^{m} y_i\, b_i = \xi\,. \tag{1.8}$$

Dokaz: Dokaz je jednostavan niz nejednakosti:

$$\begin{aligned}
\sum_{j=1}^{n} c_j x_j &\leq \sum_{j=1}^{n}\left(\sum_{i=1}^{m} y_i a_{ij}\right) x_j \\
&= \sum_{j=1}^{n}\sum_{i=1}^{m} y_i a_{ij} x_j \\
&= \sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij} x_j\right) y_i \\
&\leq \sum_{i=1}^{m} y_i b_i.
\end{aligned}$$



Slika 1.2: Vrednosti funkcije cilja primara su manje ili jednake vrednostima funkcije cilja duala. Važno pitanje je da li postoji prostor između najveće vrednosti primara i najmanje vrednosti duala.

Zamislimo interval (podskup skupa $\mathbb{R}$) na realnoj osi koji sadrži sve moguće vrednosti funkcije cilja primara, i zamislimo isti takav skup za dualni problem. Teorema slabe dualnosti nam kaže da skup vrednosti primara leži levo od skupa vrednosti duala (Slika (**??**)). Uskoro ćemo videti da su ti skupovi zatvoreni intervali i da desna strana intervala vezanog za primar se poklapa sa levom stranom intervala vezanog za dual. Drugim rečima ne postoji razmak između optimalnih vrednosti primara i duala. Nepostojanje razmaka između između optimalnih vrednosti primara i duala predstavlja zgodan način za proveru optimalnosti. Zaista, ako možemo da nađemo dopustivo rešenje primara $(x_1^*, x_2^*, \dots, x_n^*)$ i dopustivo rešenje duala $(y_1^*, y_2^*, \dots, y_m^*)$ za koje važi

$$\sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} y_i^* b_i,$$

onda možemo zaključiti da je svako od ovih rešenja optimalno za svoj problem. Da bismo videli da je rešenje primara optimalno, zamislimo da imamo drugo dopustivo rešenje $(x_1, x_2, \dots, x_n)$. Na osnovu teoreme o slaboj dualnosti, imamo da je:

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{i=1}^{m} y_i^* b_i = \sum_{j=1}^{n} c_j x_j^*.$$

Pošto je $(x_1^*, x_2^*, \dots, x_n^*)$ dopustivo, vidimo da mora biti i optimalno. Slični argumenti se mogu navesti u prilog tome da je dualno rešenje ujedno i optimalno rešenje duala. U našem primeru, posmatrajmo rešenja $x = (0, 0, 25, 3, 25)$ i $y = (1, 3)$. Oba ova rešenja su dopustiva i za oba funkcije cilja imaju vrednost 10. Teorema o slaboj dualnosti kaže da su ova rešenja ujedno i optimalna.

1.5.4 Teorema jake dualnosti

Činjenica da za probleme linearnog programiranja nikada ne postoji prostor izmađu optimalnih vrednosti primara i duala je formulisana u vidu Teoreme o jakoj dualnosti:

Teorema: Ako primar ima optimalno rešenje $(x_1^*, \ldots, x_n^*)$, onda i dual ima optimalno rešenje $(y_1^*, \ldots, y_m^*)$ i važi

$$\zeta^* = \sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} y_i^* b_i = \xi^*. \tag{1.9}$$

Dokaz teoreme jake dualnosti ćemo ilustrovati na primeru. Glavna ideja koju želimo da pokažemo je da kako simpleks metoda napreduje u rešavanju primara, tako ona implicitno rešava i dual.

Vratimo se na naš primer. Da bismo primenili simpleks metodu, uvodimo pomoćne promenljive $w_i, i = 1, 2$ za primar i pomoćne dualne promenljive za dual $z_j, j = 1, 2, 3$. Kako su nejednakosti u dualu oblika veće ili jednako, svaka pomoćna promenljiva duala je definisana kao leva minus desna strana nejednakosti. Na primer,

$$z_1 = y_1 + 3y_2 - 4.$$

Početni rečnici primara i duala su:

$$\begin{array}{rcrcrcrcr}
\zeta & = & & & 4x_1 & + & x_2 & + & 3x_3 \\
\hline
w_1 & = & 1 & - & x_1 & - & 4x_2 & & \\
w_2 & = & 3 & - & 3x_1 & + & x_2 & - & x_3
\end{array}$$

i

$$\begin{array}{rcrcrcr}
-\xi & = & & & -y_1 & - & 3y_2 \\
\hline
z_1 & = & -4 & + & y_1 & + & 3y_2 \\
z_2 & = & -1 & + & 4y_1 & - & y_2 \\
z_3 & = & -3 & & & + & y_2
\end{array}$$

Primetimo da posmatramo negativnu vrednost funkcije cilja duala jer preferiramo da se problem maksimizacije javlja u rečniku. Takođe, primetimo da su koeficijenti u rečniku duala negativne vrednosti koeficijenata u rečniku primara s tim da su kolone i vrste zamenile mesta. Zaista, ako uklonimo iz rečnika sve sem koeficijenata, imamo:

$$\begin{bmatrix} 0 & 4 & 1 & 3 \\ 1 & -1 & -4 & 0 \\ 3 & -3 & 1 & -1 \end{bmatrix} \overset{\text{neg. tr.}}{\longleftarrow - - \longrightarrow} \begin{bmatrix} 0 & -1 & -3 \\ -4 & 1 & 3 \\ -1 & 4 & -1 \\ -3 & 0 & 1 \end{bmatrix}$$

Dakle, dualni rečnik je **negativno transponovan** primarni rečnik.

Naš cilj je da primenimo simpleks metodu na primaru i da u isto vreme vršimo analogne pivotizacije na dualu. Trebalo bi da otkrijemo da osobina negativnog transponovanje će se očuvati kroz iteracije.

Kako je rečnik primara dopustiv, ne moramo da primenjujemo dvofazni algoritam. U prvoj pivotizaciji uzimamo x_3 za ulaznu promenljivu (bez obzira što x_1 ima najveći koeficijent, uzimamo x_3 jer će nas najbrže odvesti do optimalnog rešenja). Ovakav izbor ulazne promenljive implicira da izlazna promenljiva mora biti w_2. Kako su vrste i kolone zamenile mesta u dualnom rečniku, vidimo da kolona promenljive x_3 u rečniku primara odgovara vrsti promenljive z_3 u rečniku duala. Da bismo izvršili analognu pivotizaciju u dualnom rečniku, odabraćemo promenljivu y_2 za ulaznu promenljivu i z_3 za izlaznu promenljivu. Izbor ulazne i izlazne promenljive je neobičan u odnosu na to kako smo ove promenljive birali do sada; primetimo da je do sada naš izbor ovih promenljivih bio vođen željom da povećamo vrednost funkcije cilja a da pri tom očuvamo dopustivost. Ovde dualni rečnik nije dopustiv i takvo razmatranje ni nema smisla. Jednom kada napustimo dosadašnja pravila za izbor ulazne i izlazne promenljive, lako je uvideti da pivotizacija može biti izvedena sa bilo kojim izborom ulazne i izlazne promenljive koji garantuje da koeficijenti uz ulaznu promenljivu u uslovu za izlaznu promenljivu neće nestati. Ovo je slučaj sa trenutnim izborom. Stoga sprovodimo pivotizaciju primara (x_3, w_2) i duala (y_2, z_3) i dobijamo

$$
\begin{array}{rcrcrcrcr}
\zeta & = & 9 & - & 5x_1 & + & 4x_2 & - & 3w_2 \\
\hline
w_1 & = & 1 & - & x_1 & - & 4x_2 & & \\
x_3 & = & 3 & - & 3x_1 & + & x_2 & - & w_2
\end{array}
$$

$$
\begin{array}{rcrcrcr}
-\xi & = & -9 & - & y_1 & - & 3z_3 \\
\hline
z_1 & = & 5 & + & y_1 & + & 3z_3 \\
z_2 & = & -4 & + & 4y_1 & - & z_3 \\
y_2 & = & 3 & & & + & z_3.
\end{array}
$$

Primetimo da rečnici i dalje imaju osobinu negativnog transponovanja. Za sledeću pivotizacija primara uzimamo par promenljivih (x_2, w_1) i za dual par promenljivih (y_1, z_2). Pivotizacija daje:

$$
\begin{array}{rcrcrcrcr}
\zeta & = & 10 & - & 6x_1 & - & w_1 & - & 3w_2 \\
\hline
x_2 & = & \frac{1}{4} & - & \frac{1}{4}x_1 & - & \frac{1}{4}w_1 & & \\
x_3 & = & \frac{13}{4} & - & \frac{13}{4}x_1 & - & \frac{1}{4}w_1 & - & w_2
\end{array}
$$

i

$$
\begin{array}{rcrcrcr}
-\xi & = & -10 & - & \frac{1}{4}z_2 & - & \frac{13}{4}z_3 \\
\hline
z_1 & = & 6 & + & \frac{1}{4}z_2 & + & \frac{13}{4}z_3 \\
y_1 & = & 1 & + & \frac{1}{4}z_2 & + & \frac{1}{4}z_3 \\
y_2 & = & 3 & & & + & z_3.
\end{array}
$$

Rečnik primara je optimalan jer su svi koeficijenti u funkciji cilja nenegativni. Posmatrajući rečnik duala, vidimo da je on sada dopustiv. U suštini, on je takođe optimalan. Konačno vrednosti funkcija cilja primara i duala su jednake i iznose 10. Situacija sada postaje jasna. Za dobijeni problem linearnog programiranja za koji pretpostavljamo da ima optimalno rešenje, prvo primeni Fazu 1 da dobiješ dopustivo rešenje za Fazu 2. Zatim primeni simpleks metodu da pronađeš optimalno rešenje. Svaki rečnik primara dobijen primenom simpleks metode implicitno definiše odgovarajući rečnik duala na sledeći način: negativno transponuj rečnik i umesto promenljivih x_j napiši z_j a umesto promenljivih

w_i napiši y_i. Pivotizaciji primara (x_j, w_i) odgovara pivotizacija duala (y_i, z_j). Dok god rečnik primara nije optimalan, dotle implicitno definisan rečnik duala neće biti dopustiv. Ali kada se pronađe optimalno rešenje primara, odgovarajući rečnik duala postaje dopustiv. Kako su koeficijenti u funkciji cilja uvek nenegativni, ovaj rečnik duala je ujedno i optimalan. Takođe, u svakoj iteraciji trenutna vrednost funkcije cilja primara i duala je jednaka.

Formalni dokaz teoreme se može naći u knjizi.

Teorema o jakoj dualnosti nam govori, da kad god primar ima optimalno rešenje, dual takođe ima optimalno rešenje i ta rešenja se poklapaju (nema prostora između njih). Ali šta ako primar nema optimalno rešenje? Pretpostavimo da je primar neograničen. Neograničenost primara zajedno sa teoremom o slaboj dualnosti nam govori da dualni problem mora biti nemoguć. Slično, ako je dualni problem neograničen, onda je primar nemoguć.

1.5.5 Komplementarnost dualnih promenljivih

Nekada je neophodno zaključiti šta je optimalno rešenje duala kada je jedino optimalno rešenje primara poznato. Sledeća teorema poznata kao Komplementarnost dualnih promenljivih pomaže u tom slučaju.

Teorema: Neka je $(x_1, \ldots, x_n)$ dopustivo rešenje primara i neka je $(y_1, \ldots, y_m)$ dopustivo rešenje duala. Neka su $(w_1, \ldots, w_m)$ odgovarajuće dodatne promenljive primara i $(z_1, \ldots, z_n)$ odgovarajuće dodatne promenljive duala.

$(x_1, \ldots, x_n)$ je optimalno rešenje primara i $(y_1, \ldots, y_m)$ je optimalno rešenje duala ako i samo ako

$$\begin{aligned} x_j\, z_j &= 0 \text{ za } j = 1, 2, \ldots, n \\ y_i\, w_i &= 0 \text{ za } i = 1, 2, \ldots, m. \end{aligned} \tag{1.10}$$

Dokaz:

Iskaz $x_j\, z_j = 0$ zapravo tvrdi da se barem jedna od x_j i z_j anulira za svako j. Pođimo od dokaza teoreme slabe dualnosti. Teorema jake dualnosti kaže da za optimalno rešenje važi jednakost $\zeta = \xi$.

$$\zeta = \sum_{j=1}^{n} c_j x_j \quad \leq \quad \sum_{j=1}^{n} \left(\sum_{i=1}^{m} y_i a_{i,j} \right) x_j \ = \tag{1.11}$$

$$= \quad \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{i,j} x_j \right) y_i \ \leq$$

$$\leq \quad \sum_{i=1}^{m} b_i y_i = \xi \tag{1.12}$$

Zbog nenegativnosti x_j sledi da u (1.11) imamo jednakost akko

$$x_j = 0 \ \text{ ili } \ c_j = \sum_{i=1}^{m} y_i a_{i,j} \Leftrightarrow z_j = 0, \text{ za } j = 1, 2, \ldots, n.$$

Zbog nenegativnosti y_i sledi da u (1.12) imamo jednakost akko

$$y_i = 0 \;\; \text{ili} \;\; b_i = \sum_{j=1}^{n} a_{i,j} x_j \;\Leftrightarrow\; w_i = 0, \text{ za } i = 1,2,\ldots,m.$$

1.5.6 Dualna simpleks metoda

U ovom odeljku ćemo razmatrati šta se dešava ako primenimo simpleks metodu na dualni problem. Kao što smo videli u diskusiji o teoremi stroge dualnosti, simpleks metoda se može primeniti direktno na dualni problem i bez zapisivanja dualnog problema i njegovih rečnika. Dualna simpleks metoda se može posmatrati kao novi način odabira ulaznih i izlaznih promenljivih u nizu primarnih rečnika.

Počećemo sa primerom:

$$
\begin{array}{lrcrcr}
\text{maksimiziraj} & -x_1 & - & x_2 & & \\
\text{pod uslovima} & -2x_1 & - & x_2 & \leq & 4 \\
 & -2x_1 & + & 4x_2 & \leq & -8 \\
 & -x_1 & + & 3x_2 & \leq & -7 \\
 & & x_1, x_2 \geq 0. & & &
\end{array}
$$

Dualni problem je:

$$
\begin{array}{lrcrcrcr}
\text{minimiziraj} & 4y_1 & - & 8y_2 & - & 7y_3 & & \\
\text{pod uslovima} & -2y_1 & - & 2y_2 & - & y_3 & \geq & -1 \\
 & -y_1 & + & 4y_2 & + & 3y_3 & \geq & -1 \\
 & & & y_1, y_2, y_3 \geq 0. & & & &
\end{array}
$$

Uvođenjem pomoćnih promenljivih $w_i, i = 1,2,3$ za primar i $z_j, j = 1,2$ za dual, možemo napisati početne rečnike:

$$
\begin{array}{rcrcrcr}
\zeta & = & & - & x_1 & - & x_2 \\
\hline
w_1 & = & 4 & + & 2x_1 & + & x_2 \\
w_2 & = & -8 & + & 2x_1 & - & 4x_2 \\
w_3 & = & -7 & + & x_1 & - & 3x_2
\end{array}
$$

i

$$
\begin{array}{rcrcrcrcr}
-\xi & = & & - & 4y_1 & + & 8y_2 & + & 7y_3 \\
\hline
z_1 & = & 1 & - & 2y_1 & - & 2y_2 & - & y_3 \\
z_2 & = & 1 & - & y_1 & + & 4y_2 & + & 3y_3
\end{array}
$$

Kao što smo i ranije radili, posmatramot negativne vrednosti dualne funkcije cilja jer preferiramo da funkcija cilja koja se javlja u rečniku se maksimizira. Primetimo da je dualni rečnik dopustiv a da primarni nije. To sugeriše da bi bilo smisleno primeniti simpleks metodu na dualu. Uradićemo to ali ćemo ujedno analogne pivotizacije sprovesti i na primaru. Na primer, ulazna promenljiva u početnom rečniku duala je y_2 i izlazna promenljiva je z_1. Kako je w_2 komplementarna promenljiva promenljivoj y_2 i x_1 je komplementarna promenljiva z_1, koristićemo (w_2, x_1) za pivot u rečniku primara. Naravno, kako je w_2 bazična i x_1 nebazična, w_2 mora biti izlazna promenljiva a x_1 ulazna, obrnuto od onoga što imamo za komplementarne promenljive u dualnom rečniku. Rezultat ovakve pivotizacije je:

$$
\begin{array}{rcrcrcr}
\zeta & = & -4 & - & 0{,}5w_2 & - & 3x_2 \\
\hline
w_1 & = & 12 & + & w_2 & + & 5x_2 \\
x_1 & = & 4 & + & 0{,}5w_2 & + & 2x_2 \\
w_3 & = & -3 & + & 0{,}5w_2 & - & x_2
\end{array}
$$

i

$$
\begin{array}{rcrcrcrcr}
-\xi & = & 4 & - & 12y_1 & - & 4z_1 & + & 3y_3 \\
\hline
y_2 & = & 0{,}5 & - & y_1 & - & 0{,}5z_1 & - & 0{,}5y_3 \\
z_2 & = & 3 & - & 5y_1 & - & 2z_1 & + & y_3
\end{array}
$$

Nastavljamo da radimo na dualu, imamo u sledećem koraku da je y_3 ulazna a y_2 izlazna promenljiva. Stoga, za primar uzimamo da je w_3 izlazna promenljiva i w_2 ulazna promenljiva. Posle pivotizacije imamo:

$$
\begin{array}{rcrcrcr}
\zeta & = & -7 & - & w_3 & - & 4x_2 \\
\hline
w_1 & = & 18 & + & 2w_3 & + & 7x_2 \\
x_1 & = & 7 & + & w_3 & + & 3x_2 \\
w_2 & = & 6 & + & 2w_3 & + & 2x_2
\end{array}
$$

i

$$
\begin{array}{rcrcrcrcr}
-\xi & = & 7 & - & 18y_1 & - & 7z_1 & - & 6y_2 \\
\hline
y_3 & = & 1 & - & 2y_1 & - & z_1 & - & 2y_2 \\
z_2 & = & 4 & - & 7y_1 & - & 3z_1 & - & 2y_2.
\end{array}
$$

Primetimo sada da su dobijeni rečnici optimalni. Takođe, u svakom rečniku imamo očuvanu osobinu negativnog transponovanja. Stoga mi nikada ne moramo da zapisujemo rečnike duala, dualna simpleks metoda se može kompletno sprovesti i opisati na rečnicima primara. Primetimo prvo da rečnik mora biti dualno dopustiv. To znači da svi koeficijenti nebazičnih promenljivih u primaru moraju biti nepozitivni. Imajući to na umu, sprovodimo proceduru. Prvo biramo izlaznu promenljivu birajući onu baziču promenljivu čije $\bar{b}_i$ je najnegativnije (ako nema takvih onda je trenutni rečnik optimalan). Zatim biramo ulaznu promenljivu skenirajući u posmatranoj vrsti rečnika količnike koeficijenata iz vrste u funkciji cilja i posmatrane vrste, gledajući najnegativniji količnik kao što smo radili i u primarnoj simpleks metodi, tj. biramo x_k ulaznu promenljivu kao promenljivu za koju se ostvaruje minimum:

$$x_k := \min_{j \in \mathcal{N} : \bar{a}_{l,j} < 0} -\frac{\bar{c}_j}{\bar{a}_{l,j}}. \tag{1.13}$$

Jednom kad smo utvrdili ulaznu i izlaznu promenljivu, vršimo pivotizaciju i nastavljamo dalje.

1.5.7 Dualna simpleks metoda - tabelarni prikaz

U ovom odeljku ćemo prikazati tabelarno kako dualna simpleks metoda napreduje kroz iteracije. Inicijalna tabela je data sa:

	x_1	x_2	w_1	w_2	w_3	
w_1	-2	-1	1	0	0	4
w_2	-2	4	0	1	0	-8
w_3	-1	3	0	0	1	-7
ζ	1	1	0	0	0	0

Prva pivotizacija se vrši tako što prvo biramo promenljivu koja će izaći iz baze. To je promenljiva w_2 jer je njena vrednost u rečničkom rešenju najmanja (to je -8). Zatim biramo promenljivu koja će ući u bazu tako što posmatramo u vrsti izlazne promenljive w_2 strogo negativne koeficijente i pravimo negativan količnik koeficijenata iz funkcije cilja i tih negativnih koeficijenata. Za ovaj konekretni primer, jedini negativni koeficijent u posmatranoj vrsti je -2 a odgovarajući količnik iznosi $-\frac{1}{-2} = \frac{1}{2}$. Od dobijenih količnika (u ovom primeru imamo jedan) najmanji određuje koja promenljiva će ući u bazu. U našem primeru je to promenljiva x_1. Nakon pivotizacije nova tabela je data sa:

	x_1	x_2	w_1	w_2	w_3	
w_1	0	-5	1	-1	0	12
x_1	1	-2	0	$-0{,}5$	0	4
w_3	0	1	0	$-0{,}5$	1	-3
ζ	0	3	0	$0{,}5$	0	-4

Sada iz baze izlazi promenljiva ω_3 jer je njena vrednost najmanja (-3), a ulazi promenljiva w_2 (jer minimalni negativni količnik od koeficijenata iz funkcije cilja i negativnih koeficijenata u posmatranoj vrsti je jedan jedini i iznosi $-\frac{0{,}5}{-0{,}5} = 1$). Nakon pivotizacije dobijamo tabelu koja je i primarno i dualno dopustiva tj. dobijamo optimalnu tabelu.

	x_1	x_2	w_1	w_2	w_3	
w_1	0	-7	1	0	-2	18
x_1	1	-3	0	0	-1	7
w_2	0	-2	0	1	-2	6
ζ	0	4	0	0	1	-7

1.5.8 Inicijalizacija za Dualni simplex algoritam

Ako se dobije problem kod kojeg izbor dodatnih promenljivih za bazične ne daje dualno dopustiv rečnik, može se prvo rešiti Dualnim simplex algoritmom **novi problem** sa istim ograničenjima i funkcijom dobiti koja daje dualno dopustiv rečnik, recimo: $-\eta = x_1 + \cdots + x_n \to \min$.

Dual za novi problem sigurno nije nemoguć. Jedini slučaj da za novi problem Simplex algoritam ne nađe rešenje je da je dual neograničen.

Ali, tada je na osnovu slabe teoreme dualnosti primar nemoguć.

Kad se dobije rešenje novog problema, dobijeno je dopustivo rešenje početnog primara, prelazi se na njegovo rešavanje Simplex algoritmom sa originalnom funkcijom dobiti ζ.

Primer: Posmatrajmo problem:

$$\begin{array}{rcrcrcc} \zeta & = & 2x_1 & - & x_2 & \to & \max \\ & & x_1 & - & x_2 & \leq & -1 \\ & & 2\,x_1 & + & x_2 & \leq & 3 \\ & & x_1 & + & 2\,x_2 & \leq & 4 \\ & & & & \multicolumn{3}{l}{x_1 \geq 0 \quad x_2 \geq 0} \end{array} \tag{1.14}$$

Dati problem nije primarno dopustiv (jer imamo -1 na desnoj strani ograničenja), ujedno nije ni dualno dopustiv (koeficijenti u funkciji cilja nisu svi negativni).

I način

Jedan način da ga rešimo je kao što smo već pokazali, uvođenjem pomoćnog problema i primenom primarnog simpleks algoritma. Dovodimo na standardni oblik sa $=$, postavljamo pomoćni problem:

$$\begin{array}{rcrcrcrcrcrcrcc} \zeta_1 & = & & & & & & & & & & - & x_0 & \to & \max \\ \zeta & = & 2x_1 & - & x_2 & & & & & & & & & \to & \max \\ & & x_1 & - & x_2 & + & w_1 & & & & & - & x_0 & = & -1 \\ & & 2\,x_1 & + & x_2 & & & + & w_2 & & & - & x_0 & = & 3 \\ & & x_1 & + & 2\,x_2 & & & & & + & w_3 & - & x_0 & = & 4 \end{array}$$

uz zahteve nenegativnosti: $x_1 \geq 0, x_2 \geq 0, w_1 \geq 0, w_2 \geq 0, w_3 \geq 0, x_0 \geq 0$.

Početna tabela koja odgovara problemu je:

	x_1	x_2	w_1	w_2	w_3	x_0	
w_1	1	-1	1	0	0	$-\mathbf{1}$	-1
w_2	2	1	0	1	0	-1	3
w_3	1	2	0	0	1	-1	4
ζ	-2	1	0	0	0	0	0
ζ_1	0	0	0	0	0	1	0

U prvom koraku x_0 ulazi u bazu umesto w_1 (-1 je najnegativniji broj sa desne strane i on određuje da w_1 izlazi iz baze). Tako se dobija tabela:

	x_1	x_2	w_1	w_2	w_3	x_0	
x_0	-1	$\mathbf{1}$	-1	0	0	1	1
w_2	1	2	-1	1	0	0	4
w_3	0	3	-1	0	1	0	5
	-2	1	0	0	0	0	0
	1	-1	1	0	0	0	-1

Zatim nastavljamo sa pivotizacijama posmatrajući funkciju cilja pomoćnog problema. U bazu ulazi x_2 (jer je njen koeficijent u funkciji cilja pomoćnog problema najnegativniji i iznosi -1) umesto x_0 (minimalni količnik se u datoj koloni ostvaruje u prvoj vrsti). Nakon pivotizacije se dobija tabela:

	x_1	x_2	w_1	w_2	w_3	
x_2	-1	1	-1	0	0	1
w_2	**3**	0	1	1	0	2
w_3	3	0	2	0	1	2
	-1	0	1	0	0	-1

U poslednjoj tabeli je x_0 izašlo iz baze i time postalo 0 u rečničkom rešenju. Pošto $x_0 \geq 0$, time smo dobili rešenje pomoćnog problema i tu je kraj prve faze. Imamo dopustivu tabelu početnog problema. Stoga smo tabelu 1 pisali bez kolone x_0 i vrste ζ_1, da bismo nastavili drugu fazu. Posle pivotizacije dobijamo optimalnu tabelu:

	x_1	x_2	w_1	w_2	w_3	
x_2	0	1	$-2/3$	$1/3$	0	$5/3$
x_1	1	0	$1/3$	$1/3$	0	$2/3$
w_3	0	0	1	-1	1	0
	0	0	$4/3$	$1/3$	0	$-1/3$

Rešenje je: $x_1^* = \frac{2}{3}$, $x_2^* = \frac{5}{3}$, $\zeta^* = -\frac{1}{3}$.

II način

Drugi način da rešimo isti problem je pomoću dualnog simpleksa. Kako problem nije dualno dopustiv, mora se uvesti pomoćna funkcija cilja $-\eta = x_1 + x_2$ koju želimo da minimiziramo pod datim ograničenjima:

$$
\begin{array}{rcrcrcrcrcrcl}
\eta & = & -x_1 & - & x_2 & & & & & & & \rightarrow & \max \\
\zeta & = & 2x_1 & - & x_2 & & & & & & & \rightarrow & \max \\
& & x_1 & - & x_2 & + & w_1 & & & & & = & -1 \\
& & 2\,x_1 & + & x_2 & & & + & w_2 & & & = & 3 \\
& & x_1 & + & 2\,x_2 & & & & & + & w_3 & = & 4
\end{array}
$$

Polazna tabela za tako postavljen pomoćni problem je data sa:

	x_1	x_2	w_1	w_2	w_3	
w_1	1	-1	1	0	0	-1
w_2	2	1	0	1	0	3
w_3	1	2	0	0	1	4
ζ	-2	1	0	0	0	0
$-\eta$	1	1	0	0	0	0

Dualna simpleks metoda primenjena na pomoćni problem bira w_1 za izlaznu promenljivu a x_2 za ulaznu promenljivu. Nakon pivotizacije dobijamo:

	x_1	x_2	w_1	w_2	w_3	
x_2	-1	1	-1	0	0	1
w_2	3	0	1	1	0	2
w_3	3	0	2	0	1	2
ζ	-1	0	1	0	0	-1
$-\eta$	2	0	1	0	0	-1

Kako smo dobili primarno dopustivu tabelu za originalni problem, u nastavku nećemo više posmatrati pomoćni problem već nastavljamo da primenjujemo primarni simpleks na originalni problem:

	x_1	x_2	w_1	w_2	w_3	
x_2	-1	1	-1	0	0	1
w_2	3	0	1	1	0	2
w_3	3	0	2	0	1	2
ζ	-1	0	1	0	0	-1

U sledećoj iteraciji u bazu ulazi x_1 umesto w_2, te dobijamo optimalnu tabelu:

	x_1	x_2	w_1	w_2	w_3	
x_2	0	1	$-2/3$	$1/3$	0	$5/3$
x_1	1	0	$1/3$	$1/3$	0	$2/3$
w_3	0	0	1	-1	1	0
	0	0	$4/3$	$1/3$	0	$-1/3$

1.5.9 Dodatni primer

Posmatrajmo problem linearnog programiranja:

$$\begin{array}{rcrcrcrcl}
\zeta & = & 9x_1 & + & x_2 & + & x_3 & \to & \min \\
 & & 3x_1 & - & x_2 & + & 2x_3 & \geq & -1 \\
 & & 4x_1 & + & 2x_2 & - & x_3 & \geq & 5 \\
 & & \multicolumn{7}{c}{x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0}
\end{array}$$

Dovođenjem na standardni oblik imamo:

$$\begin{array}{rcrcrcrcl}
-\zeta & = & -9x_1 & - & x_2 & - & x_3 & \to & \max \\
 & & -3x_1 & + & x_2 & - & 2x_3 & \leq & 1 \\
 & & -4x_1 & - & 2x_2 & + & x_3 & \leq & -5 \\
 & & \multicolumn{7}{c}{x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0}
\end{array}$$

Dobijeni problem nije primarno dopusti ali jeste dualno dopustiv te ga je moguće rešiti pomoću dualnog simpleksa. Ako pak želimo da ga rešimo pomoću primarnog simpleksa, imamo dva načina - da uvedemo pomoćni problem $\zeta_1 = -x_0 \to \max$ ili pak da pređemo na dual i njega rešimo primarnim simpleksom. U nastavku ćemo prikazati sva tri načina:

I način - dualni simpleks

Polazna tabela je data sa:

	x_1	x_2	x_3	w_1	w_2	
w_1	-3	1	-2	1	0	-1
w_2	-4	-2	1	0	1	5
$-\zeta$	9	1	1	0	0	0

U prvoj iteraciji promenljiva w_2 izlazi iz baze a promenljiva x_2 ulazi u bazu:

	x_1	x_2	x_3	w_1	w_2	
w_1	-5	0	$-3/2$	1	$1/2$	$-3/2$
x_2	2	1	$-1/2$	0	$-1/2$	$5/2$
$-\zeta$	7	0	$3/2$	0	$1/2$	$-5/2$

U drugoj iteraciji w_1 izlazi iz baze a x_3 ulazi u bazu:

	x_1	x_2	x_3	w_1	w_2	
x_3	$10/3$	0	1	$-2/3$	$-1/3$	1
x_2	$11/3$	1	0	$-1/3$	$-2/3$	3
$-\zeta$	2	0	0	1	1	-4

Rešenje problema je $\zeta^* = 4$ i dobija se za $x_1^* = 0, x_2^* = 3, x_3^* = 1$.
II način - prelazak na dual i rešavanje pomoću primarnog simpleksa
Za primar:

$$\begin{array}{rcrcrcrcl} -\zeta & = & -9x_1 & - & x_2 & - & x_3 & \rightarrow & \max \\ & & -3x_1 & + & x_2 & - & 2x_3 & \leq & 1 \\ & & -4x_1 & - & 2x_2 & + & x_3 & \leq & -5 \\ & & \multicolumn{5}{c}{x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0} \end{array}$$

dual glasi:

$$\begin{array}{rcrcrcl} \xi & = & y_1 & - & 5y_2 & \rightarrow & \min \\ & & -3y_1 & - & 4y_2 & \geq & -9 \\ & & y_1 & - & 2y_2 & \geq & -1 \\ & & -2y_1 & + & y_2 & \geq & -1 \\ & & \multicolumn{3}{c}{y_1 \geq 0 \quad y_2 \geq 0} \end{array}$$

ili u standardnom obliku:

$$\begin{array}{rcrcrcl} -\xi & = & -y_1 & + & 5y_2 & \rightarrow & \max \\ & & 3y_1 & + & 4y_2 & \leq & 9 \\ & & -y_1 & + & 2y_2 & \leq & 1 \\ & & 2y_1 & - & y_2 & \leq & 1 \\ & & \multicolumn{3}{c}{y_1 \geq 0 \quad y_2 \geq 0} \end{array}$$

Kako je problem primarno dopustiv (svi brojevi sa desne strane ograničenja su nenegativni), možemo primeniti primarni simpleks. Uvodimo pomoćne promenljive:

$$\begin{array}{rcrcrcrcrcrcl} -\xi & = & -y_1 & + & 5y_2 & & & & & & & \rightarrow & \max \\ & & 3y_1 & + & 4y_2 & + & z_1 & & & & & = & 9 \\ & & -y_1 & + & 2y_2 & & & + & z_2 & & & = & 1 \\ & & 2y_1 & - & y_2 & & & & & + & z_3 & = & 1 \\ & & \multicolumn{9}{c}{y_1 \geq 0 \quad y_2 \geq 0 \quad z_1 \geq 0 \quad z_2 \geq 0 \quad z_3 \geq 0} \end{array}$$

Polazna tabela je:

	y_1	y_2	z_1	z_2	z_3	
z_1	3	4	1	0	0	9
z_2	-1	2	0	1	0	1
z_3	2	-1	0	0	1	1
ξ	1	-5	0	0	0	0

U prvoj iteraciji u bazu ulazi promenljiva y_2 a iz baze izlazi promenljiva z_2:

	y_1	y_2	z_1	z_2	z_3	
z_1	5	0	1	-2	0	7
y_2	$-1/2$	1	0	$1/2$	0	$1/2$
z_3	$3/2$	0	0	$1/2$	1	$3/2$
ξ	$-3/2$	0	0	$5/2$	0	$5/2$

U narednoj iteraciji pivot je (y_1, z_3) i dobijamo optimalnu tabelu:

	y_1	y_2	z_1	z_2	z_3	
z_1	0	0	1	$-11/3$	$-10/3$	2
y_2	0	1	0	$2/3$	$1/3$	1
y_1	1	0	0	$1/3$	$2/3$	1
ξ	0	0	0	3	1	4

Rešenje problema je $\zeta^* = \xi^* = 4$ i dobija se za $x_1^* = 0, x_2^* = 3, x_3^* = 1$ (vrednosti u tabeli ispod promenljivih z_1, z_2 i z_3).

II način - dvofazni primarni simpleks

Kako posmatrani problem nije primarno dopustiv

$$\begin{array}{rcrcrcrcl} -\zeta & = & -9x_1 & - & x_2 & - & x_3 & \to & \max \\ & & -3x_1 & + & x_2 & - & 2x_3 & \leq & 1 \\ & & -4x_1 & - & 2x_2 & + & x_3 & \leq & -5 \\ & & \multicolumn{7}{c}{x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0} \end{array}$$

uvodimo pomoćni problem:

$$\begin{array}{rcrcrcrcrcl} -\zeta & = & -9x_1 & - & x_2 & - & x_3 & & & \to & \max \\ \zeta_1 & = & & & & & & - & x_0 & \to & \max \\ & & -3x_1 & + & x_2 & - & 2x_3 & - & x_0 & \leq & 1 \\ & & -4x_1 & - & 2x_2 & + & x_3 & - & x_0 & \leq & -5 \\ & & \multicolumn{9}{c}{x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0 \quad x_0 \geq 0} \end{array}$$

Polazna tabela za dati problem je:

	x_1	x_2	x_3	w_1	w_2	x_0	
w_1	-3	1	-2	1	0	-1	1
w_2	-4	-2	1	0	1	-1	-5
$-\zeta$	9	1	1	0	0	0	0
ζ_1	0	0	0	0	0	1	0

U prvoj iteraciji u bazu ulazi x_0 umesto w_2:

	x_1	x_2	x_3	w_1	w_2	x_0	
w_1	1	3	-3	1	-1	0	6
x_0	4	2	-1	0	-1	1	5
$-\zeta$	9	1	1	0	0	0	0
ζ_1	-3	1	-2	1	-1	0	1

U drugoj iteraciji u bazu ulazi x_1 umesto x_0 čime se završava prva faza algoritma:

	x_1	x_2	x_3	w_1	w_2	
w_1	0	$5/2$	$-11/4$	1	$-3/4$	$19/4$
x_1	1	$1/2$	$-1/4$	0	$-1/4$	$5/4$
$-\zeta$	0	$-7/2$	$13/4$	0	$9/4$	$-45/4$

Sledeća pivotizacija (x_2, w_1) daje:

	x_1	x_2	x_3	w_1	w_2	
x_2	0	1	$-11/10$	$2/5$	$-3/10$	$19/10$
x_1	1	0	$3/10$	$-1/5$	$-1/10$	$3/10$
$-\zeta$	0	0	$-3/5$	$7/5$	$6/5$	$-46/10$

Poslednja pivotizacija (x_3, x_1) daje optimalnu tabelu:

	x_1	x_2	x_3	w_1	w_2	
x_2	$11/3$	1	0	$-1/3$	$-2/3$	3
x_3	$10/3$	0	1	$-2/3$	$-1/3$	1
$-\zeta$	2	0	0	1	1	-4

Rešenje problema je $\zeta^* = 4$ i dobija se za $x_1^* = 0, x_2^* = 3, x_3^* = 1$.

1.5.10 Duali problema u opštem obliku

Videli smo da problemi linearnog programiranja mogu biti formulisani na razne načine. U ovom odeljku ćemo izvesti kako dual izgleda kada primar nije nužno dat u standardnom obliku.

Za početak, posmatrajmo slučaj kada su ograničenja data u vidu jednakosti (i promenljive su nenegativne):

$$\zeta = \sum_{j=1}^{n} c_j\, x_j \to \max$$

$$\sum_{j=1}^{n} a_{i,j}\, x_j = b_i,\ i = 1,2,\ldots,m$$

$$x_j \geq 0,\ j = 1,2,\ldots,n.$$

Kao što smo već spomenuli, problem može biti zapisan tako da su ograničenja data u vidu nejednakosti, gde svaka jednakost se pretvara u dve nejednakosti ($\leq$ i $\geq$):

$$\zeta = \sum_{j=1}^{n} c_j\, x_j \to \max$$
$$\sum_{j=1}^{n} a_{i,j}\, x_j \leq b_i,\ i = 1,2,\ldots,m$$
$$\sum_{j=1}^{n} a_{i,j}\, x_j \geq b_i,\ i = 1,2,\ldots,m$$
$$x_j \geq 0,\ j = 1,2,\ldots,n$$

Množenjem nejednakosti $\sum_{j=1}^{n} a_{i,j}\, x_j \geq b_i$ sa (-1) dovodimo problem na standardni oblik:

$$\zeta = \sum_{j=1}^{n} c_j\, x_j \to \max$$
$$\sum_{j=1}^{n} a_{i,j}\, x_j \leq b_i,\ i = 1,2,\ldots,m$$
$$\sum_{j=1}^{n} -a_{i,j}\, x_j \leq -b_i,\ i = 1,2,\ldots,m$$
$$x_j \geq 0,\ j = 1,2,\ldots,n$$

Sada kada imamo problem dat u standardnom obliku, možemo izvesti dual. Kako imamo dva skupa nejednakosti gde svaki od njih ima m nejednakosti, treba nam takođe dva skupa od po m dualnih promenljivih. Označimo dualne promenljive vezane za prvi skup od m nejednakosti sa $y_i^+, i = 1,2,\ldots,m$, i ostale dualne promenljive sa $y_i^-, i = 1,2,\ldots,m$. Sa ovim oznakama dual je dat sa:

$$\xi = \sum_{i=1}^{m} y_i^+\, b_i - \sum_{i=1}^{m} y_i^-\, b_i \to \min$$
$$\sum_{i=1}^{m} y_i^+\, a_{i,j} - \sum_{i=1}^{m} y_i^-\, a_{i,j} \geq c_j,\ j = 1,2,\ldots,n$$
$$y_i^+ \geq 0,\ y_i^- \geq 0,\ i = 1,2,\ldots,m.$$

Dobili smo dual koji možemo smenom $y_i = y_i^+ - y_i^-, i = 1,\ldots,m$, da uprostimo:

$$\xi = \sum_{i=1}^{m} y_i\, b_i \to \min$$

$$\sum_{i=1}^{m} y_i\, a_{i,j} \geq c_j,\; j = 1,2,\ldots,n$$

U problemu koji smo dobili za promenljive ne tražimo nenegativnost. Kažemo da su te promenljive **slobodne**.

Dual od poslednjeg problema je početni primar sa jednakostima.

Na osnovu toga možemo izvući pravila za postavljanje duala:

Primar	Dual
jednakosti u uslovima	slobodne promenljive
nejednakosti u uslovima	nenegativne promenljive
slobodne promenljive	jednakosti u uslovima
nenegativne promenljive	nejednakosti u uslovima

Na primer, evo jedan par primara i duala u opštem obliku:

$$(P)\quad \begin{aligned} \zeta &= -2x_1 + x_2 \to \max \\ &- x_1 + x_2 = 1 \\ &-3x_1 + x_2 \leq -1 \\ &x_1 \geq 0 \end{aligned}$$

$$(D)\quad \begin{aligned} \xi &= y_1 - y_2 \to \min \\ &-y_1 - 3y_2 \geq -2 \\ &y_1 + y_2 = 1 \\ &y_2 \geq 0 \end{aligned}$$

1.6 Matrični zapis

Do sada smo izbegavali da koristimo matrični zapis problema linearnog programiranja. U ovom odeljku ćemo sve zapisati u matričnoj formi.

Polazimo od problema datog u standardnom obliku,

$$\begin{aligned} \zeta = \sum_{j=1}^{n} c_j\, x_j &\to \max \\ \sum_{j=1}^{n} a_{i,j}\, x_j \le b_i,\ i &= 1,2,\ldots,m \\ x_j \ge 0,\ j &= 1,2,\ldots,n. \end{aligned}$$

Već smo diskutovali da možemo pomoćne promenljive w_i označiti sa x_{n+i} i to ćemo sada i uraditi:

$$w_i = x_{n+i} = b_i - \sum_{j=1}^{n} a_{i,j}\, x_j\ , i = 1,\ldots,m.$$

Problem linearnog programiranja se može zapisati pomoću matrica na sledeći način:

$$\begin{aligned} \zeta = c^T x &\to \max \\ A\, x &= b \\ x &\ge 0, \end{aligned} \tag{1.15}$$

gde je

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} & 0 & \cdots & 1 \end{bmatrix}$$

,

$$b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}, \quad c = \begin{bmatrix} c_1 \\ \vdots \\ c_n \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \\ x_{n+1} \\ \vdots \\ x_{n+m} \end{bmatrix}$$

Koristimo osobinu množenja matrica $A\, x = b$, da se proizvod ne menja ako se kolone matrice A i vrste matrice x permutuju na isti način. Koristimo i množenje matrica po blokovima.

Permutujemo matricu A tako da se u levi deo postave kolone baznih promenljivih $\mathcal{B}$, one čine podmatricu B, a u desni deo kolone nebaznih promenljivih $\mathcal{N}$, one čine podmatricu N. Tada možemo matricu A zapisati kao:

$$A = [B\ N]$$

Istim redosledom permutujemo vrste matrica $x = \begin{bmatrix} x_B \\ x_N \end{bmatrix}$ i $c = \begin{bmatrix} c_B \\ c_N \end{bmatrix}$. Onda su jednakosti iz (1.15) ekvivalentne sa

$$\zeta = c^T x = [c_B^T\ c_N^T] \begin{bmatrix} x_B \\ x_N \end{bmatrix} = c_B^T x_B + c_N^T x_N \tag{1.16}$$

$$b = Ax = [B\ N] \begin{bmatrix} x_B \\ x_N \end{bmatrix} = Bx_B + Nx_N \tag{1.17}$$

Pri tome je matrica B invertibilna, jer je sistem (1.15) rešiv po x_B.

Rečnik se dobija rešavanjem x_B i ζ preko x_N iz (1.17) i (1.16).

$$\begin{aligned} \zeta &= c_B^T(B^{-1}b - B^{-1}Nx_N) + c_N^T x_N \\ \zeta &= c_B^T B^{-1} b - \left((B^{-1}N)^T c_B - c_N \right)^T x_N \\ x_B &= B^{-1}b - B^{-1}Nx_N \end{aligned} \tag{1.18}$$

Ako uporedimo matrični zapis (1.18) sa zapisom rečnika iz (1.7), imamo:

$$(1.7) \qquad \begin{array}{rcl} \zeta &=& \bar{\zeta} + \sum_{j \in \mathcal{N}} \bar{c}_j x_j \\ \hline x_i &=& \bar{b}_i - \sum_{j \in \mathcal{N}} \bar{a}_{i,j} x_j\ ,\ i \in \mathcal{B} \end{array}$$

$$\begin{aligned} &\bar{\zeta} = c_B^T B^{-1} b \\ &[\bar{c}_j] = c_N - (B^{-1}N)^T c_B \\ &[\bar{b}_i] = B^{-1} b \\ &[\bar{a}_{i,j}] = B^{-1} N \end{aligned}$$

U zapisu desno uglaste zagrade označavaju matricu elemenata dok indeksi idu: indeks i od 1 do m, j od 1 do n. Rečničko rešenje je

$$x_N^* = 0,\ \ x_B^* = B^{-1}b,\ \ \zeta^* = c_B^T B^{-1} b. \tag{1.19}$$

Ako želimo da dobijemo rečnik duala koristeći negativno transponovanje i da zadržimo jednostavno indeksiranje za primenu komplementarnosti dualnih promenljivih, slično preimenovanju u primaru:

$$(x_1, \ldots, x_n, w_1, \ldots, w_m) \to (x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m}),$$

dodatne dualne promenljive ćemo staviti na početak, a dualne promenljive ćemo preimenovati u nastavku:

$$(z_1, \ldots, z_n, y_1, \ldots, y_m) \to (z_1, \ldots, z_n, z_{n+1}, \ldots, z_{n+m}).$$

Dualni rečnik koji negativnim transponovanjem odgovara (1.18) je:

$$\begin{aligned} -\xi &= -c_B^T B^{-1} b - \left(B^{-1}b\right)^T z_B \\ z_N &= \left(B^{-1}N\right)^T c_B - c_N + \left(B^{-1}N\right)^T z_B. \end{aligned} \tag{1.20}$$

Rečničko rešenje dualnog rečnika:

$$z_B^* = 0,\ z_N^* = \left(B^{-1}N\right)^T c_B - c_N,\ -\xi^* = -\zeta^*. \tag{1.21}$$

Koristeći oznake sa zvezdicom: (1.19) i (1.21), zapis (1.18) i (1.20) je:

$$\begin{array}{rclcrcl} & \text{Primar} & & & & \text{Dual} & \\ \zeta &=& \zeta^* - (z_N^*)^T x_N & \qquad & -\xi &=& -\zeta^* - (x_B^*)^T z_B \\ x_B &=& x_B^* - B^{-1}N x_N & & z_N &=& z_N^* + (B^{-1}N)^T z_B, \end{array} \tag{1.22}$$

gde je

$$x_B^* = B^{-1}b,\ \zeta^* = c_B^T B^{-1} b,\ z_N^* = \left(B^{-1}N\right)^T c_B - c_N. \tag{1.23}$$

U zapisu (1.22) se vidi negativno transponovanje i simetričnost primarnog i odgovarajućeg dualnog rečnika.

Takođe je očigledno da je rečnik optimalan ako i samo ako je primarno dopustiv ($x_B^* \geq 0$) i dualno dopustiv ($z_N^* \geq 0$), zato što $x \geq 0$ i $z \geq 0$.

1.7 Analiza osetljivosti

U ovom odeljku ćemo se baviti pojmom analiza osetljivosti koji daje odgovor na pitanje: kada smo našli optimalno rešenje datog problema linearnog programiranja, koliko smemo promeniti podatke tako da trenutna podela na bazične i nebazične promenljive ostane optimalna.

Često smo u situaciji da ne treba da rešimo samo jedan problem linearnog programiranja već nekoliko bliskih problema. Mnogi su razlozi zašto se to dešava. Na primer, podaci koji definišu problem mogu biti neodređeni (nepoznati u potpunosti) i onda želimo da uzmemo u obzir nekoliko mogućih scanarija. Ili podaci mogu biti poznati ali se manjaju iz dana u dan, i problem se mora iznova rešavati svakog dana. Šta god je razlog, situacija je veoma česta u praksi. Stoga se pitamo da li je moguće iskoristiti znanje o prethodno dobijenom optimalnom rešenju za dobijanje rešenja novo nastalog problema. Naravno odgovor je pozitivan i to je predmet ovog odeljka.

Posmatraćemo brojne moguće situacije, gde sve pretpostavljaju da imamo već pronađeno optimalno rešenje problema. To znači da imamo poslednji, optimalni rečnik:

$$\begin{aligned} \zeta &= \zeta^* - (z_N^*)^T x_N \\ x_B &= x_B^* - B^{-1}N x_N. \end{aligned}$$

Pretpostavimo da želimo da promenimo koeficijente u funkciji cilja c da budu na primer $\tilde{c}$. Prirodno pitanje je kako se trenutni rečnik može prilagoditi tako da ostane optimalan rečnik novog problema. Drugim rečima, mi želimo da održimo trenutnu podelu promenljivih na bazične i nebazične i da prilagodimo ζ^*, z_N^* i x_B^*. Podsetimo se da je na osnovu (1.23)

$$x_B^* = B^{-1}b, \ \zeta^* = c_B^T B^{-1}b, \ z_N^* = \left(B^{-1}N\right)^T c_B - c_N.$$

Stoga promena koeficijenata c u $\tilde{c}$ znači da moramo da ponovo izračunamo z_N^* ζ^*, dok x_B^* ostaje nepromenjeno. Stoga posle pronalaženja z_N^* ζ^*, novi rečnik će i dalje biti primarno dopustiv, i nema potrebe za primenom Faze 1 dvofaznog simpleks algoritma: možemo direktno da primenimo simpleks algoritam i ako $\tilde{c}$ nije jako različito od c možemo očekivati da novo optimalno rešenje dobijemo u relativno malom broju koraka.

Često ne želimo da rešimo modifikovan originalni problem, već postavljamo hipotetičko pitanje: Ako promenim funkciju cilja tako da povećam ili smanjim jedan od koeficijenata u funkciji cilja za malu vrednost, kolika sme da bude ta promena a da ne utiče na promenu optimalnosti moje trenutne baze? Da bismo studirali ovo pitanje, pretpostavimo da se c promeni na $c := c + t\Delta c$, gde je t realni broj i Δc je dati vektor (koji je najčešće takav da su mu svi elementi jenaki nuli a samo jedan različit od nule). Lako je primetiti da se menja z_N^* za $t\Delta z_N$

$$z_N^* := z_N^* + t\Delta z_N,$$

gde se, uobičajeno, Δc razdvaja na Δc_B i Δc_N i gde je

$$\Delta z_N = \left(B^{-1}N\right)^T \Delta c_B - \Delta c_N.$$

Interval vrednosti t za koji će rečničko rešenje (1.22) ostati optimalno dobijamo iz uslova $z_N^* \geq 0$.

Dobija se da t mora biti u intervalu:

$$\left(\min_{j\in\mathcal{N}} -\frac{\Delta z_j}{z_j^*}\right)^{-1} \leq t \leq \left(\max_{j\in\mathcal{N}} -\frac{\Delta z_j}{z_j^*}\right)^{-1}.$$

Ilustrovaćemo ovo na primeru. Posmatrajmo problem linearnog programiranja:

$$\begin{array}{ccccccc}
5x_1 & + & 4x_2 & + & 3x_3 & \to & \max \\
2x_1 & + & 3x_2 & + & x_3 & \leq & 5 \\
4x_1 & + & x_2 & + & 2x_3 & \leq & 11 \\
3x_1 & + & 4x_2 & + & 2x_3 & \leq & 8 \\
\multicolumn{7}{c}{x_1, x_2, x_3 \geq 0.}
\end{array}$$

Ovaj problem smo već rešavali i dobili smo da optimalna tabela izgleda:

	x_1	x_2	x_3	w_1	w_2	w_3	
x_1	1	2	0	2	0	-1	2
w_2	0	-5	0	-2	1	0	1
x_3	0	-1	1	-3	0	2	1
$-\zeta$	0	3	0	1	0	1	13

Optimalna baza je $\mathcal{B} = \{1,5,3\}$. Pretpostavimo da želimo da vidimo koliko smemo da promenimo koeficijenti 5 uz promenljivu x_1 u funkciji cilja a da pritom dobijeno rešenje ostane optimalno. Iz formulacije problema vidimo da je

$$c = [5 \quad 4 \quad 3 \quad 0 \quad 0 \quad 0]^T.$$

Kako nas interesuje promena prvog koeficijenta, imamo da je

$$\Delta c = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T.$$

Razdvajamo c na Δc_B i Δc_N na osnovu optimalne baze i dobijamo da je:

$$\Delta c_B = [1 \quad 0 \quad 0]^T$$

i

$$\Delta c_N = [0 \quad 0 \quad 0]^T.$$

Dalje, treba da izračunamo Δz_N koristeći jednakost:

$$\Delta z_N = \left(B^{-1}N\right)^T \Delta c_B - \Delta c_N.$$

Možemo da izračunamo $B^{-1}N$ iz početka, ali je lakše ekstrahovati datu matricu iz optimalne tabele (ona je “umešana” sa jediničnom matricom u optimalnoj tabeli):

$$B^{-1}N = \begin{bmatrix} 2 & 2 & -1 \\ -5 & -2 & 0 \\ -1 & -3 & 2 \end{bmatrix}$$

Dobijamo da je

$$\Delta z_N = [2 \quad 2 \quad -1]^T.$$

Sada uslov

$$z_N^* := z_N^* + t\Delta z_N \geq 0,$$

daje da

$$3 + 2t \geq 0, \quad 1 + 2t \geq 0, \quad 1 - t \geq 0.$$

Ove tri nejednakosti moraju sve važiti pa sumarno daju

$$-\frac{1}{2} \leq t \leq 1.$$

U terminima koeficijenta za promenljivu x_1, vidimo da se ona konačno može kretati u rasponu od 4,5 do 6.

Slično, ako je promena b oblika

$$b := b + t\Delta b,$$

da bi rešenje (1.22) ostalo optimalno, treba da bude

$$x_B^* := x_B^* + t\Delta x_B \geq 0, \text{ gde je } \Delta x_B = B^{-1}\Delta b.$$

Odatle dobijamo da t mora biti u intervalu:

$$\left(\min_{i \in \mathcal{B}} - \frac{\Delta x_i}{x_i^*} \right)^{-1} \leq t \leq \left(\max_{i \in \mathcal{B}} - \frac{\Delta x_i}{x_i^*} \right)^{-1}.$$

Glava 2

Mrežni protok

Mnogi problemi linearnog programiranja se mogu posmatrati kao problem minimizacije troškova transporta robe kroz mrežu koji bi zadovoljio potrebe za robom onih kojima treba roba koristeći robu koju na zalihama imaju drugi. Ovakvi problemi se zovu problemi mrežnog protoka i predstavljaju važnu grupu problema linearnog programiranja. Transportne, električne, komunikacione mreže samo su neki od primera primene ove oblasti. U ovom poglavlju ćemo formulisati posebnu klasu problema linearnog programiranja tzv. probleme mrežnog protoka sa minimalnom cenom transporta. Ispotaviće se da simpleks metoda primenjena za rešavanje ovih problema ima posebnu formu i specifične osobine.

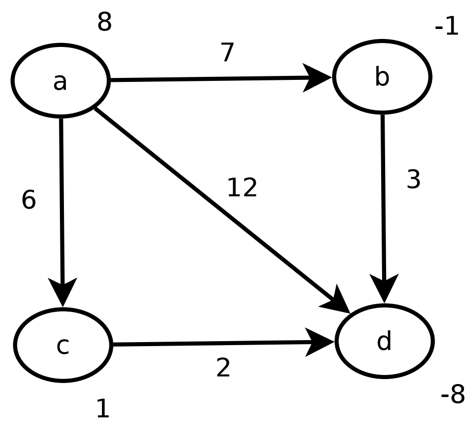
2.1 Mreže/grafovi

Mreža ili graf se sastoji od dve vrste objekata: čvorova i grana. Označićemo sa $\mathcal{N}$ skup čvorova. Neka je m ukupan broj čvorova. Čvorovi su povezani usmerenim granama. Usmerene grane znače da grana koja povezuje čvor i sa čvorom j nije isto što i grana koja povezuje čvor j sa čvorom i. Iz tog razloga ćemo granu koja povezuje čvor i sa čvorom j nazvati uređenim parom (i,j). Neka $\mathcal{A}$ označava skup svih grana u mreži. Ovaj skup je podskup skupa svih mogućih grana

$$\mathcal{A} \subset \{(i,j) : i,j \in \mathcal{N}, i \neq j\}.$$

U tipičnim mrežama skup $\mathcal{A}$ je dosta manji nego skup svih grana. Zapravo, svaki čvor je najčešće povezan samo sa susednim čvorovima. Uređeni par $(\mathcal{N},\mathcal{A})$ se naziva mreža. Nekad se takođe naziva i graf ili digraf (da bi se istaklo da su grane usmerene). Slika 2.1 pokazuje mrežu koja ima 4 čvora i 5 grana.

Da bismo definisali problem mrežnog protoka, moramo specificirati zalihe (odnosno potrebe) robe u svakom čvoru. Dakle, za svaki čvor $i \in \mathcal{N}$ označimo sa b_i zalihu robe u datom čvoru. Koristićemo konvenciju da su negativne zalihe zapravo potrebe (narudžbine). Stoga je naš problem da prevezemo zalihe koje



Slika 2.1: Mreža koja ima 4 čvora i 5 grana. Brojevi pored čvorova predstavljaju zalihe ili potrebe (negativne zalihe). Brojevi na granama predstavljaju cene prevoza jedne jedinice robe datom granom.

postoje u pojedinim čvorovima do onih čvorova kojima je potrebna roba. Prevoz se mora organizovati po usmerenim granama mreže. Kako osim zaliha i potreba koje postoje u mreži ne postoji drugi način da roba uđe ili napusti sistem, sledi da ukupne zalihe u sistemu moraju se poklapati sa ukupnim potrebama. Stoga, uvek pretpostavljamo da je:

$$\sum_{i \in \mathcal{N}} b_i = 0.$$

Takođe pretpostavljamo da trošak prevoza jedne jedinice proizvoda granom (i,j) je c_{ij}. Stoga odlučujuće promenljive predstavljaju količine robe koje se transportuju svakom od grana. Za svako $(i,j) \in \mathcal{A}$, x_{ij} će označavati količinu robe prevezenu od čvora i do čvora j granom (i,j). Cilj je da se minimizira ukupan trošak prevoza zaliha kako bi se zadovoljile potrebe:

$$\min \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij}$$

Kao što smo već napomenuli, ograničenja su da ukupan protok kroz svaki čvor mora biti jednak njegovim potrebama. Posmatrajmo fiksiran čvor, $k \in \mathcal{N}$. Ukupan protok robe koja ulazi u čvor k je dat:

$$\sum_{i:(i,k) \in \mathcal{A}} x_{ik}.$$

Slično ukupan protok robe koja izlazi iz čvora k je dat sa:

$$\sum_{j:(k,j) \in \mathcal{A}} x_{kj}.$$

Razlika između ove dve veličine mora biti jednaka potrebama posmatranog čvora:

$$\sum_{i:(i,k)\in\mathcal{A}} x_{ik} - \sum_{j:(k,j)\in\mathcal{A}} x_{kj} = -b_k, \quad k \in \mathcal{N}.$$

Drugim rečima, za svaki čvor mora da važi da je "ulaz − izlaz = potreba (narudžbina) = −zaliha".

Najzad, količina koja se prevozi mora biti nenegativna celobrojna vrednost

$$x_{ij} \geq 0, \quad x_{ij} \in \mathbb{Z}, \quad (i,j) \in \mathcal{A}.$$

Slika 2.1 pokazuje na čvorovima i granama redom potrebe/zalihe pojedinih čvorova i cene prevoza granama. U posmatranom primeru veličine koje smo uveli su:

- skup čvorova $\mathcal{N} = \{a,b,c,d\}$, $\#\mathcal{N} = m = 4$,
- skup grana $\mathcal{A} = \{(a,b),(a,c),(a,d),(b,d),(c,d)\}$, $\#\mathcal{A} = n = 5$
- zalihe
$$\begin{array}{cccccccc} & & & a & b & c & d & \\ b & = & [& 8 & -1 & 1 & -8 &]^T \end{array}$$
- cene prevoza za pojedine grane
$$\begin{array}{cccccccccc} & & & (a,b) & (a,c) & (a,d) & (b,d) & (c,d) & \\ c & = & [& 7 & 6 & 12 & 3 & 2 &]^T \end{array}$$
- količine robe za pojedine grane
$$\begin{array}{cccccccccc} & & & (a,b) & (a,c) & (a,d) & (b,d) & (c,d) & \\ c & = & [& x_{ab} & x_{ac} & x_{ad} & x_{bd} & x_{cd} &]^T \end{array}$$

Problem minimizacije cene transporta u našem primeru je:

$$\begin{array}{rcrcrcrcrcrcr} \zeta & = & 7\,x_{ab} & + & 6\,x_{ac} & + & 12\,x_{ad} & + & 3\,x_{bd} & + & 2\,x_{cd} & \to & \min \\ a: & & -\,x_{ab} & - & x_{ac} & - & x_{ad} & & & & & = & -8 \\ b: & & x_{ab} & & & & & - & x_{bd} & & & = & 1 \\ c: & & & & x_{ac} & & & & & - & x_{cd} & = & -1 \\ d: & & & & & & x_{ad} & + & x_{bd} & + & x_{cd} & = & 8 \end{array}$$

$$x_{ab} \geq 0, \quad x_{ac} \geq 0, \quad x_{ad} \geq 0, \quad x_{bd} \geq 0, \quad x_{cd} \geq 0$$

$$x_{ab} \in \mathbb{Z}, \quad x_{ac} \in \mathbb{Z}, \quad x_{ad} \in \mathbb{Z}, \quad x_{bd} \in \mathbb{Z}, \quad x_{cd} \in \mathbb{Z}$$

Problem mrežnog protoka je problem celobrojnog linearnog programiranja. U matirčnoj formi ovaj problem je dat sa:

$$\begin{aligned} \zeta = c^T x &\to \min \\ Ax &= -b \\ x &\geq 0 \end{aligned}$$

gde matrica sistema A se zove matrica incidencija grafa i data je sa:

$$A = \begin{bmatrix} -1 & -1 & -1 & & & \\ 1 & & & -1 & & \\ & 1 & & & -1 & \\ & & 1 & 1 & 1 & \end{bmatrix}$$

U opštem slučaju problem pronalaženja najjeftinijeg transporta izgleda:

$$\zeta = \sum_{(i,j)\in\mathcal{A}} c_{ij}\, x_{ij} \to \min$$

$$\sum_{\substack{i \\ (i,k)\in\mathcal{A}}} x_{ik} - \sum_{\substack{j \\ (k,j)\in\mathcal{A}}} x_{kj} = -b_k\,, \ \text{za} \ k \in \mathcal{N}$$

$$x_{ij} \geq 0\,,\, x_{ij} \in\,, \ \text{za} \ (i,j) \in \mathcal{A}$$

ili u matričnoj formi zajedno sa odgovarajućim dualom:

Primar	Dual
$\zeta = c^T x \to \min$	$-\xi = -b^T y \to \max$
$Ax = -b$	$A^T y + z = c$
$x \geq 0$	$z \geq 0$

Jednačine duala glase: $\forall (i,j) \in \mathcal{A}$, $-y_i + y_j + z_{ij} = c_{ij}$, $z_{ij} \geq 0$. Formati matrica koje figurišu u oba problema su redom $A \to m \times n$; $c, x, z \to n \times 1$; $b, y \to m \times 1$. Najzad, iz komplementarnosti dualnih promenljivih sledi da je

$$x_{ij} z_{ij} = 0, \quad (i,j) \in \mathcal{A}.$$

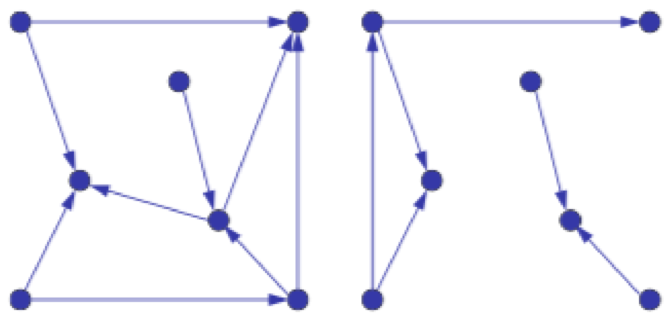
2.2 Pokrivajuća stabla

Uvešćemo niz definicija koje su nam potrebne za dalji rad. Uređena lista čvorova $(n_1, n_2, \ldots, n_k)$ se naziva *putanja* u mreži ako je svaki od susednih čvorova povezan granom u mreži (smer grane nije važan). Mreža se naziva povezana ako postoji putanja koja povezuje svaki par čvorova (Slika 2.2). U nastavku posmatramo samo povezane mreže (grafove).

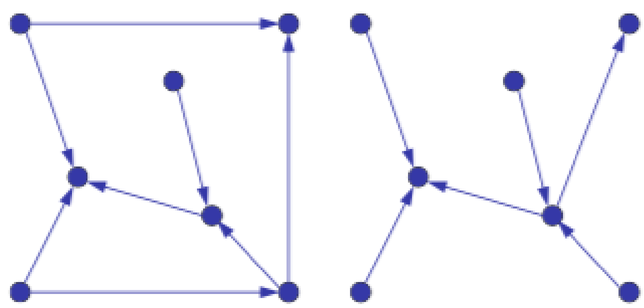
Kontura je putanja u kojoj je prvi čvor ujedno i poslednji. Mreža se zove aciklična ako ne sadrži ni jednu konturu (Slika 2.3).

Mreža se zove drvo ako je povezana i aciklična (Slika 2.4). Mreža $(\tilde{\mathcal{N}}, \tilde{\mathcal{A}})$ se zove podmreža od $(\mathcal{N}, \mathcal{A})$ ako je $\tilde{\mathcal{N}} \subset \mathcal{N}$ i $\tilde{\mathcal{A}} \subset \mathcal{A}$. Podmreža $(\tilde{\mathcal{N}}, \tilde{\mathcal{A}})$ se zove pokrivajuće drvo ako je drvo i ako je $\tilde{\mathcal{N}} = \mathcal{N}$. Kako se skup čvorova pokrivajućeg drveta poklapa sa skupom čvorova polazne mreže, pokrivajuće drvo je određeno skupom njegovih grana.

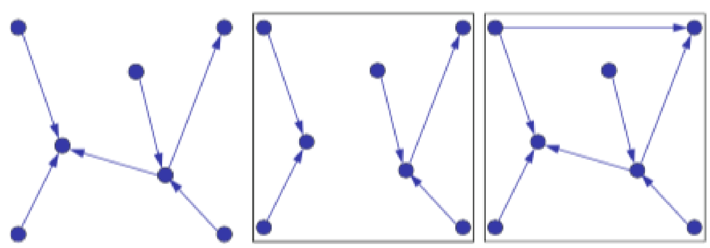
Za dati problem mrežnog protoka, bilo koji izbor vrednosti promenljivih koji zadovoljava sistem jednačina $Ax = -b$ se naziva balansirani protok. Važno je primetiti da mi ne zahtevamo da protok bude nenegativan da bi bio balansiran (ne tražimo da važi $x \geq 0$, već samo $Ax = -b$). Drugim rečima, dozvoljavamo da



Slika 2.2: Graf levo je povezan dok graf desno nije.



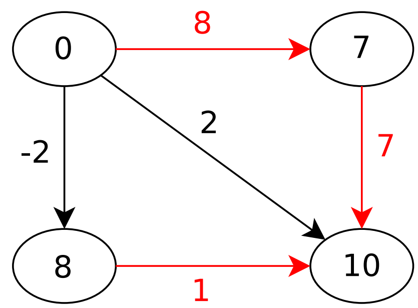
Slika 2.3: Graf levo sadrži konturu dok graf desno je acikličan.



Slika 2.4: Graf levo je drvo dok druga dva grafa (srednji i desno) nisu - prvi nije jer nije povezan a drugi nije jer sadrži konturu.

roba ide u suprotnom smeru od usmerenja grane. Ukoliko je balansiran protok nenegativan, onda je i dopustiv. Za dato pokrivajuće drvo, balansiran protok koji dodeljuje nulu svakoj grani koja nije deo pokrivajućeg drveta se zove rešenje za dato drvo. Posmatrajmo drvo prikazano na Slici 2.5.

Brojevi prikazani na granama pokrivajućeg drveta (crvene grane) daju rešenje koje odgovara potrebama/zalihama prikazanim na Slici 2.1. Oni su dobijeni



Slika 2.5: Crvene grane predstavljaju pokrivajuće drvo u grafu prikazanom na Slici 2.1. Brojevi prikazani na crvenim granama su primarni protoci (x), brojevi u kružićima su dualne promenljive (y), a brojevi pored crnih grana koje ne pripadaju pokrivajućem drvetu su dualne pomoćne promenljive (z).

tako što se krenulo od “listova” drveta i nastavilo se ka unutrašnjosti. Na primer, primarni protoci mogu se dobiti na sledeći način tako što protoke za grane koje nisu na pokrivajućem drvetu proglasimo da su jednaki nuli $x_{ac} = 0$ i $x_{ad} = 0$ pa koristeći sistem jednačina $Ax = -b$ dobijamo:

$$\begin{array}{lccccccccc} a: & & & x_{ab} & = & 8 & & & & \\ b: & x_{ab} & - & x_{bd} & = & 1 & \Longrightarrow & x_{bd} & = & 7 \\ c: & x_{ac} & - & x_{cd} & = & -1 & \Longrightarrow & x_{cd} & = & 1 \end{array}$$

Primetimo da jednačina protoka za čvor d je višak i zadovoljena je za dobijeni protok.

Ne samo da možemo naći primarne protoke, već istovremeno možemo naći i dualne promenljive. Rešenje duala se sastoji u pronalaženju vrednosti dualnih originalnih i pomoćnih promenljivih tj. vrednosti y_i i z_{ij}. Ove promenljive moraju da zadovoljavaju jednačine:

$$\forall (i,j) \in \mathcal{A}, \quad -y_i + y_j + z_{ij} = c_{ij}.$$

Zbog komplementarnosti dualnih promenljivih, na crvenim granama tj. granama pokrivajućeg drveta $\mathcal{T}$ vrednosti pomoćnih promenljivih duala $z_{ij} = 0$. Stoga,

$$-y_i + y_j = c_{ij}, \quad (i,j) \in \mathcal{T}.$$

Kako pokrivajuće drvo ima m čvorova i $m-1$ grana, ove jednačine daju sistem od $m-1$ jednačine sa m nepoznatih. Stoga možemo proizvoljno izabrati jedno y_i da bude jednako nuli (to zovemo korenom). Vrednosti y pišemo u kružićima na grafu.

Neka je čvor a koren tj. neka je $y_a = 0$ u našem primeru (Slika 2.5). Počevši od korena, i koristeći jednačine za promenljive y dobijamo:

$$\begin{array}{rrcrcrcrcr} & & & y_a & = & 0 & & & & \\ (a,b): & -y_a & + & y_b & = & 7 & \Longrightarrow & y_b & = & 7 \\ (b,d): & -y_b & + & y_d & = & 3 & \Longrightarrow & y_d & = & 10 \\ (c,d): & -y_c & + & y_d & = & 2 & \Longrightarrow & y_c & = & 8 \end{array}$$

Sada kada znamo vrednosti dualnih promenljivih y, koristeći grane koje nisu u pokrivajućem drvetu (crne grane) i cene protoka na njima dobijamo vrednosti promenljivih z_{ij}:

$$-y_i + y_j + z_{ij} = c_{ij} \quad \forall (i,j) \notin \mathcal{T}.$$

Vrednosti tih promenljivih su prikazane na slici 2.5 na crnim granama:

$$\begin{array}{rrcrcrcrcrcr} (a,c): & -y_a & + & y_c & + & z_{ac} & = & 6 & \Longrightarrow & z_{ac} & = & -2 \\ (a,d): & -y_a & + & y_d & + & z_{ad} & = & 12 & \Longrightarrow & z_{ad} & = & 2 \end{array}$$

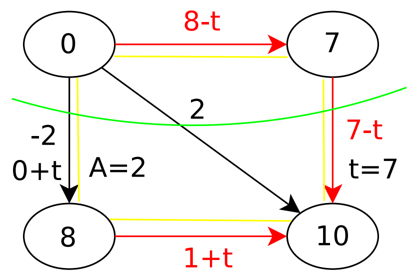
Na osnovu teorije o dualnosti, znamo da je trenutno rešenje optimalno ako su svi protoci (primarne promenljive, tj. x) nenegativni i ako su sve dualne pomoćne promenljive (z) takođe nenegativne. Rešenje koje smo dobili u našem primeru (Slika 2.5) zadovoljava prvi uslov ali ne zadovoljava drugi uslov, tj. rešenje koje smo dobili je primarno ali ne i dualno dopustivo. Stoga možemo primeniti primarni simpleks kako bismo našli optimalno rešenje.

2.3 Primarna simpleks metoda za mrežni protok

Objasnićemo primarnu simpleks metodu na našem primeru. Kao što je pomenuto ranije, rešenje prikazano na Slici 2.5 je primarno dopustivo ali nije dualno dopustivo. Osnovna ideja primarne simpleks metode je da izabere granu koja ne pripada pokrivajućem drvetu (crna grana) i koja ujedno nije dualno dopustiva i da joj dozvoli da postane grana pokrivajućeg drveta (tj. da postane crvena grana) i onda da sve prilagodi tako da i dalje imamo očuvanu primarnu dopustivost.

Prva iteracija - Za naš prvi pivot uzimamo granu koja nije bila deo pokrivajućeg drveta i koja je negativna. Neka grana (a,c) uđe u bazu (postane deo pokrivajućeg drveta). Toj grani dodajemo vrednost t i preračunaćemo sve ostale protoke (x promenljive) tako da oni koji nisu u drvetu ostaju jednaki nuli a oni koji su u drvetu se moraju ažurirati tako da imamo i dalje balansiran protok. Kada postojećem drvetu dodamo granu (a,c) ona će sa ostalim granama drveta formirati konturu (žuto na Slici 2.6, konturu čine čvorovi a, b, c i d). Protok na datoj konturi se mora promeniti tako da kompenzuje povećanje protoka na ulaznoj grani. Protoci na ostalim granama drveta koji nisu delovi konture ostaju nepromenjeni (u našem primeru takvih grana nema ali da postoje treba na njima ostaviti istu vrednost protoka tj. primarne promenljive x). U našem primeru protoci na konturi se povećavaju za vrednost t ako su usmereni kao i ulazna grana a smanjuju se za vrednost t ako su suprotnog smera od smera ulazne grane.

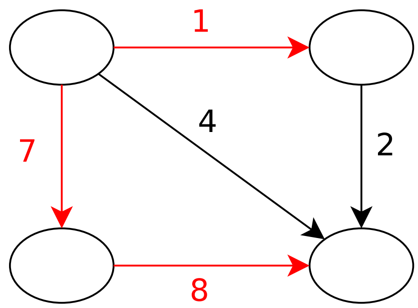
Kako t raste to će u jednom momentu grana (b,d) postati nula i ona je izlazna grana i uzmimamo da je $t = 7$.



Slika 2.6: Pivotizacija na grafu prikazanom na slici 2.5 pomoću pravila kontura i mostova.

Možemo zaključiti da je pravilo za odabir izlazne grane sledeće: "Izlazna grana mora biti suprotno orijentisana od ulazne grane u konturi, i od svih takvih grana ona mora imati minimalan protok". Pravilo za ažuriranje primarnih promenljivih u konturi glasi: "Protoci koji odgovaraju granama koje su isto usmerene kao i izlazna grana se smanjuju za vrednost izlazne grane a oni koji su suprotno usmereni se uvećavaju za datu vrednost." To pravilo se još zove i **Pravilo kontura** i ono glasi: "Uslovi zaliha ostaju zadovoljeni ako se idući po konturi u zavisnosti od smera dodaje ili oduzima ista vrednost t". Biramo vrednost $t > 0$ tako da ni jedna od bazičnih promenljivih ne postane negativna, a bar jedna postane 0. Jedna od promenljivih koja postane 0 se bira da izađe iz baze. Izbor grane sa konture za ulazak u bazu očuvava pokrivajuće drvo. Novi rečnik će biti primarno dopustiv.

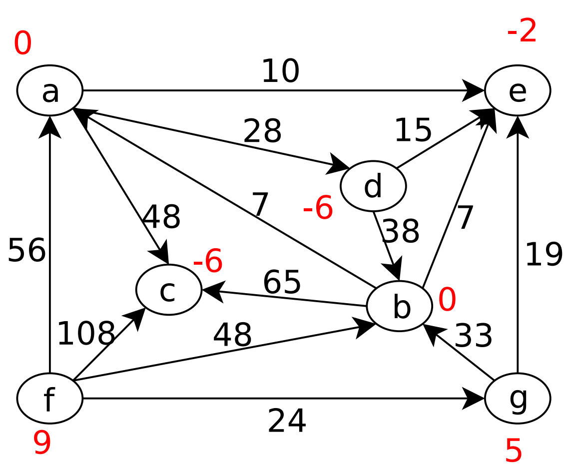
Sledeći zadatak je ažurirati dualne promenljive. Primetimo da ako izbrišemo izlaznu granu sa pokrivajućeg drveta pre nego što ulazna grana postane deo pokrivajućeg drveta, pokrivajuće drvo će se razdvojiti na dva drveta. U našem primeru to bi bila drva sa čvorovima a i b i sa čvorovima c i d. Između te dve grupe čvorova povlačimo granicu (zelena granica na Slici 2.6). Dualne promenljive ažuriramo koristeći pravilo mostova. **Pravilo mostova**: "Pri odabranoj pivotizaciji, menjaju se z promenljive dualnog rečničkog rešenja samo za grane koje idu preko granice dodavanjem ili oduzimanjem iste vrednosti A u zavisnosti od smera prelaska preko granice." Vrednost $A > 0$ u pravilu mostova uzimamo kao suprotnu od z vrednosti grane koja ulazi u bazu, jer ona postaje 0. U našem primeru z vrednost grane koja ulazi u bazu je -2, te je stoga $A = 2$. Grane koje prelaze preko granice a koja su nebažicne su grane (a,d) i (b,d) i njihove vrednosti se povećavaju za $A = 2$ jer su isto usmerene kao i ulazna grana. Protok koji dobijamo je optimalan (Slika 2.7) jer je i primarno i dualno dopustiv.



Slika 2.7: Optimalan protok - minimalna ostvarena cena transporta je $\zeta^* = 1 \cdot 7 + 7 \cdot 6 + 8 \cdot 2 = 65$.

2.3.1 Dodatni primer - primarni simpleks

Ponovićemo sve ovo na još jednom primeru. Dat je graf prikazan na Slici 2.8.



Slika 2.8: Inicijalni graf. Brojevi (crveni) pored čvorova predstavljaju zalihe/potrebe a brojevi na granama predstavljaju cene transporta jedne jedinice proizvoda.

Cene transporta jedne jedinice robe su date na granama dok su zalihe/potrebe date pored čvorova. Treba odrediti koliko jedinica proizvoda treba pojedinim granama transportovati tako da ukupna cena transporta bude minimalna. Problem koji rešavamo je:

$$
\begin{gathered}
\text{Primar}\\
\zeta = c^T x \to \min\\
Ax = -b\\
x \geq 0
\end{gathered}
$$

gde je

$$
\begin{aligned}
x^T &= \begin{bmatrix} x_{ac} & x_{ad} & x_{ae} & x_{ba} & x_{bc} & x_{be} & x_{db} & x_{de} & x_{fa} & x_{fb} & x_{fc} & x_{fg} & x_{gb} & x_{ge} \end{bmatrix}\\
A &= \begin{bmatrix}
-1 & -1 & -1 & 1 & & & & & 1 & & & & & \\
 & & & -1 & -1 & -1 & 1 & & & 1 & & & 1 & \\
1 & & & & 1 & & & & & & 1 & & & \\
 & 1 & & & & & -1 & -1 & & & & & & \\
 & & 1 & & & 1 & & 1 & & & & & & 1\\
 & & & & & & & & -1 & -1 & -1 & -1 & & \\
 & & & & & & & & & & & 1 & -1 & -1
\end{bmatrix}\\
c^T &= \begin{bmatrix} 48 & 28 & 10 & 7 & 65 & 7 & 38 & 15 & 56 & 48 & 108 & 24 & 33 & 19 \end{bmatrix}\\
b^T &= \begin{bmatrix} 0 & 0 & -6 & -6 & -2 & 9 & 5 \end{bmatrix}
\end{aligned}
$$

Za pokrivajuće drvo sa granama (a,d), (b,c), (f,a), (f,b), (g,b) i (g,e) (crvene grane na Slici 2.9) treba odrediti prvo vrednosti protoka (promenljive x), zatim u čvorovima odrediti vrednosti dualnih promenljivih y i na kraju za crne grane (grane koje ne pripadaju pokrivajućem drvetu) vrednosti dualnih pomoćnih promenljivih z.

Vrednosti protoka (vrednosti promenljivih x) određujemo tako što na crnim granama su vrednosti promenljivih x jednake nuli, tj. $x_{ac} = 0$, $x_{ae} = 0$, $x_{ba} = 0$, $x_{be} = 0$, $x_{db} = 0$, $x_{de} = 0$, $x_{fc} = 0$, $x_{fg} = 0$, a ostale vrednosti nalazimo rešavajući sistem linearnih jednačina $Ax = -b$:

$$
\begin{array}{lccccccccc}
d: & & & & & x_{ad} & = & 6 & & \\
a: & & & x_{fa} & - & x_{ad} & = & 0 & \Longrightarrow & x_{fa} = 6\\
f: & & & -x_{fa} & - & x_{fb} & = & -9 & \Longrightarrow & x_{fb} = 3\\
c: & & & & & x_{bc} & = & 6 & & \\
b: & x_{gb} & + & x_{gb} & - & x_{bc} & = & 0 & \Longrightarrow & x_{gb} = 3\\
e: & & & & & x_{ge} & = & 2 & &
\end{array}
$$

Vrednosti tih promenljivih su prikazane na slici 2.9 na crvenim granama.

Vrednosti dualnih promenljivih y dobijamo uz pomoć jednačina

$$
-y_i + y_j = c_{ij}, \quad (i,j) \in \mathcal{T}
$$

koje važe za crvene grane (pokrivajuće drvo). Kako je jedna jednačina višak, uzimamo npr. da je koren čvor g i proglašavamo da je $y_g = 0$. Ostale vrednosti dobijamo pomoću jednačina:

$$\begin{array}{rrcrcrcrcr}
 & & & y_g & = & 0 & & & & \\
(g,e): & -y_g & + & y_e & = & 19 & \Longrightarrow & y_e & = & 19 \\
(g,b): & -y_g & + & y_b & = & 33 & \Longrightarrow & y_b & = & 33 \\
(b,c): & -y_b & + & y_c & = & 65 & \Longrightarrow & y_c & = & 98 \\
(f,b): & -y_f & + & y_b & = & 48 & \Longrightarrow & y_f & = & -15 \\
(f,a): & -y_f & + & y_a & = & 56 & \Longrightarrow & y_a & = & 41 \\
(a,d): & -y_a & + & y_d & = & 28 & \Longrightarrow & y_d & = & 69
\end{array}$$

Vrednosti tih promenljivih su prikazane na slici 2.9 u kružićima.

Na kraju, pomoću jednačina

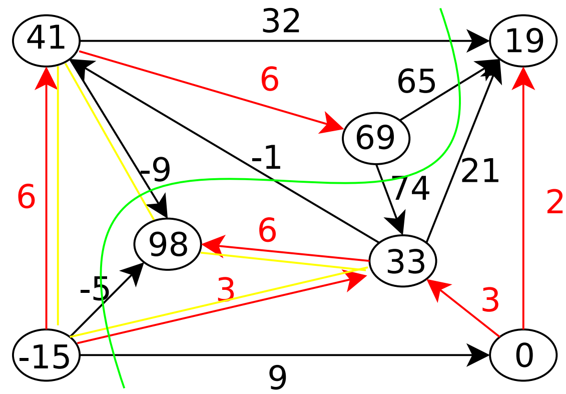
$$-y_i + y_j + z_{ij} = c_{ij} \quad \forall (i,j) \notin \mathcal{T}$$

koje važe na crnim granama, dobijamo vrednosti pomoćnih dualnih promenljivih z. Vrednosti tih promenljivih su prikazane na slici 2.9 na crnim granama:

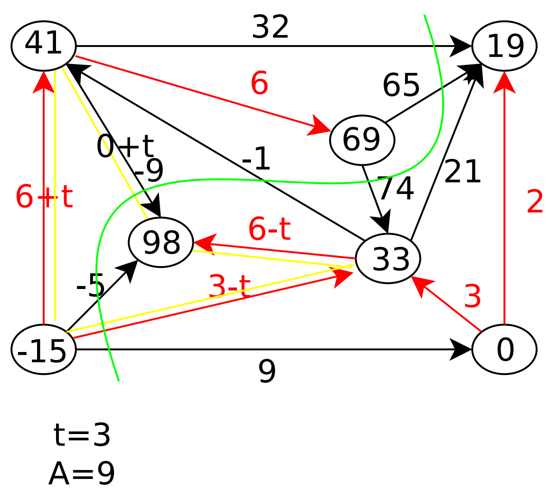
$$\begin{array}{rrcrcrcrcrcr}
(a,c): & -y_a & + & y_c & + & z_{ac} & = & 48 & \Longrightarrow & z_{ac} & = & -9 \\
(a,e): & -y_a & + & y_e & + & z_{ae} & = & 10 & \Longrightarrow & z_{ae} & = & 32 \\
(b,a): & -y_b & + & y_a & + & z_{ba} & = & 7 & \Longrightarrow & z_{ba} & = & -1 \\
(b,e): & -y_b & + & y_e & + & z_{be} & = & 7 & \Longrightarrow & z_{be} & = & 21 \\
(d,b): & -y_d & + & y_b & + & z_{db} & = & 38 & \Longrightarrow & z_{db} & = & 74 \\
(d,e): & -y_d & + & y_e & + & z_{de} & = & 15 & \Longrightarrow & z_{de} & = & 65 \\
(f,c): & -y_f & + & y_c & + & z_{fc} & = & 108 & \Longrightarrow & z_{fc} & = & -5 \\
(f,g): & -y_f & + & y_g & + & z_{fg} & = & 24 & \Longrightarrow & z_{fg} & = & 9
\end{array}$$

Sada kad smo odredili vrednosti svih promenljivih, vidimo da problem jeste primarno dopustiv (crvene grane su nenegativne) ali nije dualno dopustiv (ima crnih grana koje imaju negativnu vrednost) te vršimo pivotizacije. Najnegativnija crna grana (a,c) ulazi u bazu (postaje deo pokrivajućeg drveta). Ona sa ostalim granama pokrivajućeg drveta pravi konturu (žuto na Slici 2.9). Pomoću dobijene konture ćemo ažurirati vrednosti crvenih grana koje su deo konture (ostale crvene grane ostaju nepromenjene). Grana koja ulazi u bazu dobija vrednost t, dok ostale grane na konturi koje su isto kao i ona usmerene se povećavaju za istu vrednost t a one koje su suprotno usmerene se smanjuju za t. Najveće t koje možemo da izaberemo a da protok i dalje ostane primarno dopustiv je $t = 3$. To znači da grana (f,b) izlazi iz baze tj. ona više neće biti deo pokrivajućeg drveta.

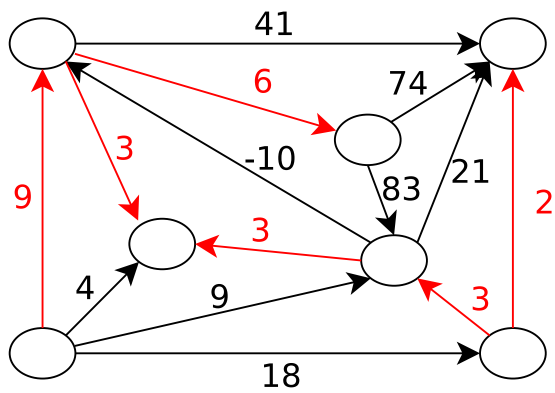
Pre nego što grana (a,c) uđe u bazu a nakon što grana (f,b) izađe iz baze, graf će se raspasti na dve povezane komponente, na grupu čvorova a, d i f i ostatak. Na Slici 2.10 granica između ove dve grupe čvorova je predstavljena zelenom krivom. Sve crne grane koje tu granicu prelaze u istom smeru kao i grana koja ulazi u bazu se uvećavaju za vrednost A, a one grane koje prelaze tu granicu u suprotnom smeru se smanjuju za A. Vrednost parametera A dobijamo kao suprotnu vrednost od z vrednosti grane koja ulazi u bazu, a to je u ovom



Slika 2.9: Graf sa vrednostima x promenljivih na crvenim granama, u kružićima su vrednosti dualnih promenljivih y a na crnim granama vrednosti pomoćnih dualnih promenljivih z.



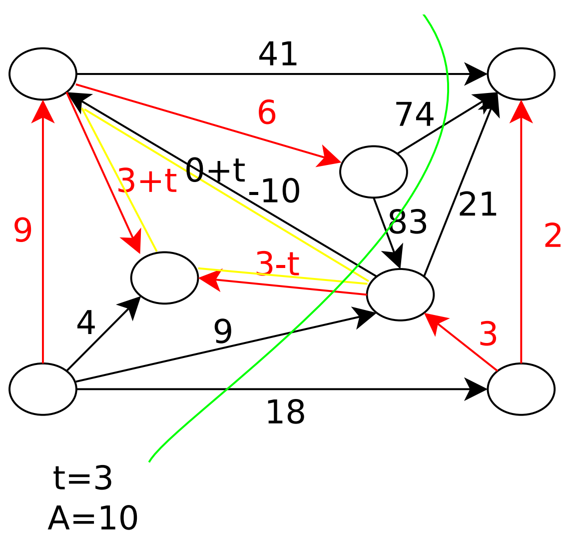
Slika 2.10: U prvoj iteraciji najnegativnija crna grana ((a,c), $z_{ac} = -9$) ulazi u bazu i obrazuje sa postojećim crvenim granama konturu (žuto). Iz baze izlazi crvena grana (f,b) (na kojoj piše $3-t$), njenim izlaskom se drvo raspada na dve povezane komponente koje razdvaja zelena granica. Pravilo kontura i mostova nam redom određuje nove vrednosti promenljivih x (crvene grane) i promenljivih z (crne grane). Kružiće (vrednosti dualnih promenljivih y) više ne moramo u narednim koracima da ažuriramo.



Slika 2.11: Novonastali graf nakon pivotizacije. On i dalje jeste primarno ali nije dualno dopustiv.

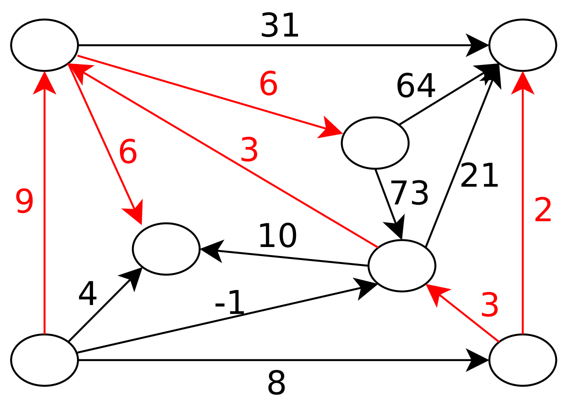
primeru 9. Nakon ažuriranja crnih grana dobijamo novi graf prikazan na Slici 2.11.

Ovaj protok nije dualno dopustiv te nastavljamo sa pivotizacijama dok ne dobijemo optimalno rešenje. U bazu ulazi najnegativnija crna grana (b, a). Njena x vrednost se sa nule povećava na t. Vrednost parametera t određujemo pomoću pravila konture (Slika 2.12) i dobijamo da je $t = 3$.



Slika 2.12: Druga iteracija.

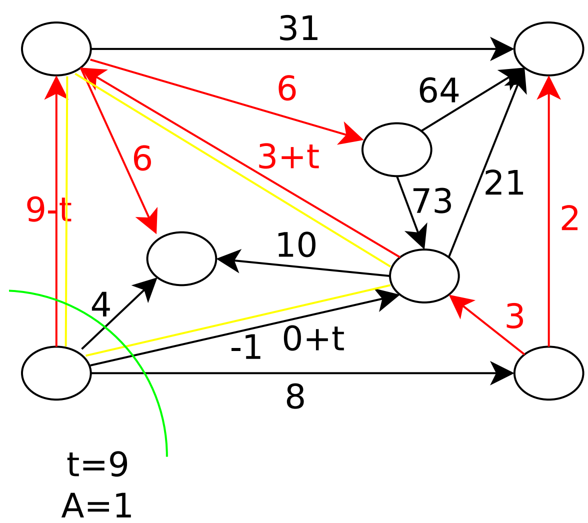
Zelena kriva predstavlja granicu između dve grupe povezanih čvorova koje nastanu od posmatranog grafa kada izađe grana (b, c) iz drveta a pre nego što uđe grana (b,a). Pomoću pravila mostova ažuriramo crne grane (na kojima pišu vrednosti pomoćnih dualnih promenljivih z). Vrednost parametera $A = 10$. Nakon ažuriranja dobijamo graf prikazan na Slici 2.13.



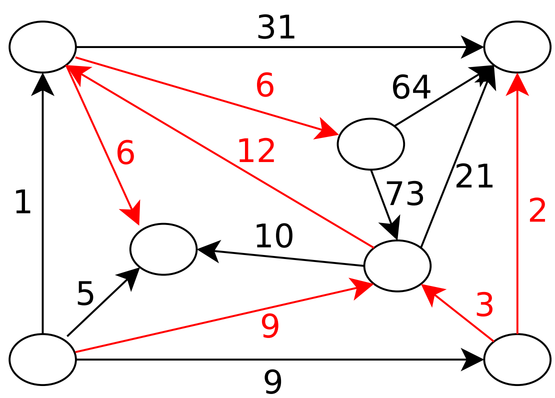
Slika 2.13: Graf nakon druge pivotizacije.

On i dalje nije optimalan te vršimo novu pivotizaciju (Slika 2.14). U bazu ulazi grana (f,b), obrazuje konturu označenu žutom bojom na slici iz koje dobijamo da je vrednost $t = 9$. Iz baze izlazi grana (f,a) čijim izlaskom (a pre nego što uđe u bazu grana (f,b)) se drvo razdvaja na dve povezane komponente - jednu čini čvor f a drugu ostali čvorovi. Zelena kriva na slici predstavlja granicu između dve grupe čvorova i crne grane koje prelaze tu granicu se ažuriraju uvećavajući/smanjujući se za vrednost $A = 1$ u zavisnosti od smera.

Nakon ove pivotizacije dobijamo optimalni protok prikazan na Slici 2.15.



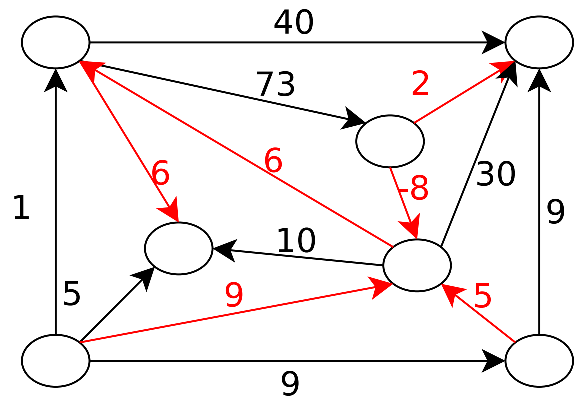
Slika 2.14: Treća iteracija.



Slika 2.15: Optimalan mrežni protok (i primarno i dualno dopustiv).

2.4 Dualna simpleks metoda za mrežni protok

U prethodnom odeljku smo naveli pravila za primarni simpleks koji koristimo u situaciji kada je mrežni protok primarno dopustiv a nije dualno dopustiv. Kada je pak mrežni protok dualno dopustiv a nije primarno dopustiv onda se dualni simpleks koristi kako bi se dobilo optimalno rešenje. Pretpostavimo da imamo mrežni protok dat na Slici 2.16. On je dualno dopustiv ali nije primarno dopustiv jer je $x_{db} < 0$. Osnovna ideja dualnog simpleksa je da se izabere crvena grana koja je primarno nedopustiva i da ona napusti pokrivajuće drvo (tj. da postane nebazična) i onda da se sve ažurira tako da se očuva dualna dopustivost.

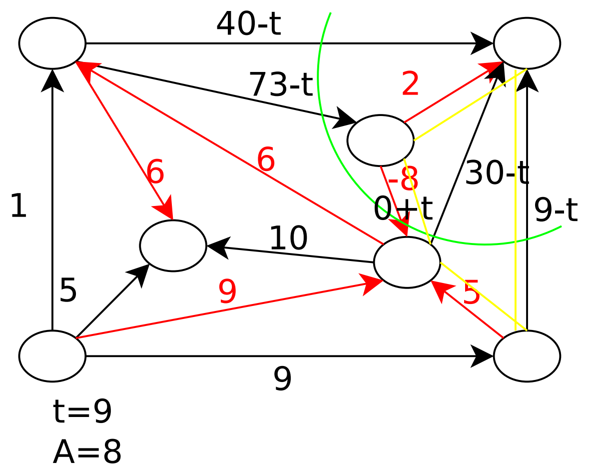


Slika 2.16: Mrežni protok koji nije primarno ali jeste dualno dopustiv.

U prvoj iteraciji grana (d,b) kao najnegativnija izlazi iz pokrivajućeg drveta i vršimo dualnu pivotizaciju na sledeći način. Uklanjanjem grane (d,b) pokrivajuće drvo će se raspasti na dve povezane komponente (na Slici 2.17 označeno zelenom krivom). Ulazna grana mora biti neka od grana koja spaja ove dve povezne komponente kako bi ih opet povezala u pokrivajuće drvo (grana koja prelazi zelenu granicu). Drugim rečima, mora biti jedna od grana (a,d), (a,e), (b,e), ili (g,e). Da bismo odlučili koja od datih grana će ući u bazu, moramo pažljivo razmotriti uticaj svake od njih.

Kao što smo već napomenuli pokrivajuće drvo se bez izlazne grane sastoji od dve povezane komponente. Ulazna grana ih ponovo mora povezati. Prvo, razmotrimo situaciju kada ulazna grana ima isti smer kao i izlazna grana. Kada toj ulaznoj grani dodamo protok, moramo na izlaznoj grani umanjiti dati protok kako bismo očuvali uslov da ukupan protok bude isti. Stoga protok na izlaznoj grani koji je bio negativan nikako ne možemo dovesti da bude jednak nuli. Drugim rečima izlazna grana na taj način ne može da napusti pokrivajuće drvo, a to nam ne odgovara. Razmotrimo sada situaciju kada ulazna grana ima suprotan smer od smera izlazne grana. Kako je to ulazna grana njena z vrednost će ulaskom u pokrivajuće drvo postati nula (smanjiće se). Sve ostale grane koje imaju isti smer i koje povezuju dve komponente pokrivajućeg drveta će se smanjiti za istu veličinu. Kako bismo očuvali nenegativnost, moramo uzeti onu čije smanjenje je najmanje. Možemo ovo pravilo sumirati: "Ulazna grana je ona koja povezuje dve povezane komponente i suprotnog je smera od smera izlazne grane. Među svim takvim granama, ona mora imati najmanju vrednost dualne pomoćne promenljive z".

U našem primeru sve grane koje povezuju dve povezane komponente imaju smer suportan smeru izlazne grane. Ona grana koja ima najmanju z vrednost je

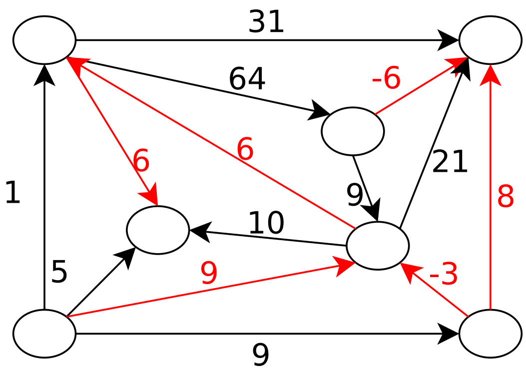


Slika 2.17: Prva iteracija dualnog simpleksa.

grana (g,e) $(z_{ge} = 9)$ (slika 2.17). Ova grana je ulazna grana. Drugim rečima, mi vrednost izlazne grane $z_{db} = 0$ uvećavamo za t. Po pravilu mostova ažuriranje z vrednosti se vrši tako što ćemo vrednosti $z_{ad} = 40$, $z_{ae} = 73$, $z_{be} = 30$ i $z_{ge} = 9$, umanjiti za t. Biramo $t = 9$, grana (g,e) ulazi u bazu.

Sada imamo poznatu i izlaznu i ulaznu granu, pa samim tim i novo pokrivajuće drvo. Sve promenljive koje su vezane za novo pokrivajuće drvo moraju biti ažurirane. Dalje, pravila za ažuriranje su sledeća. Ako bi grana (g,e) ušla u bazu pre nego što (d,b) izađe iz baze, nastala bi kontura (označena žuto na Slici 2.17). Po pravilu kontura vrednosti transporta x po konturi se u jednom smeru povećavaju, u suprotnom smanjuju za vrednost A. Na Slici 2.18 je prikazano novo rešenje.

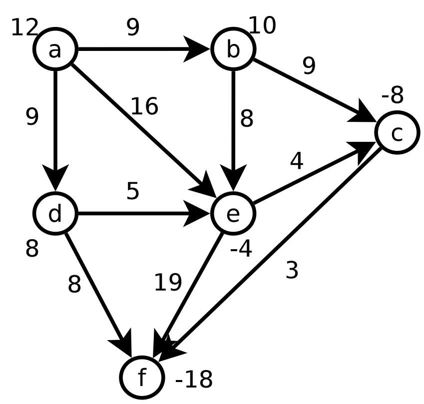
Druga iteracija - za drugog pivota postoje dva izbora izlazne grane (g,b) i (d,e). Uzimamo najnedopustiviju granu a to je (d,e). Uklanjamo je iz pokrivajućeg drveta i dobijamo dve povezane komponente. Jedna povezana komponenta sadrži samo čvor d a druga ostale čvorove. Ulazna grana mora povezati ove dve komponente i mora biti suprotnog smera od izlazne grane. Jedina takva grana je (a,d) i ona je ulazna grana. U ovom koraku $t = 64$ a $A = 6$. Posle pivotizacije dobijamo optimalno rešenje prikazano na Slici 2.15.



Slika 2.18: Mrežni protok nakon prve iteracije.

2.4.1 Dodatni primer - dualni simpleks

Rešiti problem mrežnog protoka (Slika 2.19) polazeći od pokrivajućeg drveta (a,d), (b,e), (c,f), (d,e), (d,f). Izračunali smo x,y,z vrednosti, one su prikazane na Slici 2.20.

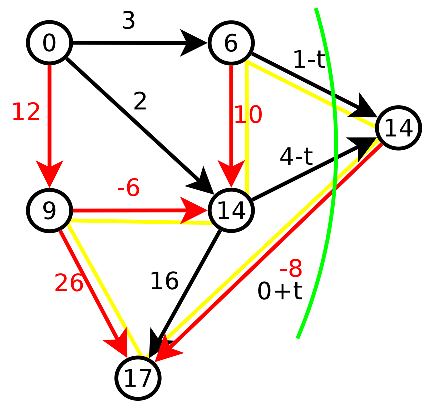


Slika 2.19: Graf sa cenama prevoza na granama i zalihama/potrebama pored čvorova.

Vidimo da je balansirani protok dualno dopustiv ($z \geq 0$) i nije primarno dopustiv ($x_{cf} = -8 < 0$).

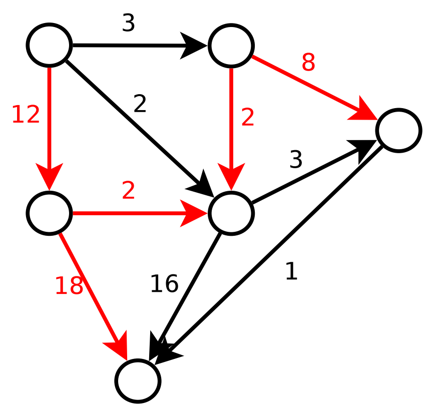
Vršimo dualnu pivotizaciju na grani (c,f). Izbacujemo granu (c,f) iz baze, skup čvorova se deli na dve grupe, na slici su odvojene zelenom granicom. Vrednost $z_{cf} = 0$ uvećavamo za t. Po pravilu mostova ažuriranje z vrednosti se vrši tako što ćemo vrednosti $z_{bc} = 1$ i $z_{ec} = 4$ umanjiti za t.

Biramo $t = 1$, grana (b,c) ulazi u bazu. Ako bi grana (b,c) ušla u bazu pre



Slika 2.20: Mrežni protok za dato pokrivajuće drvo koji nije primarno ali jeste dualno dopustiv. Pomoću pravila mostova i pravila kontura nalazimo da je vrednost $t = 1$ i $A = 8$.

nego što (c, f) izađe iz baze, nastala bi kontura $cfebc$ (žuto na Slici 2.20). Po pravilu kontura vrednosti transporta (x) po konturi se u jednom smeru povećavaju, u suprotnom smanjuju za vrednost $A = 8$. Nakon ažuriranje dobijamo optimalan protok (Slika 2.21).



Slika 2.21: Mrežni protok nakon prve iteracije je i primarno i dualno dopustiv, tj. optimalan je.

2.5 Parametarski self-dual simpleks

Ako polazno drvo nije primarno dopustivo ni dualno dopustivo, do rešenja ne možemo doći primenom primarne ili dualne simpleks metode. Zbog toga se izračunate vrednosti x i z perturbuju parametrom μ.

$$x_B^* + \mu \bar{x}_B,\ z_N^* + \mu \bar{z}_N, \text{ gde je } \bar{x}_B > 0,\ \bar{z}_N > 0.$$

Za svaki protok nalazimo oblast vrednosti μ za koje je primarno i dualno dopustiv i pišemo pored grafa. Pivotizaciju vršimo na grani gde je realizovana najmanja vrednost oblasti dopustivih μ.

$$\mu^* = \min\{\mu : z_N^* + \mu \bar{z}_N \geq 0 \ \text{ i } \ x_B^* + \mu \bar{x}_B \geq 0\}.$$

Pivotizacija može biti primarna ili dualna, zavisi od odabrane grane.

Za izbor vrednosti t na grani koja menja mesto sa izabranom granom, poređenje po veličini se vrši sa $\mu := \mu^*$.

U pivotizacijama vodimo μ u nomenklaturi sa zarezom: $a + b\mu = a, b$.

Kad dobijemo da oblast dopustivih vrednosti za μ sadrži 0, dobili smo optimalni protok (uvrštavanjem $\mu = 0$).

Ako uvrštavanje $\mu = 0$ daje primarno ili dualno dopustiv protok, možemo nastaviti sa primarnim, odnosno dualnim simpleks algoritmom.

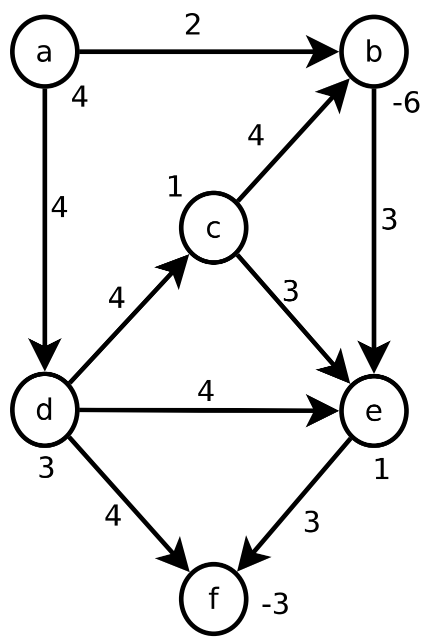
Najčešće uzimamo $\bar{x}_B = [1, \ldots, 1]^T$, $\bar{z}_N = [1, \ldots, 1]^T$, odnosno, pored x i z vrednosti dopišemo „,1".

Moše se dokazati da će se, bez degenerisanih pivotizacija, ako rešenje postoji, u konačnom broju koraka, doći do optimuma.

Posmatrajmo problem dat na Slici 2.22. Treba naći najjeftiniji transport polazeći od pokrivajućeg drveta (a,b), (c,b), (c,e), (d,e), (e,f) (crvene grane na Slici 2.23). Izračunate vrednosti x, y i z su prikazane na Slici 2.23 na redom crvenim granama, u kružićima i na crnim granama (x i z su brojevi pre zareza prikazani). Kako dobijeni protok (kada zanemarimo šta piše iza zareza) nije ni primarno ($x_{ce} < 0$) ni dualno ($z_{df} < 0$) dopustiv, primenjujemo parametarski self-dual simpleks algoritam.

Stoga vrednosti x i z perturbuju parametrom μ, tako što pored x i z vrednosti dopišemo „1". Posmatrajući grane koje su bile negativne dobijamo da $-3 + \mu \geq 0$ (na osnovu grane (d,f)) i $-1 + \mu \geq 0$ (na osnovu grane (c,e)) tj. dobijamo da $3 \leq \mu$. Grana na kojoj se ostvaruje najmanje μ (to je ovde 3) je grana (d,f). Ona je crna grana i ulazi u bazu tako što vršimo primarnu pivotizaciju - kada ona uđe u bazu sa crvenim granama obrazuje konturu (žuto na Slici 2.23), po pravilu kontura dobijamo da je $t = 3,1$, grana (d,e) izlazi iz baze, pravilo mostova nam daje $A = 3, -1$ pomoću kog ažuriramo vrednosti z (crne grane). Dobijamo nakon ažuriranja novi mrežni protok dat na Slici 2.24.

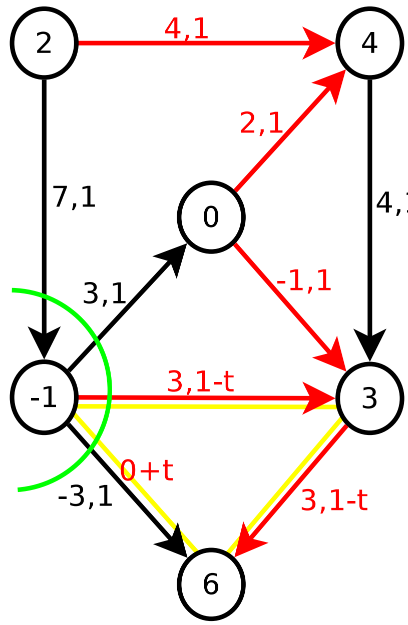
Za dobijeni protok dobijamo da vrednost parametra μ treba da bude $1 \leq \mu \leq 3$. Najmanja vrednost $\mu = 1$ se ostvaruje na grani (c,e) koja se trenutno nalazi u bazi (crvena je) te vršimo dualnu pivotizaciju. Data grana izlazi iz baze, njenim izlaskom se drvo raspada na dve povezne komponente (zelena granica



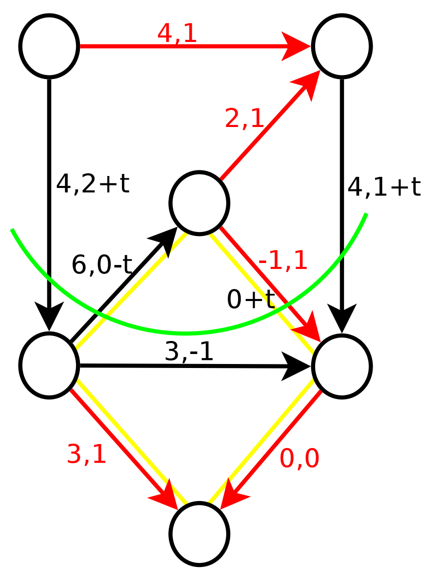
Slika 2.22: Graf sa zalihama pored čvorova i cenama transporta na granama.

na Slici 2.24). Pomoću pravila mostova dobijamo da je $t = 6,0$, a pomoću pravila kontura da je $A = 1,-1$.

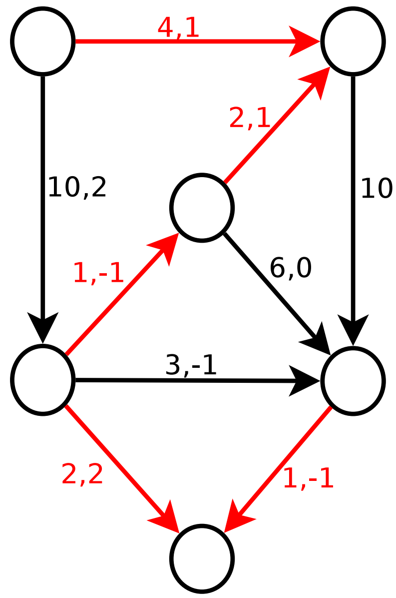
Sada posmatrajući vrednosti parametra μ dobijamo da $\mu = 0 \in [-1,1]$. Kako smo dobili da oblast dopustivih vrednosti za μ sadrži 0, dobili smo zapravo optimalni protok (uvrštavanjem $\mu = 0$).



Slika 2.23: Polazno pokrivajuće drvo (a,b), (c,b), (c,e), (d,e), (e,f) ne daje ni primarno i dualno dopustiv mrežni protok. U prvoj iteraciji dobijamo da je $3 \leq \mu$, $t = 3{,}1$, $A = 3,-1$.



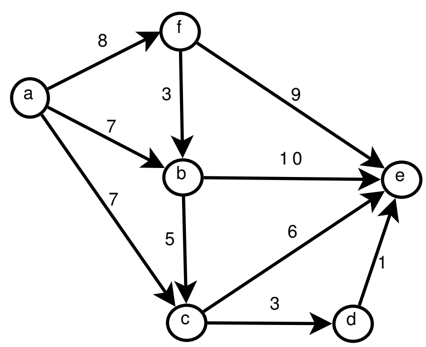
Slika 2.24: Mrežni protok $1 \leq \mu \leq 3$, $t = 6{,}0$, $A = 1,-1$.



Slika 2.25: Mrežni protok $-1 \leq \mu \leq 1$, Optimalan protok, $\zeta^* = 31$.

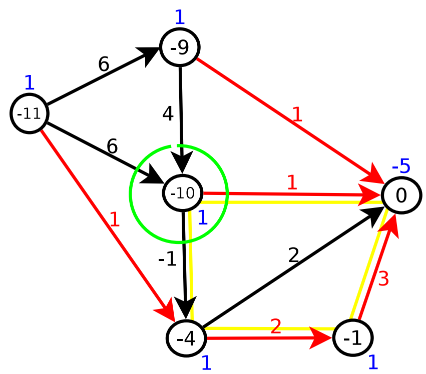
2.6 Najkraći put u mreži

Posmatramo problem pronalaska najkraćih puteva od čvorova a, b, c, d, f do čvora e ako su na Slici 2.26 date udaljenosti.



Slika 2.26: Problem pronalaska najkraćih puteva od čvorova a, b, c, d, f do čvora e. Pored grana pišu udaljenosti.

Polazno pokrivajuće drvo ima grane (a,c), (c,d), (d,e), (b,e), (f,e) (Slika 2.27). Problem nalaženja najkraćeg puta do jednog čvora (u našem primeru čvora e) se svodi na pronalaženje cene minimalnog transporta kroz mrežu, gde su cene transporta jednake udaljenostima i zaliha robe u svakom čvoru je jednaka 1, osim u odabranom čvoru do kog računamo udaljenost čija je potreba/narudžbina jednaka broju čvorova minus jedan (plavi brojevi pored kružića na Slici 2.27).



Slika 2.27: Prva iteracija primarnog simpleksa. Grana (b,c), pravilo kontura daje da je $t = 1$, grana (b,e) izlazi iz baze, pravilo mostova daje da je $A = 1$.

Posmatrani problem je primarno ali nije dualno dopustiv pa stoga primenjujemo primarni simpleks. Grana (b,c) ulazi u bazu, obrazuje konturu koja je prikazana žuto na Slici 2.27. Na osnovu pravila kontura dobijamo da je vrednost parametra $t = 1$ i pomoću nje unutar konture ažuriramo vrednosti promenljivih x na crvenim granama. Pravilo mostova nam daje da je $A = 1$ i vrednosti dualnih pomoćnih promenljivih z ažuriramo. Pravilo mostova čak ne moramo ni da koristimo, jedno kada nađemo vrednosti x, vrednosti promenljvih y i z možemo da ažuriramo kao što smo na početku računali koristeći jednačine:

$$-y_i + y_j = c_{ij}, \quad (i,j) \in \mathcal{T}.$$

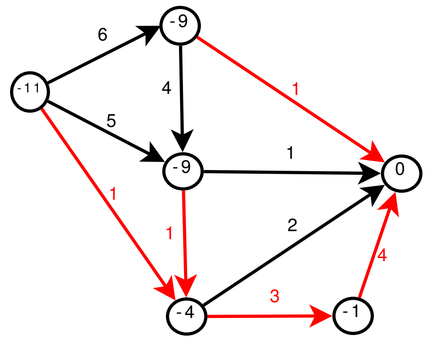
i

$$-y_i + y_j + z_{ij} = c_{ij} \quad \forall (i,j) \notin \mathcal{T}.$$

U ovom problemu moramo da ažuriramo i vrednosti promenljivih y jer će nam one dati odgovor na pitanje kolika je najkraća udaljenost svakog od čvorova do čvora e (on je uvek koren i $y_e = 0$). Nakon ažuriranja, dobijamo mrežni protok dat na Slici 2.28 koji je optimalan. Negativne vrednosti promenljivih y (-brojevi u kružićima) su najkraća rastojanja od pojedinih čvorova do posmatranog čvora e, i dati su u Tabeli 2.1.

a	b	c	d	e	f
11	9	4	1	0	9

Tabela 2.1: Najkraća rastojanja od pojedinih čvorova do posmatranog čvora e.



Slika 2.28: Optimalni mrežni protok (i primarno i dualno dopustiv). Negativne vrednosti promenljivih y (-brojevi u kružićima) su najkraća rastojanja od pojedinih čvorova do posmatranog čvora e. Napomena, na slici grana fb ima pogrešnu vrednost, umesto 4 treba da na njoj piše 3.

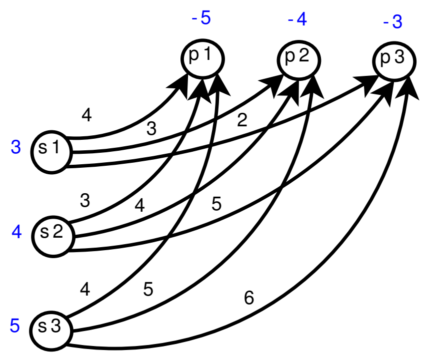
2.7 Hičkokov problem

Posmatrajmo problem gde imamo tri skladišta (snabdevača) s_1, s_2 i s_3, i tri potrošača (prodavnice) p_1, p_2 i p_3. Svaki snabdevač je povezan sa svakim potrošačem. Cene prevoza, potrebe tri potrošača i zalihe u tri skladišta date su u Tabeli 2.2. Cilj je naći optimalni (najjeftiniji tj. minimalne cene) plan transporta. Ovo je problem mrežnog protoka prikazan na Slici 2.29. Graf prikazan na Slici 2.29 se zove kompletni bipartitni graf tj. on ima osobinu da se čvorovi mogu podeliti u dve grupe - na one koji imaju zalihe, iz kojih polaze transporti (grane) i čvorove koji imaju potrebe, u njih ulaze transporti (grane). Dodatno, postoje grane od svakog skladišta do svake prodavnice. Ovo je **Hičkokov (Hitchcock) problem**.

	p_1	p_2	p_3	zalihe
s_1	4	3	2	3
s_2	3	4	5	4
s_3	4	5	6	5
potrebe	5	4	3	

Tabela 2.2: Cene prevoza, potrebe tri potrošača i zalihe u tri skladišta.

Potreban uslov da problem minimizacije cene protoka ima rešenje je da su suma zaliha i suma potreba jednake. Ako ima viška zaliha, možemo uvesti fiktivnog potrošača; ako ima pak viška potreba, možemo uvesti fiktivnog snabdevača. Cene ka njima i od njih se stavljaju da su 0. Jedno primarno dopustivo rešenje Hičkokovog problema se može dobiti metodom severozapadnog ugla ili Vogelovom metodom (o ovim dvema metodama će biti više reči u nastavku teksta). Posledica toga je da je dovoljan uslov da Hičkokov problem ima optimalno rešenje je da je zbir zaliha jednak zbiru potreba.



Slika 2.29: Mrežni protok koji opisuje transportni problem čiji podaci su dati u Tabeli 2.2.

2.7.1 Matematička formulacija Hičkokovog problema

Neka su zalihe u skladištima označene sa $a_1, a_2, \dots, a_m$, neka su potrebe u prodavnicama $b_1, b_2, \dots, b_n$ i neka su c_{ij} cene prevoza od skladišta i do prodavnice j, za $i = 1,2,\dots m$, $j = 1,2,\dots n$. Neka je x_{ij} količina robe koja se prevozi od skladišta i do prodavnice j. Ukupna cena prevoza je ζ. Sledeći problem linearnog programiranja opisuje Hičkokovov problem:

$$\zeta = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \to \min$$

$$\sum_{j=1}^{n} x_{ij} = a_i,\ i \in \{1,2,\dots,m\},$$

$$\sum_{i=1}^{m} x_{ij} = b_j,\ j \in \{1,2,\dots,n\},$$

$$x_{ij} \geq 0, \quad x_{ij} \in \mathbb{Z},\ i \in \{1,2,\dots,m\},\ j \in \{1,2,\dots,n\}.$$

Ovo je problem celobrojnog linearnog programiranja. Umesto na grafovima (kao što smo do sada rešavali probleme mrežnih protoka) radi jednostavnijeg prikaza, ovakve probleme ćemo rešavati pomoću tabela transportnog problema.

2.7.2 Tabele transportnog problema

U polja tabele unosimo x_{ij} ili z_{ij}, zavisno da li je grana od i-tog snabdevača do j-tog potrošača primarno bazična (crvena) ili nebazična (crna). Vrednosti z_{ij} upisujemo u zagrade da bismo lakše pratili koja polja sadrže x vrednosti a koja z vrednosti. U gornji levi ugao svakog polja upisujemo cenu c_{ij} transporta, a na marginama (u kolonu skroz desno i vrstu skroz dole) y vrednosti.

2.7.3 Algoritam severozapadnog ugla

Algoritam severozapadnog ugla nam služi da odredimo polazne protoke koji najčešće nisu primarno ni dualno dopustivi i nakon njega slede pivotizacije.

1. Poći od polja $(i,j) := (1,1)$
2. $x_{ij} := \min(a_i, b_j)$; $a_i := a_i - x_{ij}$; $b_j := b_j - x_{ij}$;
3. `IF` $a_i = 0$ `THEN` pređi u polje $(i+1, j)$
`ELSE` pređi u polje $(i, j+1)$;
4. Ponavljaj korake 2. do 3. i dođi do poslednjeg polja gde se moraju složiti preostala potreba i zaliha.

U našem primeru polazimo od gornjeg levog ugla (polje $(1,1)$) tabele:

4 3	3	2	3
3 2	4 2	5	4
4	5 2	6 3	5
5	4	3	

i u njega upisujemo minimalnu vrednost od $a_1 = 3$ i $b_1 = 5$ (brojevi koji odgovaraju zalihama i potrebama prvog snabdevača i prvog potrošača, tj. to su brojevi na kraju prve vrste i prve kolone). Minimum od 3 i 5 je 3 i to je polazna vrednost x_{11}. Nove zalihe i potrebe su jednake starim umenjenim za 3. Pošto smo istrošili sve zalihe (sada je vrednost zaliha u prvoj vrsti jednaka nuli), onda prelazimo na sledeće polje a to je polje ispod (polje $(2,1)$). Uvek se krećemo po tabeli tako što idemo dole (kada u posmatranoj vrsti nema više zaliha) ili desno (ako još ima zaliha). Kada dođemo u polje $(2,1)$, njemu dodeljujemo minimum od preostalih potreba u datoj koloni (ostalo je $5 - 3 = 2$) i zaliha u datoj vrsti 4. Minimum je 2 i time smo istrošili potrebe u prvoj koloni, dok su zalihe u drugoj vrsti jednake $4 - 2 = 2$. Kako zalihe u datoj vrsti još nisu istrošene, prelazimo na polje desno tj. polje $(2,2)$ i njemu dodeljujemo minimum od 2 (preostale zalihe u drugoj vrsti) i 4 potrebe u drugoj koloni što iznosi 2. Sada smo istrošili zalihe u drugoj vrsti pa prelazimo na polje dole, tj. $(3,2)$ i njemu dodeljujemo minimum od preostalih potreba $4 - 2 = 2$ i zaliha 5. Minimum je 2, time smo istrošili potrebe u drugoj koloni (potrebe drugog potrošača), dok su zalihe u trećoj koloni jednake $5 - 2 = 3$. Pomeramo se jedno polje desno i dodeljujemo preostalu vrednost 3, čime su se složile preostale potrebe i preostala zaliha. Time smo izračunali vrednosti primarnih promenljivih x, i u nastavku slede pravila za računanje vrednosti dualnih promenljivih y i pomoćnih dualnih promenljivih z.

2.7.4 Test optimalnosti

1. Biramo jedno y da je nula,

2. Računamo ostale y vrednosti tako da je za bazične promenljive $y_j - y_i = c_{ij}$

3. Računamo z vrednosti za polja koja nisu bazična tako da je $y_j - y_i + z_{ij} = c_{ij}$

4. `IF` $\forall(i,j) z_{ij} \geq 0$ `THEN` optimum `ELSE` negativan z_{ij} je pivot

Prvo računamo u našem primeru vrednosti y pa tek onda z. Vrednosti y pišemo u poslednju kolonu i poslednju vrstu. Jedno y je jednako nuli (proizvoljno i to najčešće je y u vrsti u kojoj ima najviše dodeljnih x vrednosti, to je ovde npr. y iz treće vrste). Ostale y vrednosti dobijamo tako što koristimo bazična polja (plave brojeve) i za njih treba da važi da je "y vrednost ispod tog polja - y vrednost desno = cena (gornji levi ugao posmatranog polja)". Na primer, za polje $(3,2)$ treba da važi da je njegovo y ispod minus y desno (što je nula) jednako ceni (5), tako da je y ispod tog polja zapravo jednako 5. Iz istog razloga ako posmatramo polje $(3,3)$, dobijamo da je y ispod tog polja jednako 6. Dalje, posmatramo polje $(2,2)$, njegovo y ispod minus y desno treba da bude jednako ceni 4. Kako je y ispod jednako 5, dobijamo da je y desno jednako 1. Zatim posmatramo polje $(2,1)$ i dobijamo da je y ispod koristeći jednačine za dobijanje y vrednosti jednako 4. Na kraju pomoću polja $(1,1)$ dobijamo y u prvoj vrsti (desno od posmatranog polja). Dobijeni y su 0, 1, 0 (poslednja kolona) i 4, 5, 6 (poslednja vrsta).

4 3	3 (-2)	2 $(-\mathbf{4})$	0
3 2	4 2	5 (0)	1
4 (0)	5 2	6 3	0
4	5	6	

Na kraju računamo vrednosti z koje pišemo u zagradama. Njih dobijamo uz pomoć jednačina $y_j - y_i + z_{ij} = c_{ij}$. Na primer, za polje $(1,2)$ imamo da je "y ispod - y desno + nepoznato z_{12} = cena iz gornjeg levog ugla tog polja" tj. "5 - 0 + nepoznato z_{12} = 3" odakle dobijamo da je $z_{12} = -2$. Slično dobijamo ostale z vrednosti. Kako nisu sve z vrednosti nenegativne (npr. $z_{12} = -2$), dobijeno rešenje nije dualno dopustivo (samim tim nije optimalno) te moramo vršiti pivotizacije dok ne dobijemo optimalno rešenje. Za transportni problem, pošto je u pitanju bipartitni graf a polja u tabeli predstavljaju grane grafa, pivotizaciju ćemo prilagoditi radu u tabelama. Pivot je polje sa najnegativnijom z vrednošću a to je u ovom primeru $z_{13} = -4$.

2.7.5 Pivotizacija

1. Sastaviti cikl od pivota i polja baz̆ičnih promenljivih.
2. Polazeći od pivota, idući po ciklu, naizmenično dodeliti znak $+$ i $-$ poljima cikla.
3. Od polja koja su dobila $-$ naći polje sa najmanjom vrednošću koja iznosi t.
4. Polazeći od pivota, idući po ciklu, naizmenično dodati i oduzeti vrednost t promenljivama x_{ij} u ciklu.
5. Jedno od polja koje je u prethodnom koraku dobilo vrednost 0 izbaciti iz baze, a pivota ubaciti u bazu.

U našem primeru, pivot je kao što smo rekli polje $(1,3)$ i ono sa baz̆ičnim poljima obrazuje cikl (crevna kontura na slici ispod). Cikl čine u ovoj tabeli pivot i sva bazična polja. Dodajemo naizmenično znak $+$ i $-$ poljima cikla polazeći od pivota i krećući se po ciklu. Od polja koja su dobila $-$ nalazimo polje sa najmanjom vrednošću koja iznosi $t = 2$. Polazeći od pivota, idući po ciklu, naizmenično dodajemo i oduzimamo vrednost $t = 2$ promenljivama x_{ij} u ciklu.

4 $-$ 3	3 (-2)	2 $+$ $(-\mathbf{4})$	0
3 $+$ 2	4 $-$ 2	5 (0)	1
4 (0)	5 $+$ 2	6 $-$ 3	0
4	5	6	

Na taj način dobijamo nove bazične promenljive, nakon čega moramo ponovo da računamo y i z vrednosti kao na početku. Nakon pivotizacije dobijamo sledeću tabelu:

4 1	3 (2)	2 2	4
3 4	4 (4)	5 (4)	5
4 $(-\mathbf{4})$	5 4	6 1	0
8	5	6	

Opet dobijena tabela nije optimalna i uzimamo za pivota $z_{31} = -4$ koja si bazičnim promenljivama čini cikl (crvena kontura na slici ispod, na osnovu nje dobijamo da je $t = 1$), nova pivotizacija omogućava da iz baze izbacimo x_{11} ili x_{33}. Biramo x_{11}.

4 − 1	3 (2)	2 + 2	4
3 4	4 (4)	5 (4)	5
4 + (−4)	5 4	6 −1	0
8	5	6	

Dobijamo nakon pivotizacije tabelu:

4 (4)	3 (2)	2 3	4
3 4	4 (0)	5 (0)	1
4 1	5 4	6 0	0
4	5	6	

Više nema negativnih z vrednosti, dobili smo dualno dopustivu, odnosno optimalnu, tabelu. Rešenje nije jedinstveno, ima nula z vrednosti. Najjeftiniji transport ima cenu $\zeta = 2 \times 3 + 3 \times 4 + 4 \times 1 + 5 \times 4 + 6 \times 0 = 42$.

2.7.6 Vogelova metoda

Vogelova metoda nam služi da odredimo polazne protoke koji najčešće nisu primarno ni dualno dopustivi i nakon čega slede pivotizacije. To je alternativa metodi severozapadnog ugla i njome se često brže dođe do optimalnog rešenja.

1. Za sve vrste i kolone izračunati apsolutnu vrednost razlike dve najmanje nedodeljene cene transporta.

2. U vrsti ili koloni koja daje najveću razliku otići u polje sa najmanjom nedodeljenom cenom transporta i dodeliti $x_{ij} := \min(a_i, b_j)$; $a_i := a_i - x_{ij}; b_j := b_j - x_{ij}$.

3. Ponavljaj korake 1. do 3. sve dok ne ostane samo jedna vrsta ili kolona.

4. Dodeliti nedodeljene zalihe odnosno potrebe preostalim (baznim) promenljivama.

5. Ako u bazi nema $n + m - 1$ elemenata, dodati potreban broj elemenata, ali tako da ne zatvaraju cikl sa postojećim baznim promenljivama.

Rešićemo naredni primer upotrebom Vogelove metode. Cene prevoza, potrebe četiri prodavnice i zalihe u četiri skladišta date su u tabeli. Naći optimalni plan transporta.

	p_1	p_2	p_3	p_4	zalihe
s_1	4	5	6	3	7
s_2	5	4	6	6	8
s_3	7	8	9	6	10
s_4	10	12	15	8	15
potrebe	10	10	12	8	

Radi bržeg dolaska do optimalnog rešenja, polazno rešenje ćemo naći Vogelovom metodom (njom nalazimo x vrednosti a y i z vrednosti računamo kao i ranije).

Dobijamo polaznu tabelu:

4 (-1)	5 (-2)	6 $(-\mathbf{4})$	3 7	5
5 (3)	4 8	6 (-1)	6 (6)	8
7 (3)	8 (2)	9 10	6 (4)	6
10 10	12 2	15 2	8 1	0
10	12	15	8	

Nije optimalna tabela, pivot je $z_{13} = -4$, cikl čine polja $(1,3)$, $(1,4)$, $(4,4)$ i $(4,3)$, $t = 2$. Nakon pivotizacije dobijamo tabelu:

4 (-1)	5 $(-\mathbf{2})$	6 2	3 5	5
5 (3)	4 8	6 (3)	6 (6)	8
7 (-1)	8 (-2)	9 10	6 (0)	2
10 10	12 2	15 (4)	8 3	0
10	12	11	8	

Dobijena tabela nije optimalna, vršimo pivotizaciju - pivot je $z_{12} = -2$, cikl čine polja $(1,2)$, $(1,4)$, $(4,4)$ i $(4,2)$, $t = 2$. Nakon pivotizacije dobijamo tabelu:

4 (-1)	5 2	6 2	3 3	0
5 (1)	4 8	6 (1)	6 (4)	1
7 $(-\mathbf{1})$	8 (0)	9 10	6 (0)	-3
10 10	12 (2)	15 (4)	8 5	-5
5	5	6	3	

Dobijena tabela nije optimalna, vršimo pivotizaciju - pivot je $z_{31} = -1$, cikl čine polja $(3,1)$, $(4,1)$, $(4,4)$, $(1,4)$, $(1,3)$ i $(3,3)$, $t = 3$. Nakon pivotizacije dobijamo optimalnu tabelu za koju je $\zeta = 290$.

4 (0)	5 2	6 5	3 (1)	0
5 (2)	4 8	6 (1)	6 (5)	1
7 3	8 (0)	9 7	6 (1)	-3
10 7	12 (1)	15 (3)	8 8	-6
4	5	6	2	

2.8 Problem angažovanja

Trener plivačke reprezentacije ima za štafetu $4 \times 100m$ na raspolaganju četiri plivača A, B, C i D čija su vremena na $100m$ po stilovima: slobodno, leđno, prsno, baterflaj:

	S	L	P	B
A	57	61	64	62
B	55	63	65	64
C	59	64	66	63
D	56	62	67	64

Kako da sastavi najbolju štafetu?

Ovo je transportni problem: plivači imaju na raspolaganju jedno plivanje koje treba da „prebace" na određenu stazu, cena transporta je jednaka vremenu koje plivaču treba da ispliva. Svaku stazu može plivati jedan plivač, jedan plivač treba da pliva jednu stazu. Ukupna cena transporta je jednaka ukupnom vremenu štafete. Vrednosti x_{ij} su 1 ako plivač i pliva stazu j, inače su 0. Polazno rešenje možemo naći Vogelovom metodom:

57 (2)	61 0	64 1	62 0	0
55 1	63 (2)	65 (1)	64 (2)	0
59 (3)	64 (2)	66 (1)	63 1	-1
56 0	62 1	67 (2)	64 (1)	-1
55	61	64	62	

Vidimo da je transport optimalan, tako da plivači plivaju:
$A \to$ prsno, $B \to$ slobodno, $C \to$ baterflaj, $D \to$ leđno. Ukupno vreme štafete je 244.

Glava 3

Slučajni procesi

Kažemo da je $\{X(t) : t \in [0,+\infty)\}$ **slučajni proces** ako je X funkcija $X : \Omega \times [0,+\infty) \to \mathbb{R}$, koja je za svako fiksirano $t \in [0,+\infty)$ slučajna promenljiva $X(t) : \Omega \to \mathbb{R}$.

Ako za sve $t \in [0,+\infty)$ važi $X(t) : \Omega \to \mathcal{R}_X$, gde je $\mathcal{R}_X$ najviše prebrojiv skup ($\mathcal{R}_X = \{x_1, x_2, \ldots\}$), kažemo da je slučajni proces $\{X(t) : t \in [0,+\infty)\}$ **diskretno vredan**. Tada je za proizvoljno $t \in [0,+\infty)$ slučajna promenljiva $X(t)$ diskretna. Skup $\mathcal{R}_X$ zovemo **skup stanja**.

3.1 Homogeni, diskretno vredni Markovljevi procesi

Kažemo da je diskretno vredan slučajni proces $\{X(t) : t \in [0,+\infty)\}$ **Markovljev**, ili da ima osobinu Markova, ako za sve $x_1, x_2, \ldots, x_n, x_{n+1}$ iz skupa stanja $\mathcal{R}_X$ i za sve $t_1 \le t_2 \le \cdots \le t_n \le t_{n+1}$ iz skupa $[0,+\infty)$

$$P\left(X(t_{n+1}) = x_{n+1} | X(t_1) = x_1, X(t_2) = x_2, \ldots, X(t_n) = x_n\right) =$$

$$= P\left(X(t_{n+1}) = x_{n+1} | X(t_n) = x_n\right) .$$

Posmatramo za $t_1 < t_2$ i $x_1, x_2 \in \mathcal{R}_X$ **verovatnoće prelaza**

$$p(t_1, t_2, x_1, x_2) :=$$

$$P(X(t_2) = x_2 | X(t_1) = x_1) .$$

Procese kod kojih verovatnoća $p(t_1, t_2, x_1, x_2)$ ne zavisi od t_1 i t_2 već samo od njihove razlike $t_2 - t_1$:

$$P(X(t_2) = x_2 | X(t_1) = x_1) = p(t_2 - t_1, x_1, x_2)$$

nazivamo **homogenim**. Ubuduće posmatramo homogene, diskretno vredne, Markovljeve procese.

Bez gubitka opštosti, uzimamo da je skup stanja $\mathcal{R}_X = \{0,1,2,\dots\}$.
Označavamo za $t \in [0,+\infty)$

$$p_{ij}(t) = P(X(s+t) = j | X(s) = i),$$

Vrednost $p_{ij}(t)$ tumačimo kao verovatnoću da sistem za vreme t pređe iz stanja i u stanje j. Zbog homogenosti, ta verovatnoća je ista sve s. Matrica

$$P(t) = \begin{bmatrix} p_{0,0}(t) & p_{0,1}(t) & p_{0,2}(t) & \cdots \\ p_{1,0}(t) & p_{1,1}(t) & p_{1,2}(t) & \cdots \\ p_{2,0}(t) & p_{2,1}(t) & p_{2,2}(t) & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

se naziva **matrica verovatnoća prelaza**.
Lako je videti da je $P(0) = I$.
Raspodela verovatnoće u momentu t je:

$$X(t) : \left(\begin{array}{cccc} 0 & 1 & 2 & \cdots \\ p_0(t) & p_1(t) & p_2(t) & \cdots \end{array} \right),$$

gde je $p_k(t) = P(X(t) = k)$, za poznate vrednosti $p_k(0), k = 0,1,\dots$. Verovatnosni vektor $p(t) = (p_0(t), p_1(t), \dots)$ nazivamo **vektor stanja**.

3.1.1 Jednačine Čepmen–Kolmogorov

$$\forall s,t \geq 0 \; P(s+t) = P(s)P(t) , \quad p(s+t) = p(s)P(t) .$$

3.1.2 Brzine prelaza

Definišemo **brzine prelaza** $\lambda_{ij} = \lim\limits_{t \to 0_+} \dfrac{p_{ij}(t) - p_{ij}(0)}{t} = p'_{ij}(0)$.

Matricu $\Lambda = \begin{bmatrix} \lambda_{0,0} & \lambda_{0,1} & \lambda_{0,2} & \cdots \\ \lambda_{1,0} & \lambda_{1,1} & \lambda_{1,2} & \cdots \\ \lambda_{2,0} & \lambda_{2,1} & \lambda_{2,2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$ nazivamo **matrica brzina prelaza**.

Lako se vidi da su elementi matrice Λ na dijagonali nepozitivni, a van dijagonale nenegativni.

Zato što je $P(t)$ verovatnosna matrica (svi elementi su nenegativni brojevi i zbir po vrstama je 1), zbir elemenata po vrstama matrice Λ je 0.

3.1.3 Diferencijalne jednačine Čepmen–Kolmogorov

$$P'(t) = P(t) \cdot \Lambda .$$

$$p'(t) = p(t) \cdot \Lambda \,.$$

Oznaka za izvod vektora ili matrice $p'(t)$ podrazumeva da se prvi izvod uzima po svim komponentama. Dobija se sistem dif. jed.

3.1.4 Procesi rađanja i umiranja

Za Markovljev, homogen, diskretno vredan slučajni proces čija matrica brzina prelaza je oblika

$$\Lambda = \begin{bmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & \cdots \\ \mu_1 & -(\lambda_1+\mu_1) & \lambda_1 & 0 & \cdots \\ 0 & \mu_2 & -(\lambda_2+\mu_2) & \lambda_2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

gde su $\lambda_0 \geq 0$, $\lambda_1 \geq 0$, $\ldots$, $\mu_1 \geq 0$, $\mu_2 \geq 0$, $\ldots$, kažemo da je **proces rađanja i umiranja**.

Za ovakve procese sa $X(t)$ obeležavamo veličinu populacije u momentu t. U pitanju mogu biti biološke populacije, klijenti u sistemu opsluživanja, kvarovi u nekom mehanizmu,...

Veličine λ_j se tumače kao brzine rađanja jedne jedinke za populaciju čija veličina je j. Brojevi μ_j se tumače kao brzine umiranja jedne jedinke u populaciji veličine j.

Ergodičnost

Kažemo da je proces rađanja i umiranja **ergodičan** ako

$$\forall j \in \mathcal{R}_X,\ \exists p_j^*,\ \forall i \in \mathcal{R}_X,\ \lim_{t\to\infty} p_{i,j}(t) = p_j^*.$$

Ergodični procesi posle dugog vremena ($t \to \infty$) „zaboravljaju” iz kog stanja su krenuli, vidimo po tome što je vrednost limesa ista za sve i.

Posledica ergodičnosti je da i vektori stanja teže ka ergodičnim verovatnoćama p^*:

$$\forall j \in \mathcal{R}_X, \exists p_j^*, \forall i \in \mathcal{R}_X, \lim_{t\to\infty} p_j(t) = p_j^*.$$

Ako se na diferencijalnu jednačinu Čepmen-Kolmogorov za ergodičan proces rađanja i umiranja primeni granični proces sa $t \to \infty$, ona prelazi u algebarski sistem jednačina

$$p'(t) = p(t) \cdot \Lambda \quad / \left(\lim_{t\to\infty}\right)$$

$$0 = p^* \cdot \Lambda \,,\ \text{gde je } p^* = (p_0^*, p_1^*, \ldots)$$

Taj sistem jednačina glasi:

$$\begin{aligned} 0 &= \mu_1 p_1^* - \lambda_0 p_0^*, \\ 0 &= \mu_2 p_2^* - (\lambda_1+\mu_1) p_1^* + \lambda_0 p_0^*, \\ 0 &= \mu_3 p_3^* - (\lambda_2+\mu_2) p_2^* + \lambda_1 p_1^*, \\ &\cdots \end{aligned}$$

Iz prve jednačine izrazimo p_1^* preko p_0^*: $p_1^* = \frac{\lambda_0}{\mu_1} p_0^*$ i to ubacimo u drugu jednačinu: $0 = \mu_2 p_2^* - (\lambda_1 + \mu_1) \frac{\lambda_0}{\mu_1} p_0^* + \lambda_0 p_0^*$.

Zatim izrazimo p_2^* preko p_0^*: $p_2^* = \frac{\lambda_0 \lambda_1}{\mu_1 \mu_2} p_0^*$.

Nastavljajući ovaj postupak dobijamo

$$p_k^* = \frac{\lambda_{k-1}}{\mu_k} p_{k-1}^* = \frac{\lambda_0 \lambda_1 \cdots \lambda_{k-1}}{\mu_1 \mu_2 \cdots \mu_k} p_0^*, \quad k = 1, 2, \ldots.$$

Za ergodične verovatnoće treba da važi: $p_0^* + p_1^* + \cdots = 1$. Dobijene relacije ubacimo u ovu jednakost i dobijamo

$$p_0^* \underbrace{\left(1 + \frac{\lambda_0}{\mu_1} + \frac{\lambda_0 \lambda_1}{\mu_1 \mu_2} + \cdots\right)}_{s_0} = 1\,.$$

Očigledno je da je proces rađanja i umiranja ergodičan samo ako je $s_0 < \infty$, odnosno, ako suma u zagradi konvergira. Tada je $p_0^* = 1/s_0$, a ostale ergodične verovatnoće se dobijaju iz relacija:

$$p_1^* = \frac{\lambda_0}{\mu_1} p_0^*, p_2^* = \frac{\lambda_0 \lambda_1}{\mu_1 \mu_2} p_0^*, p_3^* = \frac{\lambda_0 \lambda_1 \lambda_2}{\mu_1 \mu_2 \mu_3} p_0^*, \ldots$$

Poasonov slučajni proces

Proces rađanja i umiranja čija je matrica brzina prelaza

$$\Lambda = \begin{bmatrix} -\lambda & \lambda & 0 & 0 & \cdots \\ 0 & -\lambda & \lambda & 0 & \cdots \\ 0 & 0 & -\lambda & \lambda & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad \text{za } \lambda > 0,$$

zovemo **Poasonov slučajni proces**.

Ako se ovaj proces inicijalizuje iz stanja 0, sistem diferencijalnih jednačina Čepmen-Kolmogorov:

$$\begin{aligned} p_0'(t) &= - \lambda p_0(t),\ p_0(0) = 1, \\ p_1'(t) &= \lambda p_0(t) - \lambda p_1(t),\ p_1(0) = 0, \\ p_2'(t) &= \lambda p_1(t) - \lambda p_2(t),\ p_2(0) = 0,\ \ldots \end{aligned}$$

daje rešenje $p_k(t) = \dfrac{(\lambda t)^k}{k!} e^{-\lambda t}$, $k = 0, 1, 2, \ldots$

Broj jedinki u momentu t ima Poasonovu raspodelu: $X(t) : \mathcal{P}(\lambda t)$.

Očekivana veličina populacije u momentu t je $E(X(t)) = \lambda t$.

Ako se proces inicijalizuje iz stanja $n > 0$ jedinki, verovatnoće su iste, samo „pomerene” za n:

$$\begin{cases} p_0(t) = p_1(t) = \cdots = p_{n-1}(t) = 0, \\ p_{n+k}(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t},\ k = 0, 1, 2, \ldots \end{cases}$$

Tada je očekivana veličina populacije u momentu t: $E(X(t)) = n + \lambda t$.

Linearni proces čistog rađanja

Ako su brzine umiranja nula: $\mu_k = 0$, $k = 1, 2, \ldots$, a brzine rađanja proporcionalne veličini populacije: $\lambda_k = \lambda \cdot k$, za $\lambda > 0$, dobijamo **linearni proces čistog rađanja**.

Ovaj proces nema smisla inicijalizovati iz populacije veličine 0, zato uzimamo da je početna veličina populacije 1.

$$\Lambda = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & -\lambda & \lambda & 0 & 0 & \cdots \\ 0 & 0 & -2\lambda & 2\lambda & 0 & \cdots \\ 0 & 0 & 0 & -3\lambda & 3\lambda & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \ \lambda > 0.$$

Veličinu λ zovemo natalitet.

Sistem diferencijalnih jednačina Čepmen-Kolmogorov:

$$\begin{aligned} p_0'(t) &= 0,\ p_0(0) = 0, \\ p_1'(t) &= -\lambda p_1(t),\ p_1(0) = 1, \\ p_2'(t) &= \lambda p_1(t) - 2\lambda p_2(t),\ p_2(0) = 0, \\ p_3'(t) &= 2\lambda p_2(t) - 3\lambda p_3(t),\ p_3(0) = 0, \ldots \end{aligned}$$

daje rešenje $p_k(t) = \left(1 - e^{-\lambda t}\right)^{k-1} e^{-\lambda t}$, $k = 0, 1, 2, \ldots$ Broj jedinki u momentu t ima Geometrijsku raspodelu: $X(t) : \mathcal{G}(e^{-\lambda t})$.

Očekivana veličina populacije u momentu t je $E(X(t)) = e^{\lambda t}$.

Ako se ovaj proces inicijalizuje iz stanja od n jedinki, posmatramo ga kao n nezavisnih procesa inicijalizovanih iz stanja 1 jedinke.

Tada je očekivano stanje sistema $E(X(t)) = ne^{\lambda t}$.

Linearni proces čistog umiranja

Ako su brzine rađanja nula: $\lambda_k = 0$, $k = 0, 1, \ldots$, a brzine umiranja proporcionalne veličini populacije: $\mu_k = \mu \cdot k$, za $\mu > 0$, dobijamo **linearni proces čistog umiranja**.

Ovaj proces inicijalizujemo iz populacije veličine n. Nema rađanja, zato nas interesuje samo konačna matrica brzina prelaza: sa stanjima od 0 do n:

$$\Lambda = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \mu & -\mu & 0 & 0 & \cdots & 0 & 0 \\ 0 & 2\mu & -2\mu & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & n\mu & -n\mu \end{bmatrix}, \ \mu > 0.$$

Veličinu μ zovemo mortalitet.

Sistem diferencijalnih jednačina Čepmen-Kolmogorov:

$$\begin{array}{lcll} p_0'(t) & = & \mu p_1(t), & p_0(0)=0, \\ p_1'(t) & = & 2\mu p_2(t) - \mu p_1(t), & p_1(0)=0, \\ & & \cdots & \\ p_{n-1}'(t) & = & n\mu p_n(t) - (n-1)\mu p_{n-1}(t), & p_{n-1}(0)=0 \\ p_n'(t) & = & -n\mu p_n(t), & p_n(0)=1. \end{array}$$

Rešavanje sistema nam daje

$$p_k(t) = \binom{n}{k}\left(1-e^{-\mu t}\right)^{n-k}\left(e^{-\mu t}\right)^k.$$

Slučajna promenljive $X(t)$ ima Binomnu raspodelu $\mathcal{B}\left(n, e^{-\mu t}\right)$.

Očekivana veličina populacije u momentu t je $E(X(t)) = ne^{-\mu t}$.

Zadatak

Linearni Markovljev proces čistog rađanja sa natalitetom $\lambda = 2$ inicijalizovan je iz stanja jedne jedinke.

Izračunati verovatnoću da populacija bude veća od 3 posle jednog sata.

Kolika je očekivana veličina populacije posle jednog sata?

Rešenje

Koristimo formule za linearni proces čistog rađanja. Imamo $\lambda = 2,\ t = 1$. Uvodimo smenu

$$p = e^{-\lambda t} = e^{-2\cdot 1} = 0.135335, \quad q = 1-p = 0.864665\,.$$

Tražena verovatnoća je

$$P(X(1)>3) = 1-P(X(1)\le 3) = 1-(p_0(1)+p_1(1)+p_2(1)+p_3(1)) = 1-(0+p+qp+q^2p)$$

$$= 1-p(1+q+q^2) = 1-0.135335(1+0.864665+0.864665^2) = 0.6465$$

Očekivana veličina populacije je $e^{\lambda t} = e^2 = 7.3891$.

Zadatak

Rešiti prethodni zadatak ako je u pitanju Poasonov proces sa $\lambda = 2$.

Rešenje

Koristimo formule za pomereni Poasonov proces. Tražena verovatnoća je

$$P(X(1)>3) = 1-P(X(1)\le 3) = 1-(p_0(1)+p_1(1)+p_2(1)+p_3(1))$$

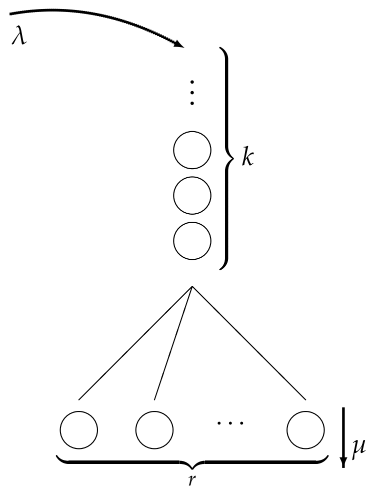
$$= 1 - (0 + e^{-\lambda t} + \lambda t e^{-\lambda t} + \frac{(\lambda t)^2}{2} e^{-\lambda t}) = 1 - e^{-\lambda t}(1 + \lambda t + \frac{(\lambda t)^2}{2})$$

$$= 1 - 0.135335(1 + 2 + \frac{2^2}{2}) = 0.3233$$

Očekivana veličina populacije kroz 1 sat je

$$E(X(1)) = 1 + \lambda t = 1 + 2 \cdot 1 = 3$$

3.1.5 Sistemi masovnog opsluživanja M|M|r|k|FIFO



Posmatramo sisteme masovnog opsluživanja sa r pribora za opsluživanje i maksimalnom dužinom reda k. Klijenti pristižu u sistem sa vremenom između dva pristizanja raspoređenom po eksponencijalnoj raspodeli $\mathcal{E}(\lambda)$, nezavisno. Ako su svi pribori za opsluživanje zauzeti, klijenti čekaju u zajedničkom redu po redosledu pristizanja (FIFO = *First In First Out*). Postoje i druge discipline reda, ako je disciplina izostavljena, podrazumeva se FIFO.

Ako je $k = \infty$, onda se u Kendalovoj notaciji ne piše. Tako bi M|M|2 bila oznaka za sistem masovnog opsluživanja sa dva pribora za opsluživanje i beskonačnim redom za čekanje.

Vreme opsluživanja na svakom priboru je nezavisno od klijenta do klijenta i ima eksponencijalnu raspodelu $\mathcal{E}(\mu)$.

Proces pristizanja klijenata u sistem je Poasonov sa parametrom λ, a opsluživanje klijenata na jednom priboru (kad nema čekanja na pristizanje klijenta) je Poasonov proces sa parametrom μ.

Ako klijente sistema posmatramo kao jedinke populacije, onda je ovo primer procesa rađanja i umiranja čija matrica brzina prelaza je

$$\Lambda = \begin{bmatrix} -\lambda & \lambda & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots \\ \mu & -(\lambda+\mu) & \lambda & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots \\ 0 & 2\mu & -(\lambda+2\mu) & \lambda & \cdots & 0 & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \\ 0 & 0 & 0 & 0 & \cdots & r\mu & -(\lambda+r\mu) & \lambda & 0 & \cdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & r\mu & -(\lambda+r\mu) & \lambda & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

Za sisteme masovnog opsluživanja ćemo izračunavati ergodične verovatnoće i posmatrati stanje sistema posle dovoljno dugog vremena. Računamo

$$s_0 = 1 + \frac{\lambda}{\mu} + \frac{\lambda^2}{\mu 2\mu} + \cdots \frac{\lambda^r}{r!\mu^r} + \frac{\lambda^{r+1}}{rr!\mu^{r+1}} + \frac{\lambda^{r+2}}{r^2 r!\mu^{r+2}} + \cdots$$

Ova suma konvergira ako je $\lambda < r\mu$, u tom slučaju je proces ergodičan, inače se proces zagušuje i kako $t \to \infty$, broj klijenata u sistemu teži beskonačnosti.

Imamo $p_0 = \frac{1}{s_0}$, a ostale verovatnoće dobijamo:

$p_k^* = \frac{\lambda}{k\mu} p_{k-1}^*$, $k = 1, 2, \ldots r$; $\quad p_k^* = \frac{\lambda}{r\mu} p_{k-1}^*$, $k = r+1, r+2, \ldots$.

Ako je maksimalna dužina reda konačna, onda je skup mogućih stanja konačan. Onda klijenti koji dođu u sistem kad su sva mesta na priborima i u redu za čekanje zauzeta, dobijaju otkaz. U tom slučaju matrica Λ ne ide do beskonačnosti, već su $\lambda_k = \lambda_{k+1} = \cdots = 0$ i $\mu_{k+1} = \mu_{k+2} = \cdots = 0$. Tada suma s_0 ima konačno mnogo nenula sabiraka, odnosno, konvergira.

Definišemo **propusnu moć sistema** kao verovatnoću da klijent može ući u sistem: $1 - p_{r+k}^*$.

Smatramo da posle dugog vremenskog perioda t dolazi do stabilizacije sistema i da sistem dobija raspodelu

$$X(\infty) : \begin{pmatrix} 0 & 1 & \cdots \\ p_0^* & p_1^* & \cdots \end{pmatrix}.$$

Očekivani broj klijenata u sistemu, L, se računa posle dugog vremenskog perioda kao očekivanje slučajne promenljive $X(\infty)$:

$$L = E(X(\infty)) = \sum_{k=1}^{\infty} k p_k^*.$$

Može se naći i očekivani broj klijenata u redu za čekanje kao:

$$L_q = \sum_{k=1}^{\infty} k p_{r+k}^*.$$

Ako se sa $\bar{\lambda}$ označi efektivna brzina ulazaka klijenata u sistem, onda

$$\bar{\lambda} = \begin{cases} k = \infty : & \lambda \\ k < \infty : & \sum_{k=0}^{\infty} k\lambda_k = (1 - p_{r+k}^*)\lambda \end{cases}$$

Littleove formule daju W, očekivano vreme koje klijent provede u sistemu; W_q, očekivano vreme koje klijent provede čekajući, koristeći $\bar{\lambda}$ i L:

$$W = \frac{L}{\bar{\lambda}}, \quad W_q = \frac{L_q}{\bar{\lambda}}, \quad W = W_o + W_q ,$$

gde je $W_o = \frac{1}{\mu}$ očekivano vreme opsluživanja.

Zadatak

Banka ima dva šaltera. Svaki šalter opslužuje klijente, nezavisno, sa vremenom opsluživanja koje ima eksponencijalu raspodelu, sa prosečnim trajanjen $1min$.

Korisnici pristižu u banku po Poasonovoj raspodeli (vreme između dva pristizanja ima nezavisnu eksponencijalnu raspodelu), prosečno 100 klijenata na sat.

Izračunati

a) verovatnoću da u banci ima više od tri klijenta.

b) verovatnoću da je prvi šalter slobodan.

c) prosečan broj klijenata u banci.

d) prosečan broj klijenata u redu za čekanje.

e) prosečan broj zauzetih šaltera.

f) prosečno vreme koje klijent provede u banci.

g) prosečno vreme koje klijent provede u redu.

Rešenje

Ovo je sistem masovnog opsluživanja M|M|2 sa $\lambda = 100h^{-1}$ i $\frac{1}{\mu} = 1min = \frac{1}{60}h$, odnosno, $\mu = 60h^{-1}$.

Sistem je ergodičan jer je $\lambda = 100 < 2\mu = 120$.

Sav račun obavljamo za stacionarno stanje. Izračunajmo ergodične verovatnoće.

$$\begin{aligned} s_0 &= 1 + \frac{\lambda}{\mu} + \frac{\lambda^2}{2\mu^2} + \frac{\lambda^3}{2^2\mu^3} + \frac{\lambda^4}{2^3\mu^4} + \cdots \\ &= 1 + \frac{\lambda}{\mu}\left[1 + \frac{\lambda}{2\mu} + \left(\frac{\lambda}{2\mu}\right)^2 + \left(\frac{\lambda}{2\mu}\right)^3 + \cdots\right] \\ &= 1 + \frac{\lambda}{\mu}\frac{1}{1-\frac{\lambda}{2\mu}} = 11 \end{aligned}$$

Ergodične verovatnoće su:

$$p_0^* = \frac{1}{11}\,,\, p_1^* = \frac{\lambda}{\mu}p_0^* = \frac{5}{33}\,,\, p_2^* = \frac{\lambda}{2\mu}p_1^* = \frac{25}{198}\,,\, p_3^* = \frac{\lambda}{2\mu}p_2^* = \frac{125}{1188}\,, \dots$$

Odnosno, slučajna promenljiva $X(\infty)$ ima raspodelu

$$X(\infty): \begin{pmatrix} 0 & 1 & 2 & 3 & \cdots \\ \frac{1}{11} & \frac{5}{33} & \frac{25}{198} & \frac{125}{1188} & \cdots \end{pmatrix}$$

Opšta formula je

$$p_0^* = \frac{1}{11}\,,\, p_1^* = \frac{5}{33}\,,\, p_k^* = p_1^*\left(\frac{\lambda}{2\mu}\right)^{k-1} = \frac{5}{33}\left(\frac{5}{6}\right)^{k-1}\,,\, k = 2,3,\dots$$

a) $p_4^* + p_5^* + \cdots = 1 - (p_0^* + p_1^* + p_2^* + p_3^*) =$
$= 1 - \left(\frac{1}{11} + \frac{5}{33} + \frac{25}{198} + \frac{125}{1188}\right) = \frac{625}{1188} = 0.5261$

b) Prvi šalter je slobodan ako nikog nema u sistemu ili je samo jedan klijent na drugom šalteru:

$$p_0^* + \frac{1}{2}p_1^* = \frac{1}{11} + \frac{5}{66} = \frac{1}{6}$$

Rezultat možemo tumačiti da je na šest sati prvi šalter slobodan oko jedan sat.

c) $L = \sum_{k=1}^{\infty} k p_k^* = \sum_{k=1}^{\infty} k p_1^* \left(\frac{\lambda}{2\mu}\right)^{k-1} = \frac{p_1^*}{(1-\frac{\lambda}{2\mu})^2} = \frac{5/33}{(1-5/6)^2} = \frac{60}{11} = 5.4545$

d) $L_q = \sum_{k=1}^{\infty} k p_{k+2}^* = \sum_{k=1}^{\infty} k p_1^* \left(\frac{\lambda}{2\mu}\right)^{k+1} = \frac{p_1^*\left(\frac{\lambda}{2\mu}\right)^2}{(1-\frac{\lambda}{2\mu})^2} = \left(\frac{\lambda}{2\mu}\right)^2 L = \frac{125}{33} = 3.7879$

e) Prvi način: Posmatramo slučajnu promenljivu $S =$ broj klijenata na šalterima.

Njena raspodela je: $S: \begin{pmatrix} 0 & 1 & 2 \\ p_0^* & p_1^* & p_2^* + p_3^* + \cdots \end{pmatrix}$.

Prosečan broj zauzetih šaltera je jednak očekivanju slučajne promenljive S.

$L_s = E(S) = 0 \cdot p_0^* + 1 \cdot p_1^* + 2\,(p_2^* + p_3^* + p_4^* + \cdots) =$
$= \frac{5}{33} + 2\left(1 - \frac{1}{11} - \frac{5}{33}\right) = \frac{5}{3} = 1.6667$

Drugi način: $L_s + L_q = L \Rightarrow L_s = L - Lq = \frac{60}{11} - \frac{125}{33} = \frac{5}{3} = 1.6667$

f) Maksimalna dužina reda je ∞, zato $\bar{\lambda} = \lambda$.

Little1: $W = \frac{L}{\bar{\lambda}} = \frac{\frac{60}{11}}{100} = \frac{3}{55}h = 3min16.36sec$

g) Prvi način (Little2): $W_q = \dfrac{L_q}{\bar{\lambda}} = \dfrac{\frac{125}{33}}{100} = \dfrac{5}{132} h = 2 min 16.36 sec$

Drugi način, koristeći očekivano vreme opsluživanja:

$$W_o = 1 min = \frac{1}{60} \;\Rightarrow\; \text{(Little3)} : W_o + W_q = W \Rightarrow$$

$$\Rightarrow\; W_q = W - W_o = \frac{3}{55} h - \frac{1}{60} h = \frac{5}{132} h = 2 min 16.36 sec$$