INFORMATICA 1/85

REDUCIRANJE SIMPLEKSNE TABELE SPLOŠNE METODE SIMPLEKSOV

TALIB DAMIJ
JANEZ GRAD

UDK: 519.852.61

ISKRA DELTA, LJUBLJANA,
UNIVERZA EDVARDA KARDELJA V LJUBLJANI,
EKONOMSKA FAKULTETA BORISA KIDRIČA,
LJUBLJANA

Metoda simpleksov je iteracijski postopek, ki se odlikuje po enostavnosti in hitrosti, vendar pa je velik porabnik pomnilnika. V našem članku opišemo, kako je možno postopek simpleksov modificirati ter s tem zmanjšati zahteve po pomnilniku samo na prostor, ki ga potrebuje vhodna matrika neenačb.

REDUCTION OF THE SIMPLEX TABLEAU WITHIN THE GENERAL SIMPLEX METHOD. The general simplex method is known as a simple and fast iterative process but it as also known as a big computer storage consumer. In our paper we describe the necessary modifications of this method which enable us to reduce the size of simplex tableau to the size of the input matrix of the set of linear unequations.

1. UVOD

Metoda simpleksov je najbolj splošna metoda za numerično reševanje problemov linearnega programiranja; z njo lahko rešimo slehernı linearni program, kar je izrednega pomena za njeno uporabo pri različnih problemih optimizacije v gospodarstvu.

Metoda simpleksov je v bistvu iteracijski postopek, pri katerem v okviru vsakega iteracijskega koraka iz že znane bazne možne rešitve iščemo novo optimalnejšo možno rešitev. Odlikuje se po tem, da je zelo primerna za računalniško obdelavo, vendar pa je pri velikih sistemih tudi velik porabnik pomnilnika. Za metodo je namreč značilno, da temelji na tako imenovani simpleksni tabeli, v katero morajo biti razvrščene vse spremenljivke ter da je potrebno na vsakem koraku izračunati in shraniti v pomnilniku celotno tabelo. Tak način dela pa zahteva precej računalniškega pomnilnika, kar predstavlja osnovni problem uporabe te metode. To je tudi pomemben razlog, zaradi katerega se uporabniki raje odločajo za uporabo drugih metod, kot na primer revidirane metode simpleksov, ki so sicer bolj zapletene, vendar pa so manjši porabniki pomnilnika.

V tem prispevku opišemo, kako je možno postopek simpleksov modificirati ter s tem zmanjšati zahteve po pomnilniku samo na prostor, ki ga potrebuje vhodna matrika neenačb. Pri tem podajamo metodo simpleksov le toliko, kolikor je potrebno za spremljanje razvoja postopka reduciranja simpleksne tabele, podroben opis teoretičnih osnov te metode pa je opisan na primer v |2| in |3|.

2. SPLOŠNA METODA SIMPLEKSOV

2.1. Definicija linearnega programa

Splošni problem linearnega programiranja za minimum namenske funkcije definiramo v matematični obliki takole: določiti je treba vrednosti spremenljivk $x_1,\dots,x_s$, ki zadoščajo pogojem nenegativnosti

$$x_i \geq 0 \qquad (i=1,\dots,s)$$

in linearnim pogojem

$$a_{11}x_1 + \dots + a_{1s}x_s \geq b_1$$
$$\dots\dots\dots\dots\dots\dots$$
$$a_{m1}x_1 + \dots + a_{ms}x_s \geq b_m$$

tako, da ima namenska funkcija

$$c_1x_1 + \dots + c_sx_s$$

minimum. Vsi koeficienti so realna števila.

Zgornji zapis linearnega programa ni uporaben za numerično reševanje, ker nastopajo v obliki neenačb. Take pogoje spremenimo v enačbe s privzemom dopolnilnih spremenljivk. Če odštejemo spremenljivke $x_{s+1},\dots,x_{s+m}$ zaporedoma od levih strani neenačb, preidejo te neenačbe v enačbe. Dopolnilne spremenljivke upoštevamo tudi v namenski funkciji, kjer je $c_{s+1} = \dots = c_{s+m} = 0$.

Da bi omogočili začetek numeričnega reševanja, vrinemo v vsako pogojno enačbo še po eno umetno spremenljivko, torej k levi strani enačb prištejemo zaporedoma spremenljivke $x_{s+m+1},\dots,x_{s+2m}$. Umetne spremenljivke upoštevamo tudi v namenski funkciji, kjer je $c_{s+m+1} = \dots = c_{s+2m} = M$.

Tako dopolnjen linearni program lahko zapišemo v matrični obliki takole:

potrebno je izračunati vektor x, ki zadošča neenačbi

$x \geq 0$

in enačbi

$Ax = b$

tako, da ima namenska funkcija

cx

minimum. Pri tem so:

$$x = \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} \qquad A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ a_{m1} & & a_{mn} \end{bmatrix} \qquad b = \begin{bmatrix} b_1 \\ \cdot \\ \cdot \\ \cdot \\ b_m \end{bmatrix}$$

$$c = [c_1 \; \cdots \; c_n]$$

kjer je $n = s+2m$.

Problem linearnega programiranja zapišemo lahko v vektorski obliki tudi takole:

potrebno je izračunati vektor x, ki zadošča neenačbi

$x \geq 0$

in enačbi

$x_1 p^1 + \ldots + x_n p^n = p^0$

tako, da ima namenska funkcija

cx

minimum. Pri tem so:

$$p^0 = \begin{bmatrix} b_1 \\ \cdot \\ \cdot \\ \cdot \\ b_m \end{bmatrix} \qquad p^1 = \begin{bmatrix} a_{11} \\ \cdot \\ \cdot \\ \cdot \\ a_{m1} \end{bmatrix} \ldots p^n = \begin{bmatrix} a_{1n} \\ \cdot \\ \cdot \\ \cdot \\ a_{mn} \end{bmatrix}$$

Vektorji $p^0, p^1, \ldots, p^s, p^{s+1}, \ldots, p^{s+m}, p^{s+m+1}, \ldots p^n$ so vektorji m-dimenzionalnega vektorskega prostora V^m. Med njimi sestavlja zadnjih m vektorjev sistem m linearno neodvisnih vektorskih enot. To so vektorji

$$p^{s+m+1} = (1, 0, \ldots 0), \ldots, p^n = (0, 0, \ldots 1)$$

2.2. Določitev začetne baze

Dogovorimo se, da začetno bazo vektorskega prostora V^m določimo iz vektorjev umetnih spremenljivk ter vektorjev dopolnilnih spremenljivk, ki so vektorske enote. Ti vektorji torej sestavljajo bazno matriko B, ostali vektorji dopolnilnih spremenljivk pa naj sestavljajo matriko, ki jo označimo z D.

Potem lahko matriko A napišemo kot:

$$A = [F, D, E] \qquad (1)$$

Struktura vektorjev bazne matrike B je znana, saj je sestavljena iz vektorskih enot. Koeficienti baznih vektorjev v namenski funkciji so ravno tako znani in sicer so enaki M (kjer je M primerno velika vrednost) za umetne spremenljivke in 0 za dopolnilne spremenljivke.

Isto velja za matriko D, saj ima vsak vektor te matrike samo en element različen od nič in sicer (-1). Koeficienti vektorjev matrike D v namenski funkciji pa so enaki 0.

Iz tega sledi, da je struktura vektorjev, ki so v matrikah D in B ter njihovi koeficienti v namenski funkciji že vnaprej določeni s predpisanimi oblikami pogojnih enačb in neenačb linearnega programa.

Če najdemo način, s katerim lahko na začetku reševanja linearnega programa ugotovimo, kateri vektorji pridejo v sestavo začetne baze ter v ustreznem trenutku reševanja, kateri vektor matrike D je potrebno transformirati, potem lahko opustimo formiranje matrik D in B ter omejimo reševanje linearnega problema samo na matriko dimenzije m x s. Z računalniškega vidika pomeni tako zmanjšanje dimenzije simpleksne tabele zadostno zmanjšanje računalniškega pomnilnika, potrebnega za izvajanje programov, ki rešujejo probleme linearnega programiranja z uporabo splošne metode simpleksov.

V ta namen naj bo q vektor reda m, ki ima za svoje elemente simbole relacij v pogojnih enačbah in neenačbah obravnavanega linearnega programa ($\geq$, $=$, $\leq$). Ta vektor ima odločilno vlogo v modificiranem postopku, kajti s pomočjo tega vektorja lahko določimo vektorje matrik D in B ter njihove koeficiente v namenski funkciji.

Za določitev začetne baze je potrebno ugotoviti indekse baznih vektorjev ter njihove koeficiente v namenski funkciji.

Zato definirajno naslednje vrednosti:

g^1 naj bo število, ki nam pove, koliko je pogojnih neenačb v linearnem programu ($\geq$, $\leq$);

gq naj bo število pogojnih neenačb ali enačb oblike ($\geq$) ali ($=$);

d naj bo število vhodnih in dopolnilnih spremenljivk v sestavi matrike A; to vrednost dobimo kot

$d = s + gl$

n je število vseh spremenljivk matrike A; to vrednost dobimo kot

$n = s + gl + gq$

Z njimi moremo določiti indekse baznih vektorjev ter njihove koeficiente v namenski funkciji brez oblikovanja matrik D in B.

Nadalje definirajmo še naslednja vektorja reda m:

c^B, ki vsebuje koeficiente baznih vektorjev v namenski funkciji in

v^B, ki ima za elemente indekse baznih vektorjev.

Postopek kreiranja začetne baze je takle:

A) Izračunajo se vrednosti gl, gq, d in n.

B) Glede na vrednost elementa q_i, $i = 1,\ldots,m$, se izračuna indeks i-tega baznega vektorja in njegov koeficient v namenski funkciji z naslednjim algoritmom:

1) $i = 1$

2) q_i ima vrednost ($\geq$); potem je

$gq = gq-1$

$gl = gl-1$

$v_i^B = n-gq$

$c_i^B = M$

3) q_i ima vrednost ($\le$); potem je

$gl = gl-1$

$v_i^B = d-gl$

$c_i^B = 0$

4) q_i ima vrednost ($=$); potem je

$gq = gq-1$

$v_i^B = n-gq$

$c_i^B = M$

5) $i = i+1$

6) Če je $i \le m$, se postopek nadaljuje na točki 2).

S takim postopkom kreiranja začetne baze eliminiramo oblikovanje matrike B, kajti vse značilnosti te matrike so dosegljive z uporabo vektorjev c^B in v^B.

2.3. Izboljšanje možne rešitve

V izrazu (1) je A matrika dimenzije m x n. Vendar zaradi zgoraj prikazanega postopka eliminacije matrike B ter zaradi postopka eliminacije matrike D, katerega opis je podan v naslednjem, velja, da je A matrika dimenzije m x s, kar označimo z $A_{m,s}$. Kljub temu pa mora matrika A ohraniti vse značilnosti celotne simpleksne tabele, potrebne za reševanje linearnega programa.

V ta namen definirajmo naslednja vektorja:

c^A naj bo vektor reda s, ki ima za svoje elemente koeficiente vektorjev matrike $A_{m,s}$ v namenski funkciji,

v^A pa naj bo vektor reda s, ki ima za svoje elemente indekse vektorjev matrike $A_{m,s}$.

Iz prvotne baze dobimo novo bazo tako, da kak njen vektor zamenjamo s kakim drugim vektorjem, ki ga še ni v bazi.

Vzemimo, da odstranimo iz baze vektor p^r in ga zamenjamo z vektorjem p^k.

Za vsako možno rešitev ima namenska funkcija določeno vrednost. Pri prehodu iz prvotne v novo možno rešitev se vrednost namenske funkcije izboljša.

Definirajmo sledeče količine:

$$z_j = c_1^B a_{1j} + \dots + c_m^B a_{mj}, \quad j = 1,\dots,s$$

kjer z_j pomeni vrednost vektorja p^j, izraženega z baznimi vektorji, in

$$z_0 = c_1^B p_1^0 + \dots + c_m^B p_m^0$$

kjer z_0 pomeni vrednost namenske funkcije, ki ustreza prvotni bazi.

V matriko A vpeljemo (m+1) vrstico, katere koeficienti $a_{m+1,j}$ so definirani z

$$a_{m+1,j} = z_j - c_j^A, \quad j = 1,\dots,s$$

Razlika med prvotno in novo vrednostjo namenske funkcije je tem večja, čim večja je diferenca $z_k - c_k^A$. Ker želimo to razliko povečati, da dobimo čim boljšo novo možno rešitev, uvedemo v novo bazo tisti vektor p^j, ki mu ustreza največja pozitivna diferenca $z_j - c_j^A$. Vzemimo, da ustreza vektorju p^k največja diferenca

$$z_k - c_k = \max_j (z_j - c_j^A) \qquad (2)$$

kjer je $z_j - c_j^A > 0$ in $j = 1,\dots,s$.

Če nobena od diferenc ni pozitivna, obstoječe rešitve ni mogoče več izboljšati, zato je to že optimalna rešitev.

2.4. Zamenjava baze

V bazo uvedemo tisti vektor p^k, pri katerem ima izraz (2) največjo vrednost, iz baze pa odstranimo vektor p^r, ki ga določimo tako, da vsako komponento p^0 delimo s homologno komponento vektorja p^k, pri tem pa upoštevamo samo pozitivne komponente vektorja p^k, najmanjši od teh ulomkov določa vektor p^r:

$$\theta = \min_i \frac{x_i}{a_{ik}} = \frac{x_r}{a_{rk}}$$

kjer $a_{ik} > 0$, za $1 \le i \le m$.

Če noben koeficient a_{ik} ni večji od 0, je rešitev neomejena.

Vrstico (m+1) vpeljemo tudi za vektor p^0, katerega koeficient je definiran s

$$p_{m+1}^0 = z_0$$

Vse koeficiente v simpleksni tabeli (koeficienti vektorja p^0 in matrike $A_{m+1,s}$) transformiramo po naslednjem transformacijskem zakonu:

a) $$x_r' = \frac{x_r}{a_{rk}}$$

$$a_{rj}' = \frac{a_{rj}}{a_{rk}}, \text{ kjer } j = 1,\dots,s \qquad (3)$$

b) $$x_i' = x_i - \frac{x_r}{a_{rk}} \cdot a_{ik}$$

$$a_{ij}' = a_{ij} - \frac{a_{rj}}{a_{rk}} \cdot a_{ik}$$

kjer je $i = 1,\dots,m+1$, $i \ne r$ in $j = 1,\dots,s$.

Uvedba vektorja p^k v bazo in odstranitev vektorja p^r iz baze se izvrši tako, da se v v_r^B vnese vrednost iz v_k^A (indeks vektorja p^k), v c_r^B pa se vnese vrednost iz c_k^A (koeficient vektorja p^k v namenski funkciji). Prvotni vrednosti komponent v_r^B in c_r^B se predhodno shranita v postavkah, ki ju označimo z vbr in cbr. Torej je:

$vbr = v_r^B$

$cbr = c_r^B$

$v_r^B = v_k^A$

$c_r^B = c_k^A$

2.5. Zamenjava vektorjev matrike A

Matrika A ima na začetku numeričnega reševanja isto strukturo kot matrika F, torej predstavlja matriko vektorjev vhodnih spremenljivk. Prav tako pa sta na začetku numeričnega reševanja v^A in c^A vektorja indeksov vhodnih spremenljivk oziroma njihovih koeficientov v namenski funkciji. V matriki A in v vektorjih v^A ter c^A se v dani iteraciji vrši zamenjava vektorjev, njihovih indeksov ter koeficientov v namenski funkciji.

Smisel zamenjave vektorjev matrike A je v naslednjem:

1) izkoriščanje prostora vektorja p^k v matriki A, saj postane vektor p^k po uvedbi v bazo vektorska enota;
2) ohranitev značilnosti vektorjev dopolnilnih spremenljivk v simpleksni tabeli tako, da ob določenih pogojih dane iteracije uvedemo ustrezni dopolnilni vektor namesto vektorja p^k v matriko A;
3) ohranitev značilnosti vektorja p^r, ki ga odstranimo iz baze z uvedbo tega vektorja v matriko A na mesto vektorja p^k zaradi njegove možne vrnitve v bazo.

Izvršitev zamenjave vektorjev matrike A je odvisna od vektorjev q in c^B. S pomočjo teh vektorjev lahko sklepamo, če je potrebno izvršiti zamenjavo, poleg tega pa, kateri vektor je treba v dani iteraciji uvesti v matriko A.

Definirajmo vrednost ql, ki naj pove, koliko pogojnih enačb je v prvih r enačbah ali neenačbah linearnega programa. To vrednost potrebujemo za ugotavljanje indeksa ustreznega dopolnilnega vektorja, ki ga moramo uvesti v matriko A.

Preden realiziramo postopek zamenjave vektorjev matrike A, moramo zaradi degeneracije in zamenjave vektorjev shraniti značilnosti vektorjev matrike A, ki imajo po odstranitvi iz baze možnost ponovne vrnitve v bazo. To so vektorji vhodnih in dopolnilnih spremenljivk.

V ta namen vpeljemo vektor v reda d, ki ima za svoje elemente značilnosti teh vektorjev. Vrednost elementa v_j ($j = 1, \ldots, d$) mora zagotavljati hitro ugotavljanje mesta in strukture vektorja p^j v dani iteraciji.

Postopek oblikovanja vektorja v:

A) Za $j = 1, \ldots, s$ velja

   $v_j = j$

   Gre za vektorje vhodnih spremenljivk, ki sestavljajo matriko A na začetku numeričnega reševanja.

   p^j je vektor j-te vhodne spremenljivke, njegove komponente pa so shranjene v j-tem stolpcu matrike A.

B) Za $j = s + 1, \ldots, d$

   1. $j = s$, $r = 0$
   2. $j = j + 1$
   3. $r = r + 1$
   4. q_r ima vrednost ($\leq$); potem je

      $v_j = d + r$

      taka vrednost elementa v_j pomeni, da je p^j vektorska enota, katere r-ti element je enak 1 in se trenutno nahaja na r-tem mestu v bazi.
   5. q_r ima vrednost ($\geq$); potem je

      $v_j = -(s + r)$

      negativna vrednost elementa v_j pomeni, da je p^j nebazni vektor dopolnilne spremenljivke x_j, njegov r-ti element pa je enak -1.
   6. q_r ima vrednost (=); če $r < m$, se postopek nadaljuje na točki 3, sicer pa se konča.
   7. če je $j < d$, se postopek nadaljuje na točki 2.
   8. konec postopka.

Postopek zamenjave vektorjev matrike A:

A) Vrednost q_r je ($\geq$) in vrednost cbr (vrednost c_r^B pred zamenjavo) je enaka M.

   Iz tega sledi, da je p^r umetni vektor, zato ga po odstranitvi iz baze zanemarimo, ker nima nobene možnosti vrnitve v bazo, namesto transformiranega vektorja p^k uvedemo ustrezni dopolnilni vektor p^d.

   Vsi elementi vektorja p^d so enaki nič, razen r-tega elementa, ki je enak (-1). Koeficient p_{m+1}^d pa je definiran s

   $p_{m+1}^d = - c_r^B$ (vrednost pred zamenjavo).

   Za uvedbo p^d v matriko A je potrebno

   - shraniti vrednost vektorja p^k; za ta namen definirajmo vektor p reda (m+1), kjer

     $p_i = a_{ik}$, pri $i = 1, \ldots, m+1$
   - uvesti vektor p^d v k-ti stolpec matrike A, kjer

     $a_{ik} = 0$, za $i = 1, \ldots, m, i \neq r$

     $a_{rk} = -1$

     $a_{m+1,k} = -$ cbr
   - za r izračunati ql
   - ugotoviti indeks vektorja p^d in njegov koeficient v namenski funkciji ter ažurirati vektor v s tem, da

     $j = v_k^A$

     $u = s + r - ql$

     $c_k^A = 0$

     $v_k^A = u$

     $v_u = k$

     $v_j = d+r$

B) Vrednost q_r je (=) in vrednost cbr (vrednost c_r^B pred zamenjavo) je enaka M.

   V tem primeru ne izvršimo nobene zamenjave, ker je p^r umetni vektor, poleg tega pa ne obstoji v tej situaciji kak ustrezni dopolnilni vektor.

   Za uvedbo vektorja p^k v bazo je treba ažurirati vektor v s tem, da

   $u = v_k^A$

   $v_u = d+r$

C) Za vse druge možnosti moramo ugotoviti naslednje:

1. vektor p^r je v sestavi matrike A; potem se zadovoljimo z ažuriranjem vektorja v s tem, da

$$v_{(vbr)} = vbr$$

2. izvršiti moramo zamenjavo vektorjev, tako da uvedemo namesto vektorja p^k v matriko A vektor p^r zaradi možnosti njegove ponovne vrnitve v bazo.

Vektor p^r kot bazni vektor ima strukturo vektorske enote, katere r-ta komponenta je enaka 1, koeficient p^r_{m+1} pa je enak nič.

Za uvedbo p^r v matriko A je potrebno:

- shraniti vrednost vektorja p^k, kjer

$$p_i = a_{ik} \quad , \quad \text{pri } i = 1, \ldots, m+1$$

- uvesti vektor p^r v matriko A s tem, da postavimo

$$a_{ik} = 0 \quad , \quad \text{pri } i = 1, \ldots, m, \; i \neq r$$

$$a_{rk} = 1$$

$$a_{m+1,k} = 0$$

- ugotoviti indeks vektorja p^r in njegov koeficient v namenski funkciji ter ažurirati vektor v s tem, da

$$j = v^A_k$$

$$v^A_k = vbr$$

$$c^A_k = cbr$$

$$v_{(vbr)} = k$$

$$v_j = d+r$$

2.6. Degeneracija

Število θ oziroma p^r določimo z najmanjšim izmed ulomkov x_i/a_{ik}, $(a_{ik} > 0)$. Če je teh najmanjših ulomkov več, dobimo novo možno rešitev, ki ima manj kot m pozitivnih komponent in je zato degenerirana. V primeru degeneracije ni enolično določen vektor p^r, ki ga odstranimo iz baze.

Denimo, da pridemo v linearnem programu do degeneracije in sicer zato, ker sta vsaj dva od navedenih ulomkov enaka. Vzemimo, da sta enaka ulomka x_r/a_{rk} in x_s/a_{sk}. Zato primerjamo druga dva ulomka a_{r1}/a_{rk} in a_{s1}/a_{sk}; med njima izberemo tistega, ki je najmanjši. Če sta pa tudi ta dva ulomka enaka, še ne pridemo do odločitve in naprej primerjamo še naslednja dva ulomka a_{r2}/a_{rk} in a_{s2}/a_{sk} ter izberemo tistega, ki je manjši. Če še ne pridemo do odločitve, nadaljujemo primerjanje, dokler ne pridemo do nje.

Denimo, da ugotovimo s takim primerjanjem, da je ulomek y_r/a_{rk} manjši od ulomka y_s/a_{sk}. Število r določa zato enolično vektor p^r, ki ga odstranimo iz baze |2|.

Zaradi postopka zamenjave vektorjev, je potrebno v primeru degeneracije uporabiti vektor v, kjer element v_j $(j = 1,\ldots,d)$ rabi za ugotovitev značilnosti (indeks, struktura in mesto) vektorja p^j:

- indeks elementa v_j predstavlja indeks vektorja p^j,
- vrednost elementa v_j pa omogoča ugotoviti strukturo in pozicijo vektorja p^j s tem, da

a) če $v_j > d$

pomeni, da se vektor p^j nahaja v bazi, njegov r-ti element je enak 1 in vrednost indeksa r dobimo z

$$r = v_j - d$$

b) če $v_j \leq s$ in $v_j \neq k$

pomeni, da je vektor p^j v sestavi matrike A, njegova struktura pa daje v_j-ti stolpec matrike A

c) če $v_j < 0$

pomeni, da je p^j vektor dopolnilne spremenljivke x_j, njegov r-ti element enak -1 in vrednost indeksa r dobimo z

$$r = -(v_j + s)$$

3. POSTOPEK REŠEVANJA Z METODO SIMPLEKSOV

Postopek reševanja linearnega programa z metodo simpleksov obsega naslednje korake:

1) - Določitev začetne baze vektorskega prostora V^m z vključitvijo postopka kreiranja začetne baze (podpoglavje 2.2.)
   - Vstavitev začetnih vrednosti vektorjev c^A in v^A

2) Vključitev postopka oblikovanja vektorja v (podpoglavje 2.5.)

3) Izračun naslednjih vrednosti:

$$z_j = c^B_1 \; a_{1j} + \ldots + c^B_m a_{mj} \; , \; \text{za } j = 1,\ldots,s$$

$$a_{m+1,j} = z_j - c^A_j$$

$$z_0 = x_{m+1} = c^B_1 x_1 + \ldots + c^B_m x_m$$

$$= c^B_1 b_1 + \ldots + c^B_m b_m = p^0_{m+1}$$

4) Določitev vektorja p^k oziroma indeksa k:

$$z_k - c^A_k = \max_{1 \leq j \leq s} (z_j - c^A_j), \; \text{za } z_j - c^A_j > 0$$

V primeru, da je $z_k - c^A_k \leq \epsilon$, kjer je ϵ primerno majhno vnaprej predpisano pozitivno število, je obstoječa možna rešitev optimalna in iteracijski postopek se konča.

5) Če so vse vrednosti $a_{ik} < 0$, $(i = 1,\ldots,m)$, se iteracijski postopek konča in rešitev je neomejena.

6) Določitev vektorja p^r oziroma indeksa r:

$$0 < \theta = \frac{x_r}{a_{rk}} = \min_{1 \leq i \leq m} (x_i/a_{ik}), \; \text{za } a_{ik} > 0$$

Če doseže vrednost (x_i/a_{ik}) minimum pri več vrednostih indeksa i, je rešitev degenerirana.

7) Uvedba vektorja p^k v bazo (podpoglavje 2.4.)

8) Zamenjava vektorjev matrike A z uporabo postopka, opisanega v podpoglavju 2.5.

9) Transformacija tabele:

$$x'_r = \frac{x_r}{p_r}$$

$$a'_{rj} = \frac{a_{rj}}{p_r}, \quad \text{za } j = 1, \ldots, s$$

$$x'_i = x_i - \frac{x_r}{p_r} \cdot p_i, \quad \text{za } i = 1, \ldots, m+1, \; i \neq r$$

$$a'_{ij} = a_{ij} - \frac{a_{rj}}{p_r} \cdot p_i$$

10) Vrnitev k točki (3).

4. ZAKLJUČEK

Glavna značilnost prikazanega načina uporabe splošne metode simpleksov je, da v dani iteraciji vključuje v matriko A samo tiste vektorje, ki določajo vektor p^k v sledeči iteraciji.

Posledica tega je zmanjšanje računalniškega pomnilnika, potrebnega za hranjenje matrike A ter hitrejša transformacija simpleksne tabele v dani iteraciji. To pa omogoča ekonomičnejše in hitrejše obdelave problemov linearnega programiranja.

Na koncu naj omenimo primer obdelave linearnega modela s 100 spremenljivkami in 100 pogojnimi neenačbami. Obdelava tega modela s formiranjem celotne simpleksne tabele zahteva (128.512 bytov), medtem pa obdelava istega modela z uporabo reducirane simpleksne tabele zahteva (48.128 bytov) računalniškega pomnilnika.

5. LITERATURA

1) Grad, J., Strukturno programiranje linearnega programa, Ekonomska revija, Ljubljana, 28 (1977), str. 289-300.

2) Vadnal, A., Rešeni problemi linearnega programiranja, Knjižnica Sigma, Mladinska knjiga, Ljubljana, 1977.

3) Vadnal, A., Primjena matematičnih metoda u ekonomiji, Informator, Zagreb, 1980.