4. Linearno programiranje i simpleksni algoritam

4.1. Uvod

U okviru ovog poglavnja je data definicija problema linearnog programiranja, te su razmotrene osobine skupa dozvoljenih vrijednosti ovakvog problema. Obzirom da su ograničenja i kriterij linearni, priroda problema linearnog programiranja je jako dobro poznata. Pokazano je da se rješenje nalazi na granici skupa dozvoljenih vrijednosti. U okviru poglavlja je predstavljen simpleksni algoritam, kao još uvijek osnovni algoritam za rješavanje problema linearnog programiranja. Razmotrene su različite pojave do kojih može doći, a koje otežavaju rješavanje problema linearnog programiranja, te pokazano kako se one tretiraju. Definiran je dualni problem linearnog programiranja i pokazana njegova veza sa polaznim, primarnim problemom.

4.2. Definicija problema linearnog programiranja

Kako je u prethodnim poglavljima već rečeno, ukoliko su svi elementi modela problemske situacije linearni, radi se o problemu linearnog programiranja. Problemi linearnog programiranja su dobro izučeni i poznate su njihova svojstva u pogledu egzistencije rješenja. Osnovnu poteškoću kod problema linearnog programiranja predstavlja njihova velika dimenzionalnost, obzirom da može postojati više hiljada problemskih varijabli i ograničenja.

4.2.1. Standardna forma problema linearnog programiranja

U nastavku ćemo razmotriti elemente problema linearnog programiranja i definirati njegovu standardnu formu. Ova standardna forma će biti polazna osnova za predstavljanje simpleksnog algoritma i načina tretiranja svih specijalnih slučajeva u okviru problema linearnog programiranja. Treba napomenuti da se u literaturi mogu pronaći i drugačije postavke standardne forme problema linearnog programiranja, ali to ne umanjuje opštost razmatranja iz ovog poglavlja, obzirom da svi koraci algoritma i opisani postupci imaju analognu varijantu za drugačije definiranu standardnu formu.

Problemska situacija se modelira problemom linearnog programiranja (LP) kada su i kriterij i ograničenja koja definiraju skup dozvoljenih vrijednosti problemskih varijabli linearni.

U općem obliku, kriterij problema linearnog programiranja možemo zapisati kao:

$$P = f(x_1, x_2, \ldots, x_n) = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$$

U ovom izrazu x_i predstavljaju n problemskih varijabli, a c_i n koeficijenata kriterija (cijena). Ukoliko problemske varijable i koeficijente kriterija predstavimo vektorima dimenzije $nx1$:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \boldsymbol{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

kriterij se može iskazati izrazom:

$$P = f(\boldsymbol{x}) = \boldsymbol{c}' \boldsymbol{x}$$

Obzirom da je skup dozvoljenih vrijednosti varijabli iskazan linearnim ograničenjima, njihov oblik će u općem slučaju biti:

$$\begin{array}{l} a_{11} x_1 + a_{12} x_2 + \ldots + a_{1j} x_j + \ldots + a_{1n} x_n \leqslant b_1 \\ a_{21} x_1 + a_{22} x_2 + \ldots + a_{2j} x_j + \ldots + a_{2n} x_n \leqslant b_2 \\ \vdots \qquad \vdots \qquad \vdots \qquad \vdots \quad \vdots \\ a_{i1} x_1 + a_{i2} x_2 + \ldots + a_{ij} x_j + \ldots + a_{in} x_n \leqslant b_i \\ \vdots \qquad \vdots \qquad \vdots \qquad \vdots \quad \vdots \\ a_{m1} x_1 + a_{m2} x_2 + \ldots + a_{mj} x_j + \ldots + a_{mn} x_n \leqslant b_m \end{array}$$

Ovaj sistem nejednakosti definira skup od m ograničenja koja se nazivaju balansnim uslovima. Osim balansnih uslova, u standardnoj formi problema linearnog programiranja su vrijednosti svih varijabli veće ili jednake nuli, što je definirano skupom prirodnih ograničenja:

$$x_1 \geqslant 0, \;\; x_2 \geqslant 0, \;\; \ldots, \;\; x_j \geqslant 0, \;\; \ldots, \;\; x_n \geqslant 0$$

Slobodne članove b_i iz balansnih uslova predstavimo vektorom $\boldsymbol{b}$ dimenzija $mx1$, te koeficijente a_{ij} matricom $\boldsymbol{A}$ dimenzija mxn :

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1j} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2j} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & & & \vdots \\ a_{i1} & a_{i2} & & a_{ij} & & a_{in} \\ \vdots & & & & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mj} & \ldots & a_{mn} \end{bmatrix} \qquad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Sada se ograničenja mogu zapisati kao:

$$\boldsymbol{A}\boldsymbol{x} \leqslant \boldsymbol{b}$$
$$\boldsymbol{x} \geqslant 0$$

Matrica $\boldsymbol{A}$ se naziva tehnološkom matricom, a vektor $\boldsymbol{b}$ vektorom slobodnih članova iz balansnih uslova ili vektorom resursa.

Nakon što smo definirali sve elemente problema linearnog programiranja, za standardnu formu ovog problema ćemo usvojiti:

$$P^* = \max_{\substack{\boldsymbol{A}\boldsymbol{x} \leqslant \boldsymbol{b} \\ \boldsymbol{x} \geqslant 0}} \boldsymbol{c}'\boldsymbol{x} \qquad \boldsymbol{x}^* = \operatorname{argmax}_{\substack{\boldsymbol{A}\boldsymbol{x} \leqslant \boldsymbol{b} \\ \boldsymbol{x} \geqslant 0}} \boldsymbol{c}'\boldsymbol{x}$$

Očigledno smo za standardnu formu problema usvojili problem traženja tačke maksimuma.

4.2.2. Svođenje problema linearnog programiranja na standardnu formu

Obzirom da ćemo algoritam za rješavanje problema linearnog programiranja predstaviti u formi za rješavanje standardnog problema linearnog programiranja, potrebno je prije rješavanja problem svesti u tu formu. U nastavku će biti predstavljen način svođenja nekih od tipičnih formi koje modeliraju linearne probleme u standardnu formu.

a) Problem traženja minimuma kriterija

Kako je već prije rečeno, između traženja minimuma i traženja maksimuma postoji veza data izrazom:

$$P_1(\boldsymbol{x}) = -P(\boldsymbol{x}) = -\boldsymbol{c}'\boldsymbol{x}$$
$$\min P(\boldsymbol{x}) = -\max P_1(\boldsymbol{x})$$

Pri tome rješenje problema ostaje isto, bez obzira da li se radi o traženju maksimuma ili minimuma kriterija.

b) U problemu postoje ograničenja oblika $d_j \leq x_j$

Ovakva ograničenja se mogu smatrati specijalnim slučajem balansnih uslova i na standardnu formu ih je moguće svesti množenjem sa -1. Međutim, da bi se umanjio broja ograničenja balansnih uslova, u slučaju postojanja ovakvih ograničenja se može uvesti smjena varijabli. Naime, pošto vrijedi:

$$0 \leqslant x_j - d_j$$

varijabla x_j se u problemu može smjeniti varijablom $x_{\alpha j}$ prema izrazu:

$$x_{\alpha j} = x_j - d_j$$

Sada za varijablu $x_{\alpha j}$ vrijedi standardno prirodno ograničenje $x_{\alpha j} \geqslant 0$.

Dakle, nakon što se u kriterij i sva ograničenja uvedu varijable $x_{\alpha j}$, problem se rješava. Nakon određivanja rješenja problema je potrebno odrediti rješenje polaznog problema tako što se uklone varijable $x_{\alpha j}$ u skladu sa izrazom:

$$x_j^* = x_{\alpha j}^* + d_j$$

c) U problemu postoje ograničenja oblika $d_j \geq x_j$

I ovakva ograničenja predstavljaju specijalni slučaj balansnih uslova koja se mogu ukloniti smjenom varijabli slično kao u prethodnom slučaju:

$$\begin{gathered} d_j - x_j \geqslant 0 \\ x_{\beta j} = d_j - x_j \\ x_{\beta j} \geqslant 0 \end{gathered}$$

Nakon što se riješi problem linearnog programiranja sa uvedenim varijablama $x_{\beta j}$, potrebno je odrediti rješenje polaznog problema u skladu sa izrazom:

$$x_j^* = d_j - x_{\beta j}^*$$

d) U problemu postoje ograničenja oblika $d_{j1} \leq x_j \leq d_{j2}$

Ovakva ograničenja se, kao i u prethodna dva slučaja, mogu eliminirati uvođenjem varijable $x_{\alpha j}$, s tim da je sada potrebno dodati i jedan balansni uslov po uvedenoj varijabli, prema izrazima:

$$\begin{aligned} &0 \leqslant x_j - d_{j1} \leqslant d_{j2} - d_{j1} \\ &x_{\alpha j} = x_j - d_{j1}\ , \quad b_{\alpha j} = d_{j2} - d_{j1} \\ &x_{\alpha j} \geqslant 0\ , \quad x_{\alpha j} \leqslant b_{\alpha j} \end{aligned}$$

Uvođenjem varijable $x_{\alpha j}$ je eliminiran jedan balansni uslov, a slobodni član iz drugog balansnog uslova je modificiran. Jasno, i ovdje je potrebno nakon rješavanja odrediti rješenje polaznog problema u skladu sa izrazom:

$$x_j^* = x_{\alpha j}^* + d_{j1}$$

e) U problemu postoji varijabla x_j slobodna po znaku

Kako je definirano u standardnoj formi problema linearnog programiranja, sve problemske varijable su nenegativne. Ukoliko linearni model problema sadrži jednu ili više varijabli za koje ne postoji prirodno ograničenje (odnosno koje su slobodne po znaku), tada se problem na standardnu formu može svesti tako što će se svaka takva varijabla razložiti na dvije varijable za koje važe prirodna ograničenja:

$$\begin{gathered} x_j = x_{j+} - x_{j-} \\ x_{j+} \geqslant 0 \\ x_{j-} \geqslant 0 \end{gathered}$$

Smjenu varijable x_j varijablama x_{j+} i x_{j-} treba izvršiti kako u svim ograničenjima tako i u kriteriju. Nakon rješavanja problema sa uvedenim varijablama x_{j+} i x_{j-} se rješenje polaznog problema određuje kao:

$$x_j^* = x_{j+}^* - x_{j-}^*$$

f) U problemu postoje ograničenja oblika $|a_{i1}x_1+a_{i2}x_2+\ldots+a_{in}x_n|\leqslant b_i$

U slučaju postojanja ovakvog ograničenja, ono se u standardnoj formi problema linearnog programiranja može zamijeniti sa dva ograničenja tipa nejednakosti:

$$a_{i1}x_1+a_{i2}x_2+\ldots+a_{in}x_n\geqslant -b_i$$
$$a_{i1}x_1+a_{i2}x_2+\ldots+a_{in}x_n\leqslant b_i$$

Rješenje ovako modificiranog problema je ujedno rješenje polaznog problema.

g) U problemu postoje ograničenja oblika $|a_{i1}x_1+a_{i2}x_2+\ldots+a_{in}x_n|\geqslant b_i$

Ovo ograničenje se može predstaviti ograničenjima tipa nejednakosti:

$$a_{i1}x_1+a_{i2}x_2+\ldots+a_{in}x_n\geqslant -b_i$$

ili

$$a_{i1}x_1+a_{i2}x_2+\ldots+a_{in}x_n\leqslant b_i$$

Postupak rješavanja problema sa ovakvim ograničenjem se sastoji u rješavanju dva podproblema, pri čemu u definiranju skupa dozvoljenih vrijednosti svakog od njih učestvuje po jedno od gore navedenih ograničenja tipa nejednakosti.

Nakon rješavanja ova dva problema, za rješenje polaznog problema se uzima ono koje je bolje u smislu definicije optimuma. Drugim riječima, za rješenje polaznog problema se uzima rješenje onog podbroblema koje daje veću vrijednost kriterija u slučaju traženja maksimuma. U slučaju traženja minimuma se uzima rješenje onog podproblema koje daje manju vrijednost kriterija.

h) Traži se minimum kriterija definiranog kao $P(\boldsymbol{x})=|c_1x_1+c_2x_2+\ldots+c_nx_n|$

Kao i u prethodnom slučaju, rješavaju se dva podproblema. Nakon njihovog rješavanja, za rješenje polaznog problema se uzima ono koje daje manju vrijednost kriterija.

Za prvi podproblem se uzima kriterij $P_1(\boldsymbol{x})=-\boldsymbol{c}'\boldsymbol{x}$. Uz to, skupu balansnih uslova se dodaje ograničenje:

$$c_1x_1+c_2x_2+\ldots+c_nx_n\geqslant 0$$

Drugim riječima, traži se maksimum kriterija $P_1(\boldsymbol{x})$ kada je njegova vrijednost nenegativna.

Za drugi podproblem se uzima kriterij $P_2(\boldsymbol{x})=\boldsymbol{c}'\boldsymbol{x}$, uz dodatnu nejednakost koja ograničava traženje maksimum na slučaj kada je kriterij $P_2(\boldsymbol{x})$ nepozitivan:

$$c_1x_1+c_2x_2+\ldots+c_nx_n\leqslant 0$$

4.3. Rješenje problema linearnog programiranja

U prethodnom odjeljku smo definirali polaznu standardnu formu problema linearnog programiranja i vidjeli načine na koje se neki od češćih modela problema linearnog programiranja mogu svesti na ovu standardnu formu.

Sada ćemo razmotriti osobine rješenja problema linearnog programiranja, te utvrditi gdje se to rješenje nalazi u slučaju da egzistira. Za ova razmatranja je od značaja poznavanje osobina skupa dozvoljenih vrijednosti.

4.3.1. Vršne i kvazivršne tačke skupa dozvoljenih vrijednosti

Potencijalno rješenje problema linearnog programiranja predstavlja tačka prostora E^n koja zadovoljava sva ograničenja tipa nejednakosti.

Kako je već definirano, skup dozvoljenih vrijednosti je definiran sa m ograničenja balansnih uslova i n prirodnih ograničenja.

Obzirom da je tačka u prostoru E^n definirana presjekom n hiperravni, ukupan broj tačaka koje možemo odrediti tako što će od svih $m+n$ ograničenja po n njih biti granično zadovoljeno je dat izrazom:

$$q=\frac{(n+m)!}{n!m!}$$

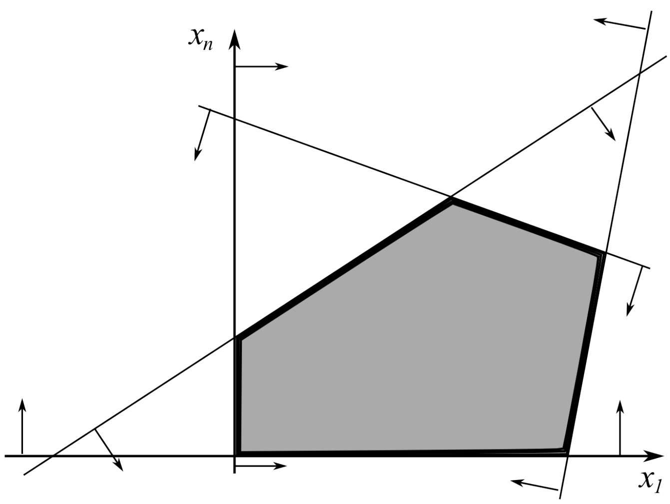
Definicija:

Svaka od tačaka koja granično zadovoljava n ograničenja, a zadovoljava i preostalih m ograničenja se naziva ***vršnom tačkom*** *skupa dozvoljenih vrijednosti.*

Definicija:

Svaka od tačaka koja granično zadovoljava n ograničenja, a ne zadovoljava barem jedno od preostalih m ograničenja se naziva ***kvazivršnom tačkom*** *skupa dozvoljenih vrijednosti.*

Vršne i kvazivršne tačke skupa dozvoljenih vrijednosti se mogu vidjeti na slici 4.1.



Slika 4.1: Vršne i kvazivršne tačke skupa dozvoljenih vrijednosti problema linearnog programiranja

Vršne i kvazivršne tačke su od velikog značaja za rješavanje problema linearnog programiranja. Naime, ukoliko postoji rješenje problema LP, ono se nalazi ili u vršnoj tački skupa dozvoljenih vrijednosti ili u linearnoj kombinaciji vršnih tačaka. Kvazivršne tačke su od značaja za tzv. dualni problem linearnog programiranja, o čemu će naknadno biti riječi.

Dakle, da bi se odredilo rješenje problema linearnog programiranja, dovoljno je pretražiti sve vršne tačke skupa dozvoljenih vrijednosti. Međutim, određivanje vršnih tačaka nije jednostavno pogotovo ako se u obzir uzme velika dimenzionalnost problema linearnog programiranja. Uz to, da bismo odredili sve vršne tačke skupa, potrebno je od svih $n+m$ ograničenja tipa nejednakosti sistematično postaviti po n jednačina (granično zadovoljenih ograničenja), riješiti ih i svaki put provjeriti preostalih m ograničenja da bi se utvrdilo radi li se o vršnoj ili kvazivršnoj tački.

4.3.2. Prošireni problem linearnog programiranja

Za određivanje vršnih tačaka je pogodinije kada se ograničenja tipa nejednakosti svedu na jednakosti. Radi toga se u svaki od balansnih uslova uvodi po jedna slack varijabla.

Za ograničenje:

$$a_{i1}x_1+a_{i2}x_2+\ldots+a_{ij}x_j+\ldots+a_{in}x_n\leqslant b_i$$

se uvodi slack varijabla sa koeficijentom +1, tako da će ograničenje tipa jednakosti biti:

$$a_{i1}x_1+a_{i2}x_2+\ldots+a_{ij}x_j+\ldots+a_{in}x_n+x_{n+i}=b_i$$

Slično, za ograničenje:

$$a_{i1}x_1+a_{i2}x_2+\ldots+a_{ij}x_j+\ldots+a_{in}x_n\geqslant b_i$$

se uvodi slack varijabla sa koeficijentom -1, tako da će ograničenje tipa jednakosti biti:

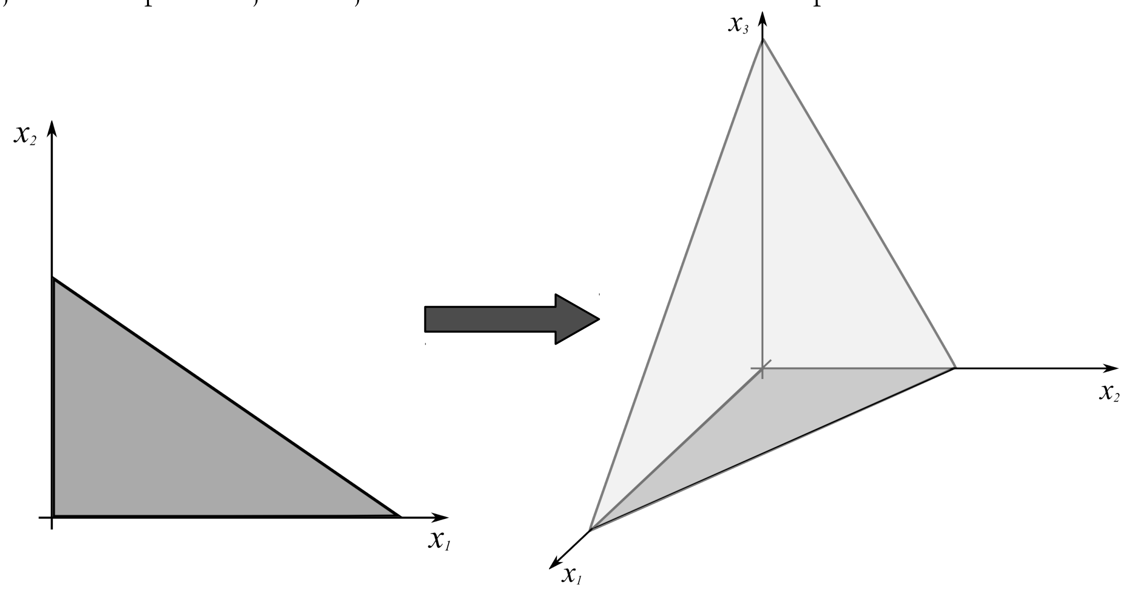
$$a_{i1}x_1+a_{i2}x_2+\ldots+a_{ij}x_j+\ldots+a_{in}x_n-x_{n+i}=b_i$$

Pri tome sve slack varijable moraju biti nenegativne, tj. mora vrijediti nejednakost:

$$x_{n+i}\geqslant 0$$

Ovo ograničenje ima formu prirodnog ograničenja, tako da se nakon uvođenja slack varijabla može tretirati kao i bilo koja druga problemska varijabla.

Grafička interpretacija svođenja ograničenja tipa nejednakosti na jednakosti je data na slici 4.2. Jasno se vidi da je ovim skup dozvoljenih vrijednosti vezan za n-dimenzionalnu hiperravan u E^{n+m}.



Slika 4.2: Grafička interpretacija efekta svođenja ograničenja tipa nejednakosti na jednakosti

Problem linearnog programiranja u koji smo uveli slack varijable nazivamo proširenim problemom linearnog programiranja. Ukoliko vektoru problemskih varijabli $\boldsymbol{x}$ dodamo slack varijable, dobićemo vektor varijabli proširenog problema linearnog programiranja:

$$\tilde{\boldsymbol{x}}=\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ x_{n+1} \\ \vdots \\ x_{n+m} \end{bmatrix}$$

Obzirom da slack varijable ne smiju utjecati na rješenje proširenog problema, njihovi koeficijenti kriterija će biti 0. Sada će vektor koeficijenata proširenog problema linearnog programiranja dimenzija $(n+m)\times 1$ biti:

$$\tilde{\boldsymbol{c}}=\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Sada se kriterij proširenog problema LP može predstaviti kao:

$$\tilde{P}=\tilde{\boldsymbol{c}}\,'\,\tilde{\boldsymbol{x}}$$

Tehnološka matrica proširenog problema će biti dimenzija $m\times(n+m)$:

$$\tilde{\boldsymbol{A}}=\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & a_{1\,n+1} & \dots & a_{1\,n+m} \\ a_{21} & a_{22} & \dots & a_{2n} & a_{2\,n+1} & \dots & a_{2\,n+m} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & a_{m\,n+1} & \dots & a_{m\,n+m} \end{bmatrix}$$

Pri tome nedijagonalni elementi podmatrice dimenzija $m\times m$ koja je dodana tehnološkoj matrici $\boldsymbol{A}$ imaju vrijednost 0, dok elementi glavne dijagonale mogu imati vrijednost +1 ili -1, ovisno o obliku nejednakosti u koju je slack varijabla uvedena.

Vektor slobodnih članova balansnih uslova $\boldsymbol{b}$ za prošireni problem se ne razlikuje u odnosu na ovaj vektor za standardnu formu ograničenja u obliku nejednakosti.

Sada se skup dozvoljenih vrijednosti za prošireni problem linearnog programiranja može predstaviti kao:

$$\tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}=\boldsymbol{b}$$
$$\tilde{\boldsymbol{x}}\geqslant 0$$

Rješenje ovako definiranog problema je:

$$P^*=\tilde{P}^*=\max_{\substack{\tilde{A}\tilde{x}=b\\ \tilde{x}\geqslant 0}}\tilde{\boldsymbol{c}}'\tilde{\boldsymbol{x}} \qquad \tilde{\boldsymbol{x}}^*=\operatorname{argmax}_{\substack{\tilde{A}\tilde{x}=b\\ \tilde{x}\geqslant 0}}\tilde{\boldsymbol{c}}'\tilde{\boldsymbol{x}}$$

4.3.3. Osobine skupa dozvoljenih vrijednosti

Razmotrimo sada formalnije koje sve tačke skupa dozvoljenih vrijednosti mogu predstavljati rješenje problema linearnog programiranja.

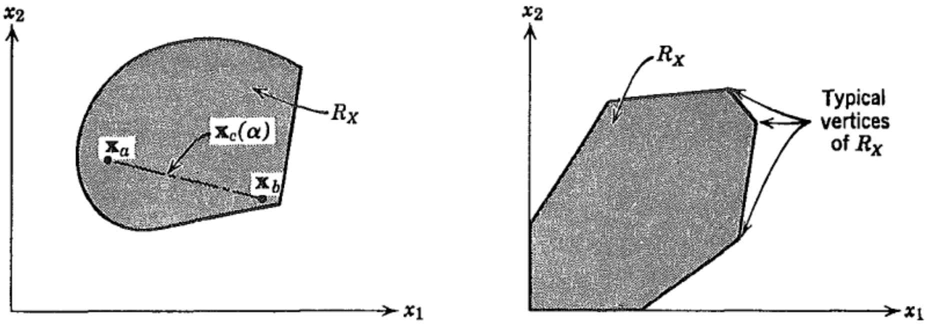
Definicija:

Za skup Ω kažemo da je konveksan ukoliko za svaku tačku $\boldsymbol{x}_c$ duži koja spaja tačke $\boldsymbol{x}_a\in\Omega$ i $\boldsymbol{x}_b\in\Omega$ tj.

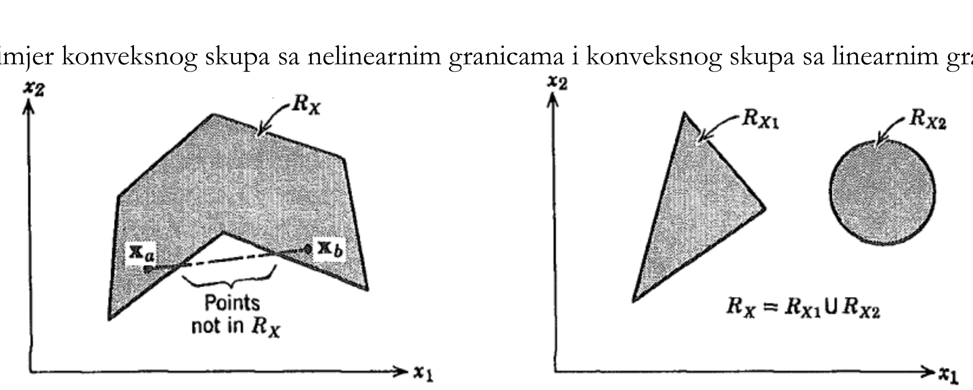
$$\boldsymbol{x}^c=\alpha\,\boldsymbol{x}^a+(1-\alpha)\,\boldsymbol{x}^b,\quad 0\leqslant\alpha\leqslant 1$$

vrijedi $\boldsymbol{x}_c\in\Omega$.

Primjeri konveksnih skupova su predstavljeni na slici 4.3, a primjeri nekonveksnih skupova na slici 4.4.



Slika 4.3: Primjer konveksnog skupa sa nelinearnim granicama i konveksnog skupa sa linearnim granicama



Slika 4.4: Primjer nekonveksnog skupa sa linearnim granicama i nekonveksnog skupa koji se sastoji od dvije nepovezane oblasti

Za skup dozvoljenih vrijednosti problema linearnog programiranja vrijedi sljedeća

Tvrdnja:

Skup dozvoljenih vrijednosti za probleme linearnog programiranja je konveksan.

Dokaz:

Da bismo dokazali ovu tvrdnju, poći ćemo od definicije konveksnosti, te ćemo uzeti proizvoljnu tačku neke duži definirane tačkama iz skupa dozvoljenih vrijednosti, tj.

$$\boldsymbol{x}^c = \alpha\, \boldsymbol{x}^a + (1-\alpha)\, \boldsymbol{x}^b, \quad 0 \leqslant \alpha \leqslant 1$$

Na osnovu polazne pretpostavke da $\mathbf{x}_a \in \Omega$ *i* $\mathbf{x}_b \in \Omega$*, možemo pisati:*

$$\begin{aligned}
&\boldsymbol{x}^a \in \Omega \Rightarrow \boldsymbol{A}\,\boldsymbol{x}^a = \boldsymbol{b}, \quad \boldsymbol{x}^a \geqslant 0\\
&\boldsymbol{x}^b \in \Omega \Rightarrow \boldsymbol{A}\,\boldsymbol{x}^b = \boldsymbol{b}, \quad \boldsymbol{x}^b \geqslant 0\\
&\boldsymbol{A}\,\boldsymbol{x}^c = \boldsymbol{A}[\alpha\,\boldsymbol{x}^a + (1-\alpha)\,\boldsymbol{x}^b] = \alpha\,\boldsymbol{A}\,\boldsymbol{x}^a + (1-\alpha)\,\boldsymbol{A}\,\boldsymbol{x}^b = \alpha\,\boldsymbol{b} + (1-\alpha)\,\boldsymbol{b} = \boldsymbol{b}
\end{aligned}$$

Obzirom da je $0 \leq \alpha \leq 1$*, vrijedi da je:*

$$\alpha\,\boldsymbol{x}^a \geqslant 0 \wedge (1-\alpha)\,\boldsymbol{x}^b \geqslant 0 \Rightarrow \boldsymbol{x}^c \geqslant 0$$

Kako je pokazano da vrijedi:

$$\boldsymbol{A}\,\boldsymbol{x}^c = \boldsymbol{b} \wedge \boldsymbol{x}^c \geqslant 0 \Rightarrow \boldsymbol{x}^c \in \Omega$$

Ovim je tvrdnja dokazana.

Ako vrijednost kriterija posmatramo kao dodatnu varijablu (dimenziju) prostora, tj. $P = x_{n+m+1}$, tada imamo još jednu jednačinu hiperravni u prostoru E^{n+m+1}. Kako svakoj tački skupa $\boldsymbol{x} \in \Omega$ odgovara neka vrijednost kriterija, odnosno tačka na hiperravni $P(\mathbf{x}) = x_{n+m+1}$, ovim se cijeli skup Ω iz prostora E^{n+m} preslikava na skup Ω^P u prostoru E^{n+m+1}. Kao i za skup Ω, može se pokazati da je ovaj skup konveksan. Radi toga postoje jedinstveni (globalni) minimum i maksimum problema linearnog programiranja, što olakšava njegovo rješavanje iterativnim postupcima.

Definicija:

Tačka $\boldsymbol{x}^v$ *predstavlja ekstremalnu tačku skupa* Ω *ukoliko bilo koja tačka* $\boldsymbol{x}^c$ *definirana kao:*

$$\boldsymbol{x}^c = \beta\,\boldsymbol{x}^v + (1-\beta)\,\boldsymbol{x}^a, \quad \beta \geqslant 1, \quad \boldsymbol{x}^a \in \Omega, \quad \boldsymbol{x}^a \neq \boldsymbol{x}^v$$

ne pripada skupu Ω.

Kada je skup Ω mnogostraničnik, njegove ekstremalne tačke su u stvari vršne tačke mnogostraničnika.

Teorema:

Vršna tačka skupa Ω ima najviše m nenultih elemenata.

Dokaz:

Pretpostavimo da je $\boldsymbol{x}^q$ vršna tačka skupa Ω koja ima više od m nenultih elemenata. Uzmimo neku tačku $\boldsymbol{x}^a \in \Omega$ koja ima nule na istim mjestima kao i tačka $\boldsymbol{x}^q$, ali koja nije ujedno i vršna tačka. Posmatrajmo sada tačku:

$$\boldsymbol{x}^c = \beta \boldsymbol{x}^q + (1-\beta)\boldsymbol{x}^a, \quad \beta \geqslant 1$$

Za neko dovoljno malo β svi elementi ove tačke na mjestima nenultih elemenata $\boldsymbol{x}^q$ će biti nenegativni. Isto tako, pokazuje se da vrijedi:

$$\boldsymbol{A}\boldsymbol{x}^c = \boldsymbol{b}$$

što znači da se radi o tački skupa Ω. Međutim, to ne može biti slučaj ako je $\boldsymbol{x}^q$ vršna tačka.

S druge strane, ako je od ukupno $n+m$ varijabli njih n jednako nuli, tada se preostalih m nenultih može jedinstveno odrediti na osnovu sistema od m ograničenja tipa jednakosti. Radi toga ne može postojati ni jedna druga tačka $\boldsymbol{x}^a \in \Omega$ koja ima nenulte vrijednosti na istim mjestima kao i $\boldsymbol{x}^q$, odnosno bilo koja tačka $\boldsymbol{x}^c$ ne može pripadati skupu Ω jer će barem jedna komponenta ove tačke biti negativna.

U nastavku ćemo bez dokaza navesti teoremu, koja je osnova za gradnju iterativnog postupka za pretraživanje skupa dozvoljenih vrijednosti kada se radi o problemu linearnog programiranja.

Teorema:

Pod pretpostavkom da postoji konačan ekstremum funkcije P, taj ekstremum se dostiže u barem jednoj vršnoj tački konveksnog skupa Ω. Ukoliko se ekstremum dostiže u više od jedne vršne tačke, tada isto vrijedi i za svaku linearnu konveksnu kombinaciju takvih vršnih tačaka, odnosno za cijelu granicu skupa Ω.

Ova teorema je ključna prije svega iz razloga što pokazuje da je pri rješavanju problema linearnog programiranja pretraživanje skupa dozvoljenih vrijenosti dovoljno ograničiti na pretraživanje njegovih vršnih tačaka, pošto se rješenje ako postoji nalazi ili u vršnoj tački ili na cijeloj granici skupa Ω. Uz to, ukoliko rješenje problema linearnog programiranja nije jedinstveno, onda postoji beskonačno mnogo istovrijednih rješenja.

Iako se ovim bitno smanjuje veličinu prostora koji se pretražuje, broj vršnih tačaka je obično vrlo veliki, a određivanje svih vršnih tačaka nije jednostavno. Radi toga je za rješavanje problema linearnog programiranja potrebno definirati algoritam, koji će polazeći od neke početne vršne tačke prelaziti od jedne do druge vršne tačke, poboljšavajući pri tome vrijednost kriterija koliko je god moguće. Upravo ovo čini simpleksni algoritam.

4.3.4. Simpleksni algoritam

Simpleksni algoritam je razvio George Dantzig 1947. godine, a objavio ga je 1951. godine. Simpleksni algoritam se često svrstava u 10 najvažnijih algoritama razvijenih u XX stoljeću, što postaje jasno ako se u obzir uzme značaj linearnog programiranja kao modela velikog broja praktičnih problema koji se susreću u različitim oblastima. Isto tako, simpleksni algoritam je još i danas najbolji algoritam za rješavanje problema linearnog programiranja u općem slučaju. Iako postoje brojne modifikacije osnovnog algoritma, te razne druge metode za rješavanje ovakvih problema, one su efikasnije samo za specijalne klase problema linearnog programiranja, dok u je drugim slučajevima njihova efikasnost manja.

Naziv algoritma dolazi definicije simpleksa, kao geometrijskog skupa tačaka i činjenice da skup dozvoljenih rješenja problema linearnog programiranja predstavlja simpleks.

Definicija:

k-dimenzionalni simpleks je skup koji se sastoji od $k+1$ tačaka u prostoru dimenzionalnosti veće od k, te svih tačaka koje se mogu odrediti kao:

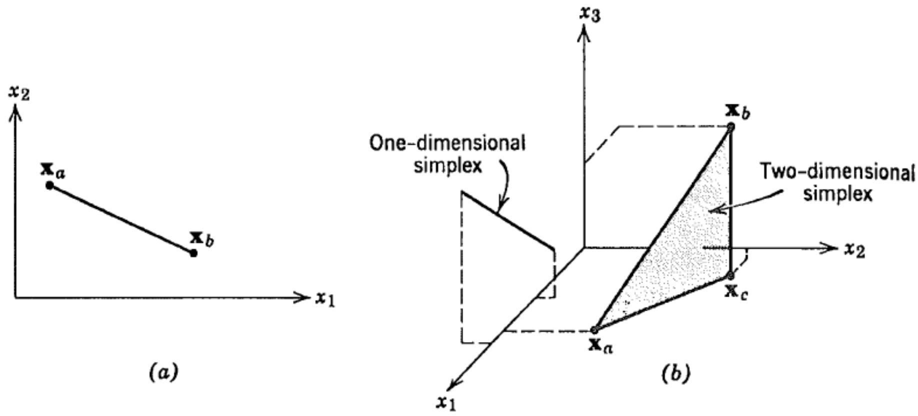
$$\boldsymbol{x}=\alpha_1\boldsymbol{x}_1+\alpha_2\boldsymbol{x}_2+\ldots+\alpha_{k+1}\boldsymbol{x}_{k+1}$$
$$\sum_{j=1}^{k+1}\alpha_j=1$$

Linearno nezavisne tačke predstavljaju vršne tačke simpleksa. Može se lako pokazati da je simpleks konveksan skup. Na slici 4.5 a) je predstavljen primjer jednodimenzionalnog simpleksa u prostoru E^2, a na slici 4.5 b) su predstavljeni primjeri jednodimenzionalnog i dvodimenzionalnog simpleksa u prostoru E^3.

Simpleksni algoritam predstavlja iterativni postupak, koji polazeći od odabrane početne vršne tačke pretražuje dio postojećih vršnih tačaka na skupu dozvoljenih vrijednosti. Pri tome simpleksni algoritam omogućava određivanje rebra, odnosno granice skupa po kojoj kriterij najbrže raste (ili opada), te određivanje naredne vršne tačke. Pretraživanje vršnih tačaka se zaustavlja onda kada u tekućoj vršnoj tački ne postoji ni jedno rebro simpleksa koje dovodi do poboljšanja vrijednosti kriterija (slika 4.6).

Određivanje naredne vršne tačke je omogućeno činjenicom da svaka vršna tačka, kako je to već rečeno, ima najviše m nenultih koordinata. Zahvaljujući tome, vektor $\tilde{\boldsymbol{x}}$ u svakoj vršnoj tački možemo razložiti na tzv. bazni i nebazni dio:

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x}^D \\ \boldsymbol{x}^B \end{bmatrix}$$

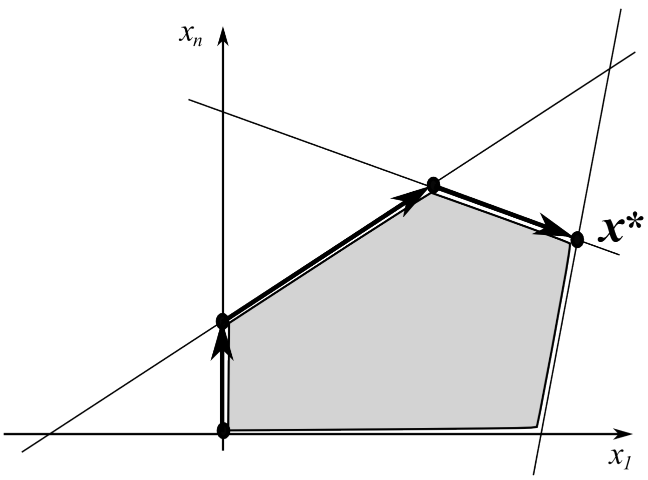


Slika 4.5: Primjeri simpleksa: a) jednodimenzionalni simpleks u E^2

b) jednodimenzionalni i dvodimenzionalni simpleks u E^3

Bazni dio vektora $\boldsymbol{x}^B$ ima dimenzije $m \times 1$ i čine ga nenulte komponente vektora $\tilde{\boldsymbol{x}}$, dok nebazni dio vektora $\boldsymbol{x}^D$ ima dimenzije $n \times 1$ i svi njegovi elementi imaju vrijednost 0. U skladu sa ovom dekompozicijom vektora $\tilde{\boldsymbol{x}}$ se može razložiti i tehnološka matrica proširenog problema linearnog programiranja na svoj bazni i nebazni dio $\tilde{\boldsymbol{A}} = [\boldsymbol{D}\ \boldsymbol{B}]$, pri čemu je:

$$\tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} = [\boldsymbol{D}\ \boldsymbol{B}] \begin{bmatrix} \boldsymbol{x}^D \\ \boldsymbol{x}^B \end{bmatrix} = \boldsymbol{b}$$



Slika 4.6: Iterativno pretraživanje vršnih tačaka

Pošto zbog $\boldsymbol{x}^D = \boldsymbol{0}$ u vršnoj tački uvijek vrijedi da je $\boldsymbol{D}\boldsymbol{x}^D = \boldsymbol{0}$, preostaje da u vršnoj tački uvijek vrijedi i jednakost $\boldsymbol{B}\boldsymbol{x}^B = \boldsymbol{b}$. Ovdje se radi o sistemu od m jednačina sa m nepoznatih koji omogućava da se na osnovu poznavanja baznog dijela tehnološke matrice odrede nenulti elementi vektora $\tilde{\boldsymbol{x}}$ u vršnoj tački, jer je $\boldsymbol{x}^B = \boldsymbol{B}^{-1}\boldsymbol{b}$. Ono što simpleksni algoritam omogućava je da se na sistematičan način jedna bazna matrica prevede u drugu i time ostvari prelazak iz jedne u drugu vršnu tačku.

Prije nego što vidimo na koji način se ostvaruje ovaj prelazak, razmotrimo pitanje određivanja početne vršne tačke. Ovu tačku je najlakše odrediti ako se pođe od definicije tehnološke matrice proširenog problema linearnog programiranja. Naime, ovu matricu formiramo pri svođenju sistema ograničenja tipa nejednakosti iskazanih u standardnoj formi u sistem ograničenja tipa nejednakosti uvođenjem slack varijabli sa koeficijentom +1. Drugim riječima, ova matrica se dobije tako što se tehnološkoj matrici problema linearnog programiranja iskazanog u standardnoj formi doda jedinična matrica dimenzija $m \times m$:

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & 1 & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2n} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & 0 & 0 & \dots & 1 \end{bmatrix} = [\boldsymbol{A}\ \boldsymbol{E}]$$

Za nebazne elemente usvojimo n problemskih varijabli:

$$\boldsymbol{x}^D = \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Sada se sistem ograničenja tipa jednakosti može pisati kao:

$$\tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} = [\boldsymbol{A}\ \boldsymbol{E}]\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{x}^B \end{bmatrix} = \boldsymbol{b}$$

Odavde se vidi da je u tom slučaju $\boldsymbol{E}\boldsymbol{x}^B = \boldsymbol{x}^B = \boldsymbol{b}$. Drugim riječima, ukoliko za nebazne varijable odaberemo problemske varijable, tada su preostale koordinate početne vršne u stvari elementi vektora resursa $\boldsymbol{b}$.

Očigledno je da je za ovako jednostavno određivanje početne vršne tačke neophodno postojanje jedinične matrice u okviru tehnološke matrice proširenog problema linearnog programiranja. Treba napomenuti da je ovako određena početna vršna tačka u pravilu prilično daleko od optimuma.

Pogledajmo sada na koji način možemo ostvariti prelaz iz jedne u drugu vršnu tačku. Ponovo ćemo poći od ograničenja tipa jednakosti:

$$\tilde{\boldsymbol{A}}\,\tilde{\boldsymbol{x}}=x_1\,\boldsymbol{a}_1+x_2\,\boldsymbol{a}_2+\ldots+x_{n+m}\,\boldsymbol{a}_{n+m}=\boldsymbol{b}$$

Ovdje $\boldsymbol{a}_j$ predstavlja kolonu matrice $\tilde{\boldsymbol{A}}$ koju množi varijabla x_j vektora $\tilde{\boldsymbol{x}}$. Imajući ponovo u vidu da u početnoj vršnoj tački n od ukupno $n+m$ elemenata vektora $\tilde{\boldsymbol{x}}$ ima vrijednost 0, ova jednačina se reducira na jednačinu:

$$x_1^1\boldsymbol{a}_1{}^1+x_2^1\boldsymbol{a}_2{}^1+\ldots+x_m^1\boldsymbol{a}_m{}^1=\boldsymbol{b}$$

Vrijednost kriterija u ovoj tački možemo odrediti na osnovu reduciranog izraza:

$$P(\boldsymbol{x}^1)=c_1^1x_1^1+c_2^1x_2^1+\ldots+c_m^1x_m^1$$

Prelazeći od jedne do druge vršne tačke, na N-toj iteraciji će ovi izrazi biti:

$$\begin{gathered}x_1^N\boldsymbol{a}_1{}^N+x_2^N\boldsymbol{a}_2{}^N+\ldots+x_m^N\boldsymbol{a}_m{}^N=\boldsymbol{b}^N\\ P(\boldsymbol{x}^N)=c_1^N x_1^N+c_2^N x_2^N+\ldots+c_m^N x_m^N\end{gathered}$$

Treba voditi računa da za svaku iteraciju k u općem slučaju $x_j{}^k$ predstavlja neku drugu varijablu vektora $\tilde{\boldsymbol{x}}$. Isto vrijedi i za ostale vektore i skalare koji se pojavljuju u ovim izrazima.

Ako su svi vektori $\boldsymbol{a}_j{}^N$ međusobno linearno nezavisni, onda oni formiraju bazu m-dimenzionalnog prostora. Radi toga se bilo koji drugi vektor ovog prostora, pa tako i preostalih n vektora koji odgovaraju kolonama nebaznog dijela tehnološke matrice proširenog problema linearnog programiranja, može predstaviti njihovom linearnom kombinacijom:

$$\boldsymbol{a}_j{}^N=y_{1j}^N\boldsymbol{a}_1{}^N+y_{2j}^N\boldsymbol{a}_2{}^N+\ldots+y_{mj}^N\boldsymbol{a}_m{}^N\quad,\quad j=m+1,m+2,\ldots,m+n$$

Ovdje $y_{ij}{}^N$ predstavljaju koeficijente linearne kombinacije. Iskoristimo ove koeficijente da definiramo skalare $s_j{}^N$ kao:

$$s_j^N=y_{1j}^N c_1{}^N+y_{2j}^N c_2{}^N+\ldots+y_{mj}^N c_m{}^N\quad,\quad j=m+1,m+2,\ldots,m+n$$

Značenje ovih skalara će postati jasno malo kasnije. Sada ćemo reduciranom ograničenju tipa jednakosti dodati i oduzeti svaki od vektora $\boldsymbol{a}_j{}^N$ pomnožen nekim jedinstvenim nenegativnim realnim brojem θ^N :

$$x_1^N \boldsymbol{a}_1{}^N + x_2^N \boldsymbol{a}_2{}^N + \ldots + x_m^N \boldsymbol{a}_m{}^N - \theta^N \boldsymbol{a}_j{}^N + \theta^N \boldsymbol{a}_j{}^N =$$
$$= (x_1^N - \theta^N y_{1j}^N)\boldsymbol{a}_1{}^N + (x_2^N - \theta^N y_{2j}^N)\boldsymbol{a}_2{}^N + \ldots + (x_m^N - \theta^N y_{mj}^N)\boldsymbol{a}_m{}^N + \theta^N \boldsymbol{a}_j{}^N = \boldsymbol{b}^N$$
$$j = m+1, m+2, \ldots, m+n$$

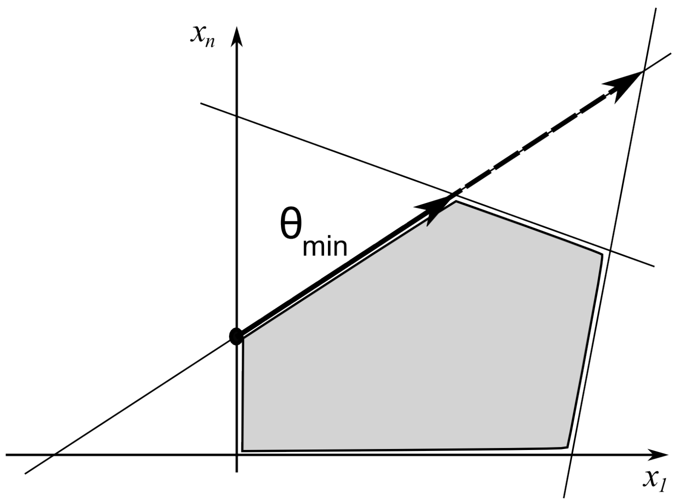
Na sličan način ćemo i reduciranom izrazu za kriterij dodati sa obje strane $\theta^N(c_j^N - s_j^N)$:

$$c_1^N x_1^N + c_2^N x_2^N + \ldots + c_m^N x_m^N + \theta^N(c_j^N - s_j^N) =$$
$$= (x_1^N - \theta^N y_{1j}^N) c_1{}^N + (x_1^N - \theta^N y_{1j}^N) c_2{}^N + \ldots + (x_m^N - \theta^N y_{mj}^N) c_m{}^N + \theta^N c_j{}^N = P(\boldsymbol{x}^N) + \theta^N(c_j^N - s_j^N)$$
$$j = m+1, m+2, \ldots, m+n$$

Potražimo sada takvo *j* da se ostvari najveće moguće povećanje vrijednosti desne strane ovog izraza za proizvoljno $\theta^N > 0$. To će se desiti za najveću pozitivnu vrijednost razlike $(c_j^N - s_j^N)$. Ovu razliku ćemo nazivati simpleksnom razlikom i ona za svako *j* označava jedinično povećanje vrijednosti kriterija za varijablu $x_j{}^N$.

Ako sada u izrazu za ograničenje tipa jednakosti za tako odabrano *j* pustimo da θ^N postepeno raste od 0 prema beskonačnosti, u nekom trenutku će se desiti da će izraz u zagradi uz neki od vektora $\boldsymbol{a}_j{}^N$ postati jednak nuli. Takva vrijednost θ^N se može odrediti kao (slika 4.7):

$$\theta^N = \min_{i,\, y_{ij}^N > 0} \frac{x_i^N}{y_{ij}^N} = \frac{x_u^N}{y_{uj}^N}$$



Slika 4.7: Određivanje vrijednosti θ^N

Ako proanaliziramo dobiveni izraz, možemo zaključiti da je ograničenje tipa jednakosti i dalje zadovoljeno, s tim da je jedan član izraza postao jednak nuli (jedna bazna varijabla je postala nebazna), dok se pojavio drugi nenulti član θ^N uz novi vektor $\boldsymbol{a}_j{}^N$ (jedna nebazna varijabla je postala bazna).

Drugim riječima, obzirom da ponovo imamo m nenultih vrijednosti koje zadovoljavaju ograničenje tipa jednakosti, radi se o m nenultih koordinata nove vršne tačke. Sve koordinate ove vršne tačke možemo odrediti prema izrazima:

$$\begin{array}{ll}
x_1^{N+1}=x_1^N-\theta^N y_{1j}^N & (\boldsymbol{a}_1{}^{N+1}=\boldsymbol{a}_1{}^N) \\
x_2^{N+1}=x_2^N-\theta^N y_{2j}^N & (\boldsymbol{a}_2{}^{N+1}=\boldsymbol{a}_2{}^N) \\
 & \vdots \\
x_u^{N+1}=\theta^N & (\boldsymbol{a}_u{}^{N+1}=\boldsymbol{a}_j{}^N) \\
 & \vdots \\
x_m^{N+1}=x_m^N-\theta^N y_{mj}^N & (\boldsymbol{a}_m{}^{N+1}=\boldsymbol{a}_m{}^N) \\
x_{m+1}^{N+1}=x_{m+2}^{N+1}=\ldots=x_{m+n}^{N+1}=0 &
\end{array}$$

Novu, bolju vrijednost kriterija možemo odrediti na osnovu poznate vrijednosti kriterija za vršnu tačku iteracije N:

$$P(\boldsymbol{x}^{N+1})=P(\boldsymbol{x}^N)+\theta^N(c_j^N-s_j^N) \quad , \quad (c_j^N-s_j^N)>0$$

Opisani izrazi definiraju jednu iteraciju simpleksnog algoritma. Transformacije se provode sve dok postoji barem jedna pozitivna simpleksna razlika, koja omogućava povećanje vrijednosti kriterija.

Proceduru simpleksnog algoritma je moguće formalizirati, tako da se olakša korištenje datih izraza.