Tihomir Hunjak
Fakultet organizacije i informatike
V a r a ž d i n

UDK: 65.012.1
Stručni rad

PRIKAZ BALASOVOG ALGORITMA ZA 0-1 PROGRAMIRANJE

U uvodnom dijelu rada govori se o cjelobrojnom i nula-jedan programiranju.

Daljnji tekst sadrži prikaz Balasovog algoritma za nula-jedan programiranje (poznatog još pod nazivom metoda djelomičnog nabrajanja). Rješavanje zadataka pomoću navedenog algoritma prikazano je detaljno na jednom primjeru, a za drugi su pregledno navedeni algoritamski koraci koji se primjenjuju tokom rješavanja zadatka.

1. UVOD

Poznato je da se velik broj ekonomskih (i ne samo ekonomskih) problema može prikazati matematičkim modelima koji se rješavaju metodama poznatim pod zajedničkim nazivom linearno, odnosno nelinearno programiranje. Osim mnogo upotrebljavanih metoda općeg karaktera,kojima se može rješavati velik broj problema linearnog programiranja (takva je npr. dobro poznata simpleks metoda), ponekad se pokaže potreba za algoritmima specijalne namjene, ovisno o specifičnostima problema koji se rješava.Efikasnost tih algoritama ima i svoju cijenu - oni obično nisu primjenjivi na veći broj problema. Takav je slučaj i s problemima nula-jedan programiranja i njihovim rješavanjem.

Pod problemima 0-1 programiranja podrazumijevaju se problemi programiranja u kojima varijable uzimaju vrijednosti 0 ili 1. Velik broj realnih problema, za koje je moguće formirati matematičke modele s 0-1 varijablama, bio je razlogom za izdvajanje ove klase problema iz šire klase problema cjelobrojnog programiranja. Takvi su problemi na primjer:

- problem trgovačkog putnika
- problem dostave
- problem ranca
- višestruki problem ranca
- problem ravnoteže tekuće vrpce
- problem rasporeda poslova
- problem investicijskog odlučivanja.

Izdvajanje 0-1 programiranja kao posebne grane programiranja bilo je omogućeno i pronalaženjem posebnih metoda za rješavanje tih problema. Te metode i algoritmi razlikuju se prema pristupu u rješavanju problema. Tako se razlikuju:[1)]

- algebarski pristup
- kombinatorni pristup
- nabrajanje
- heuristički pristup.

Neki problemi cjelobrojnog linearnog programiranja mogu se uz stanovite uvjete[2)], prevesti u probleme 0-1 programiranja i na taj način se kod njihovog rješavanja mogu koristiti odgovarajuće metode koje imaju odredjenih prednosti[3)] pred metodama cjelobrojnog programiranja. Isto vrijedi i za odredjenu klasu problema linearnog programiranja s varijablama diskretnih vrijednosti (ne nužno cjelobrojnim).

U ovom članku prikazan je Balasov aditivni algoritam za rješavanje problema 0-1 linearnog programiranja, u kojem se koristi metoda djelomičnog nabrajanja. Ovaj algoritam objavljen je [4)] 1965. godine, a ovdje je dan prikaz prema originalu, pogodan za ručno rješavanje problema. U članku je u potpunosti prikazan tok rješavanja jednog primjera tom metodom. Kako je rješavanje problema većih dimenzija teško zamislivo bez pomoći elektronskog računskog stroja, za ovaj algoritam načinjen je i program za sistem B-1714. Taj program sadrži i odredjene modifikacije originalnog algoritma, a bit će prikazan u posebnom članku.

2. IDEJA BALASOVOG ADITIVNOG ALGORITMA

Opći oblik linearnog programa s 0-1 varijablama može se formulirati na slijedeći način:

odrediti x' koji minimizira (maksimizira)

$$c'x' \qquad (1)$$

$$\text{uz ograničenja } A'x' \gtreqless b' \qquad (2)$$

$$x'_j = 0 \text{ ili } 1, \; j \in N \qquad (3)$$

---

1) *S.Ashour i A.R.Char: "Computational Experience on Zero-One Programming Approach to verious Combinatorial Problems, Journal of the Operations Research Society of Japan, Vol.13, No.2, 1970, str.78-108.*

2) *Lj.Martić, "Matematičke metode za ekonomske analize II, Narodne novine, Zagreb, 1972, str. 256-257.*

3) *S.Ashour i A.R.Char, op.cit.*

4) *E.Balas, "An Additive Algorithm for Solving Linear Programs with Zero-One Variables", Operations Research, Vol.13, 1965, str. 517-546.*

pri čemu je x' n komponentni stupac vektor, c' je dani n komponentni redak vektor, $A' = (a_{ij})$ je matrica tipa $q \times n$ i b' je stupac vektor od q komponenata.

Skup $\{1,2,\ldots,q\}$ označit će se s Q, a skup $\{1,2,\ldots,n\}$ s N.

Da bi se ovako formulirani problem mogao riješiti pomoću aditivnog algoritma, on se prevodi u nešto drugačiji oblik koji zadovoljava zahtjeve:

- sva ograničenja moraju biti tipa $<$
- koeficijenti funkcije cilja moraju biti nenegativni.

Ako problem (1'), 2'), (3') ne ispunjava ove zahtjeve, on se na željeni oblik može dovesti slijedećim transformacijama:

a) sve jednadžbe se zamijene s dvije nejednadžbe
b) sve nejednadžbe tipa $>$ množe se s -1
c) ovisno o predznaku koeficijenta varijable u funkciji cilja i o tome da li se rješava problem maksimuma ili problem minimuma uvode se nove varijable prema slijedećoj shemi:

$$x_j = \begin{cases} & \text{za max} & \text{za min} \\ x'_j & \text{kada je } c'_j < 0 & \text{kada je } c'_j > 0 \\ 1-x'_j & \text{kada je } c'_j > 0 & \text{kada je } c'_j < 0 \end{cases}$$

Uz ove transformacije i uz uvodjenje m-komponentnog nenegativnog oslabljenog vektora y problem prelazi u oblik:

pronaći x takav da vrijedi

$$z = cx = \min \qquad (1)$$
$$\text{uz uvjete } Ax + y = b \qquad (2)$$
$$x_j = 0 \text{ ili } 1 \ (j \varepsilon N) \qquad (3)$$
$$y \geq 0 \qquad (4)$$

gdje je $c > 0$ i x, c, A i b su dobiveni iz x', c', A', b'.

Aditivni algoritam je načinjen tako da bez obzira na zadani problem transformirani problem je problem minimuma.

Dimenzija od x i c ostaje n. Dimenzija od b se u odnosu na dimenziju od b' može promijeniti pa se nju označi s m i uvodi oznaka M za skup $\{1,2,\ldots,m\}$. Očito vrijedi $m>q$.

Problem (1), (2), (3), (4) se označi s P.
S a_j označi se j-ti stupac matrice A.

(n+m) dimenzionalni vektor u(x,y) će se zvati rješenje ako zadovoljava (2) i (3), moguće rješenje ako zadovoljava (2), (3) i (4),a optimalno (moguće) rješenje ako zadovoljava (1), (2), (3) i (4).

Neka je P^s oznaka za linearni program definiran s (1), (2),(4) i ograničenjima

$$x_j > 0 \quad (j \varepsilon N) \quad (3a)$$

$$x_j = 1 \quad (j \varepsilon J_s) \quad (3b_s)$$

gdje je J_s podskup od N (kada je s=o to je problem P^o dan s (1), (2), (3a) i (4).

Kod rješavanja algoritam počinje s linearnim programom P^o s vektorom $(u^o=(x^o,y^o) = (0,b)$ koji je očito dualno moguće rješenje za P^o (zbog c > 0).

Cdgovarajuća baza sastoji se od jedinične matrice

$$I_{(m)} = (e_i) \quad (i=1,....,m).$$

Za one i koji imaju svojstvo $y_i^o < 0$, bira se, u skladu s odredjenim pravilom (sva pravila koja se spominju u ovom pregledu aditivnog algoritma bit će precizno formulirana u prikazu algoritma po koracima) vektor a_{j1} takav da je $a_{ij_1} < 0$, i uvodi ga se u bazu. No, umjesto uvodjenja vektora a_{ij_1} u bazu na mjesto vektora e_i, kao što je to slučaj kod dualne simpleks metode, dodaje se problemu P^o novi uvjet $x_j = 1$ u malo promijenjenom obliku

$$-x_j + Y_{m+1} = -1$$

pri čemu je Y_{m+1} artificijelna varijabla.

Time se dolazi do problema p^1 s $J_1 = \{j_1\}$ koji obuhvaća (1), (2), (3a) i (4) s dodatnim ograničenjem

$$x_j = 1 \quad (3b_1).$$

Lako se vidi da je skup $x_j = 0$ (j ϵ N), yi = bi (i ε M), $y_{m+1} = -1$ dualno moguće rješenje za problem P^1.

U proširenoj bazi $I_{(m+1)} = (e_i)$ (i=1,...,m+1), (m+1). jedinični vektor e_{m+1} odgovara vektoru y_{m+1}. On je na mjestu jediničnog vektora e_{m+1} pa se zbog toga uvodi u bazu vektor a_j, odnosno varijabla x_j poprima vrijednost 1 u novom rješenju problema P^1 koje je očito dualno moguće.

Budući da artificijelna varijabla y_{m+1} dobiva vrijednost 0 i ubuduće ne igra nikakvu ulogu u algoritmu, može biti ispuštena i novo rješenje može se zbog toga pisati u obliku

$$u^1 = (x^1, y^1) = (x_1^1, \ldots, x_n^1, y_1^1, \ldots, y_m^1).$$

Zbog toga je novo dualno moguće rješenje problema P^1 vektor $u^1 = (x^1, y^1)$ čije su komponente

$$x_j = \begin{cases} 1 & (j = j_1) \\ 0 & (j \in N - \{j_1\}) \end{cases} \qquad y_i^1 = b_i - a_{ij} \quad (i \in M).$$

Vidi se da se u formiranju novih rješenja koriste operacije zbrajanja i odbijanja pa je to bio razlog da se algoritmu dalo ime aditivni algoritam.

Ako vektor-rješenje u^1 još uvijek ima negativne komponente, tada se u skladu s gore navedenim pravilom izabere drugi vektor a_{j2} za uvodjenje u bazu, a problemu P^1 dodaje novo ograničenje $x_{j2} = 1$, u obliku

$$-x_{j2} + y_{m+2} = -1$$

gdje je y_{m+2} druga artificijelna varijabla.

To dovodi do problema P^2 sastavljenog od (1), (2), (3a) i (4) i dodatnog skupa ograničenja (3b) sačinjenog od $x_{j1}=1$, $x_{j2}=1$. Skup

$$x_{j1} = 1, \quad x_j=0 \quad |j \in (N-\{j_1\})|, \quad y_i = b_i - a_{ij1} \ (i \in M), \quad y_{m+2} = -1$$

je dualno moguće rješenje od P^2. Vektor a_{j2} uveden je umjesto e_{m+2}, a x_{j2} uzima vrijednost 1 u novom rješenju za P^2, što očigledno znači da je ono dualno - moguće.

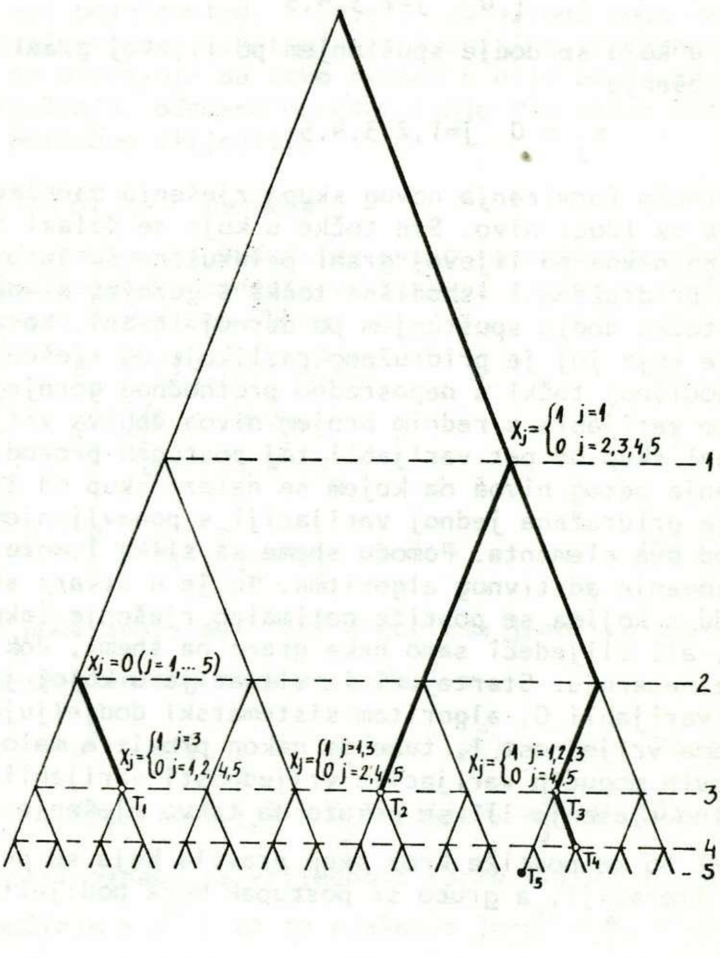
Kako varijabla y_{m+2} neće ubuduće igrati nikakvu ulogu u algoritmu (jer cijelo vrijeme ima vrijednost 0), može se ispustiti (kao što je to bio slučaj s varijablom y_{m+1}) i novo dualno moguće rješenje za problem P^2 je $u^2 = (x^2, y^2)$, gdje je

$$x_j^2 = \begin{cases} 1 & (j = j_1, j_2) \\ 0 & j \in (N - \{j_1, j_2\}) \end{cases}, \quad y_i^2 = y_i^1 - a_{ij_2}$$

Taj postupak se ponavlja tako dugo dok se ne dodje do rješenja u^s kojem su sve komponente nenegativne, ili dok se ne pokaže da takvo rješenje za problem P^s ne postoji. Ako se dodje do nenegativnog vektora $u^s = (x^s, y^s)$, on je optimalno rješenje za problem P^s. Takvo rješenje je ujedno i moguće rješenje problema P, ali ne mora biti i njegovo optimalno rješenje. Postupak traženja optimalnog rješenja za problem P počinje u tom slučaju od nekog rješenja $u^p (p < s)$, izabranog u skladu s odredjenim pravilom, s time da druga pravila odredjuju izbor vektora koji se uvodi u bazu. Taj postupak traje tako dugo dok se ne dobije drugo moguće rješenje u^t sa svojstvom da je $z_t < z_s$ (z_p je oznaka za vrijednost funkcije cilja z za vektor u^p, $p = 0,1,\ldots$) ili se evidentno ustanovi nedopustivost takvog rješenja. Niz u^q, $q = 0,1,\ldots$, konvergira prema optimalnom rješenju.

Taj postupak mogao bi se nazvati i pseudo-dualnim algoritmom, jer, kao i u dualnoj simpleks metodi, starta s dualnim mogućim rješenjem i postepeno se približava primal - mogućem području, čuvajući cijelo vrijeme svojstvo dualnog mogućeg rješenja. Medjutim, stvarna dualna simpleks iteracija se ne provodi, ne upotrebljava se ni dualni simpleks kriterij za izbor vektora koji ulazi u bazu, a takodjer ni jedan od vektora e_i $i = 1,\ldots,m$ nije eliminiran iz baze u smislu da je nadomješten nekim drugim vektorom. Do promjene baze dolazi dodavanjem novog jediničnog retka ili jediničnog stupca, odnosno njihovim ispuštanjem i, što je najvažnije, koeficijenti matrice A ostaju nepromijenjeni. Kako novi jedinični reci i jedinični stupci uvedeni u svakoj iteraciji ne igraju nikakvu ulogu u slijedećoj iteraciji, nije potrebno ni jedne ni druge sačuvati ni izričito ispisati.

Djelovanje aditivnog algoritma može se lakše shvatiti ako se usporedi s rješavanjem zadanog problema putem testiranja skupa svih mogućih rješenja. Kako varijable, koje se javljaju u problemu P, mogu poprimiti samo dvije vrijednosti - 0 ili 1, skup $U = \{u\}$ svih rješenja od (2) i (3) je konačan i broj elemenata tog skupa je 2^n, gdje je n broj varijabli koje se javljaju u problemu P. Postupak promatranja svih 2^n mogućih varijacija s ponavljanjem od dva elementa prikazan je na slici 1 za problem u kojem se javlja 5 varijabli.[5)]



Slika 1.

5) Na slici 1 može se pratiti tok rješavanja primjera 1 koji je dan nakon ovog prikaza aditivnog algoritma. Grane na slici, koje su izvučene debljom linijom, upravo ilustriraju tok rješavanja primjera 1.

Shema na slici 1 ima pet nivoa. Točci iz koje se počinje crtati shema pridruženo je rješenje za koje sve varijable imaju vrijednost nula. Dolaskom na prvi nivo moguća su dva rješenja koja se razlikuju u vrijednosti varijable x_1. Točka u koju se dodje iz početne točke spuštanjem po desnoj grani pridružena je rješenju

$$x_j = \begin{cases} 1 & j=1 \\ 0 & j=2,3,4,5 \end{cases}$$

a točka u koju se dodje spuštanjem po lijevoj grani pridružena je rješenju

$$x_j = 0 \quad j=1,2,3,4,5.$$

Isti princip formiranja novog skupa rješenja zadržava se i kod prelaska na idući nivo. Sve točke u koje se dolazi spuštanjem s gornjeg nivoa po lijevoj grani pridružene su istom rješenju kome je pridružena i ishodišna točka s gornjeg nivoa, a ako se u neku točku dodje spuštanjem po desnoj grani, to znači da se rješenje koje joj je pridruženo razlikuje od rješenja pridruženog ishodišnoj točki s neposredno prethodnog gornjeg nivoa po tome što varijabla s rednim brojem nivoa dobiva vrijednost 1. Za zadani skup od pet varijabli taj postupak provodi se do popunjavanja petog nivoa na kojem se nalazi skup od 32 točke - svaka je pridružena jednoj varijaciji s ponavljanjem petog razreda od dva elementa. Pomoću sheme sa slike 1 može se objasniti djelovanje aditivnog algoritma. To je u stvari skup pravila u skladu s kojima se postiže optimalno rješenje (ako takvo postoji), ali slijedeći samo neke grane na shemi, dok se druge grane zanemaruju. Startajući iz situacije u kojoj je vrijednost svih n varijabli 0, algoritam sistematski dodjeljuje nekim varijablama vrijednost 1, tako da nakon provjere malog broja od skupa svih mogućih varijacija vrijednosti varijabli, pronalazi optimalno rješenje ili se pokaže da takvo rješenje ne postoji.

U stvari to se postiže kroz skup pravila koja se primjenjuju u svakoj iteraciji, a grubo se postupak može podijeliti u dva dijela:

a) odredjuje se podskup varijabli koje kandidiraju za dodjelu vrijednosti 1
b) izmedju tih varijabli izaberu se one kojima se dodjeljuje vrijednost 1.

Nakon odredjenog broja iteracija postaje jasno da je optimalno rješenje postignuto ili da u optimalnom rješenju nemaju vrijednost 1 sve one varijable kojima je ta vrijednost dodijeljena tokom postupka.

Postupak se tada zaustavlja i ponovo se kreće iz prethodne etape. Drugim riječima, pravila algoritma omogućuju da se identificiraju one grane stabla rješenja koje ne vode k rješenju boljem od već postignutog, i koje se zbog toga mogu napustiti. Efikasnost algoritma u znatnoj mjeri ovisi o efikasnosti postupka kojim se utvrdjuje da neko rješenje nije bolje od već postignutog rješenja, odnosno o reduciranju što većeg broja grana koje nije potrebno slijediti.

Neke definicije i oznake

Promatra se problem P. Budući da svako ograničenje skupa (2) sadrži točno jednu komponentu vektora y, rješenje $u^p=(x^p,y^p)$ je jedinstveno odredjeno skupom

$$J_p = \{j \mid j\varepsilon N,\ x_j^p=1\}$$

zbog

$$x_j^p = \begin{cases} 1 & j \varepsilon J_p \\ 0 & j \varepsilon N \setminus J_p \end{cases} \tag{5}$$

$$y_i^p = b_i - \sum_{j\varepsilon J_p} a_{ij} \qquad i\varepsilon M \tag{6}$$

Kao što je pokazano, aditivni algoritam generira niz rješenja.

S-ti član tog niza se označi s

$$u^s = u(j_1,\ldots..,j_r)=(x^s,y^s) \tag{7}$$

gdje je

$$\{j_1,\ldots..,j_r\} = J_s =\{j \mid j\varepsilon N,\ x_j^s=1\} \tag{8}$$

dok će z_s biti oznaka za vrijednost forme (1) za u^s. Niz ovih rješenja počinje s u^o i za to rješenje je $J^o = \emptyset$, $x^o = 0$, $y^o=b$ i $z_o=0$. Naravno, $J_o \subset J_p$ za svaki $p \neq 0$. Skup vrijednosti koje poprima funkcija cilja za moguća rješenja dobivena do s-te iteracije označi se sa

$$Z_s = \{z_p \mid p < s,\ u^p > 0\} \tag{9}$$

Ako taj skup nije prazan, njegov najmanji element će se zvati plafon za u^s. Ako pak je taj skup prazan, ulogu plafona preuzeti će ∞. Plafon za u^s bit će označen sa

$$z^{*s} = \begin{cases} \infty \text{ ako je } Z_s = \emptyset \\ \min\limits_{Z_s} z_p \text{ ako je } Z_s \neq \emptyset \end{cases} \tag{10}$$

Za svaku iteraciju s+1 novi vektor koji se uvodi u bazu bit će izabran iz podskupa skupa $\{a_j \mid j\varepsilon N\}$, koji se zove skup poboljšavajućih vektora za rješenje u^s. Odgovarajući skup indeksa j označi se s N_s (naravno, $N_s \subseteq N$), a u daljnjem tekstu će se definirati preciznije.

Nadalje treba definirati neke vrijednosti koje će se koristiti kao kriterij za izbor vektora koji se uvodi u bazu. Za svako rješenje u^s i svaki $j\varepsilon N_s$ definiraju se vrijednosti

$$v_j^s = \begin{cases} \sum\limits_{i\varepsilon M_j^{s-}} (y_i^s - a_{ij}) & j\varepsilon N_s;\ M_j^{s-} \neq \emptyset \\ 0 & j\varepsilon N_s;\ M_j^{s-} = \emptyset \end{cases} \tag{11}$$

gdje je $\quad M_j^{s-} = \{i \mid y_i^s - a_{ij} < 0\}$ (12)

Značenje tih vrijednosti je očito; v_j^s je suma negativnih komponenata vektora rješenja u^{s+1}, koji se dobije iz vektora rješenja u^s povećanjem skupa J_s na $J_{s+1} = J_s \cup \{j'\}$. Kao što je spomenuto, vrijednost v_j^s služi za izbor novog vektora koji se uvodi u bazu, ako je ustanovljeno da je to efikasan kriterij, potrebno je istaći da je on empirički i da nije bitan za funkcioniranje aditivnog algoritma - on može dobro raditi i primjenom nekog drugog kriterija. Izbor kriterija za uvodjenje novog vektora u bazu utiče na efikasnot algoritma, ali nema utjecaj na njegovu konačnost. U skladu s aditivnim algoritmom vrijednosti v_j^k pridodate svakom rješenju u^k postepeno se brišu u idućim iteracijama u skladu s ulogom koju igraju.

Skup takvih j, za koje su vrijednosti v_j^k pridodate rješenju u^k brisane prije nego je dobiveno rješenje u^s, označi se s $C_k^s (k<s)$. ($C_k^k = \emptyset$ po definiciji).

Skup indeksa j za koje su vrijednosti v_j^p pridodate svakom rješenju u^p sa svojstvom $p < s$ i $J_p \subset J_s$, brisane prije postizanja rješenja u^s označi se s

$$C^s = \bigcup_{p|J_p \subset J_s} C_p^s \tag{13}$$

Dalje se za rješenje u^s definira skup takvih $j \varepsilon (N-C^s)$ da, ako je a_j uvedeno u bazu tako da je postignuto

$$J_{s+1} = J_s \cup \{j\}$$

vrijednost funkcije cilja postigne plafon za u^s.

Oznaka za taj skup je

$$D_s = \{j | j\varepsilon(N-C^s), \quad c_j > z^{*(s)} - z_s\} \tag{14}$$

Iz skupa indeksa $j \varepsilon |N-(C^s \cup D_s)|$ izdvojit će se u poseban skup oni j koji imaju svojstvo da, ako se pripadni a_j uvede u bazu tako da je

$$J_{s+1} = J_s \cup \{j\},$$

nenegativan y_i^s poraste.

Skup takvih j označava se s

$$E_s = \{j | j\varepsilon | N - (C^s \cup D_s)|, \; y_i^s < 0 \rightarrow a_{ij} > 0\} \tag{15}$$

Sada se može dati prava definicija skupa poboljšavajućih vektora za rješenje u^s. To je skup takvih a_j, za koje j pripada skupu

$$N_s = N - (C^s \cup D_s \cup E_s) \tag{16}$$

Očigledno, $N_s = \emptyset$ za svako moguće rješenje u^s. Slično s D_s definira se par rješenja u^s i u^k $(k < s)$ skup onih $j \varepsilon (N_k - C_k^s)$ za koje, ako je a_j uveden u bazu tako da je postignuto $J_{s+1} = J_k \cup \{j\}$, z_{s+1} postiže plafon za u^s

$$D_k^s = \{j | j\varepsilon(N_k - C_k^s), \; c_j > z^{*(s)} - z_k\} \tag{17}$$

Konačno, za dani par rješenja u^s i u^k takav da je

$$u_s=u(j_1,\ldots,j_r),\ u^k=u(j_1,\ldots,j_{r-h})(1\leq h\leq r;\ J_k\subset J_s)$$

definira se skup poboljšavajućih vektora za rješenje u^k, preostalih nakon iteracije s. To je skup takvih a_j za koje j pripada skupu

$$N_k^s = N_k - (C_k^s \cup D_k^s) \qquad (18)$$

Skupovi definirani s (16) i (18) igraju centralnu ulogu u ovom algoritmu. Čim se dostigne neko rješenje u^s, za daljnje uvodjenje u bazu dolaze u obzir samo poboljšavajući vektori za to rješenje. Ako se ustanovi da je skup poboljšavajućih vektora za rješenje u^s prazan, to se interpretira kao "stop signal", što znači da ne postoji moguće rješenje u^t takvo da je $J_s\subset J_t$ i $z_t < z^{*(s)}$. U takvom slučaju ponavlja se postupak za neko prethodno rješenje u^k, identificirano u skladu s odgovarajućim pravilima, i za svako takvo rješenje u^k samo skup poboljšavajućih vektora preostalih nakon iteracije s dolazi u obzir za uvodjenje u bazu.

3. PRIKAZ ADITIVNOG ALGORITMA

U daljnjem tekstu bit će aditivni algoritam prikazan po koracima. Starta se s dualnim mogućim rješenjem u^o, za koje su

$$x_o = 0,\quad y^o = b \ \text{ i } \ z_o = 0 \qquad (19)$$

Neka je nakon s-te iteracije dobiveno rješenje $u^s=u(j_1,\ldots,j_r)$ za koje je

$$x_j^s = \begin{cases} 1 & j \in J_s \\ 0 & j \in (N-J_s) \end{cases} \qquad (20)$$

i

$$z_s = \sum_{j\in J_s} c_j \qquad (21)$$

Tada se dalje postupa na slijedeći način:

Korak 1 : Ispitivanje y_i^s $(i\in M)$.

1a. Ako je $y_i^s > 0$ $(i\in M)$, skup $z_s=z^{*(s)}$. Skup D_k^s je definiran sa (17) za svaki $k<s$, brišu se vrijednosti v_j^k za $j\in D_k^s$, $k<s$ i prelazi na korak 5. Ako do ovog slučaja dodje za u^o, tada je u^o optimalno rješenje i algoritam je okončan.

1b. Ako postoji i_1 takav da je $y^s_{i_1} < 0$, prelazi se na

Korak 2 : Identificiranje poboljšavajućeg vektora za rješenje u^s formiranjem skupa N_s kao što je to definirano sa (16).

2a. Ako je $N_s = \emptyset$, tada ne postoji poboljšavajući vektor za u^s i^s prelazi se na korak 5.

2b. Ako je $N_s \neq \emptyset$, prelazi se na

Korak 3 : Ispitivanje relacije

$$\sum_{j \varepsilon N_s} a^-_{ij} < y^s_i \qquad (i / \; y^s_i < 0) \tag{22}$$

gdje su a^-_{ij} negativni elementi matrice A.

3a. Ako postoji takav i_1 M za koji nejednakost (22) ne vrijedi, prelazi se na korak 5.

3b. Ako su sve relacije (22) zadovoljene kao stroge nejednakosti, treba izračunati vrijednosti v^s_j prema definicionim izrazima (11) i (12) za svaki $j \varepsilon N_s$ i izabrati onaj j_{s+1} za koji je

$$v^s_{j+1} = \max_{j \varepsilon N_s} v^s_j \tag{23}$$ [6]

izbrisati $v^s_{j_{s+1}}$ i preći na korak 8.

3c. Ako sve relacije (22) vrijede, i postoji podskup M^s od M takav da su relacije (22) zadovoljene kao jednakosti za $i \varepsilon M^s$, prelazi se na

Korak 4 : Ispitivanje relacije

$$\sum_{j \varepsilon F_s} c_j < z^{x(s)} - z_s \tag{24}$$

gdje je ^sF_s skup takvih $j \varepsilon N_s$, za koje je $a_{ij} < 0$ za najmanje jedan $i \varepsilon M^s$.

6) Ako jednakosti (23) i (28) vrijede za više od jednog $j = j_{s+1}$, j_{s+1} se bira tako da je $c_{j_{s+1}} = \min_{j \varepsilon J_{max}} c_j$ (gdje je J_{max} skup onih j za koje vrijedi taj slučaj). Ako ta relacija takodjer vrijedi za više indeksa j, tada se bira bilo koji od njih kao j_{s+1}.

4a. Ako vrijedi (24), briše se v_j^s za sve $j \in F_s$ (bez izračunavanja njihovih numeričkih vrijednosti). Skup $J_{s+1} = J_s \cup F_s$ izračunaju se vrijednosti funkcije cilja

$$z_{s+1} = z_s + \sum_{j \in F_s} c_j \tag{25}$$

i oslabljenih varijabli

$$y_i^{s+1} = y_i^s - \sum_{j \in F_s} a_{ij} \quad (i \in M) \tag{26}$$

za novo rješenje u^{s+1}, i prelazi se na novu iteraciju (starta se ponovo s korakom 1).

4b. Ako (24) ne stoji, briše se v_j^s za svaki $j \in N_s$ (bez izračunavanja numeričkih vrijednosti), i prelazi se na

Korak 5 : Identificiranje poboljšavajućih vektora za rješenje (u^k($k/J_k \subset J_s$) preostalih nakon interacije s, ispitivanje skupova N_k^s ($k/J_k \subset J_s$) definiranih s (18) za opadajući redoslijed brojeva k, tako dugo dok se ne pronadje k_1 takav da vrijedi $J_{k_1} \subset J_s$ i $N_{k_1}^s \neq \emptyset$ ili se ustanovi da je skup N_k^s prazan za svaki k sa svojstvom $J_k \subset J_s$.

5a. Ako je $N_k^s = \emptyset$ za sve k takve da je $J_k \subset J_s$, tada ne postoje poboljšavajući vektori ni za jedan u^k($k/J_k \subset J_s$) i algoritam je okončan. U tom slučaju, ako je $Z_s = \emptyset$, P nema moguće rješenje. Ako pak je $Z_s = \emptyset$, tada je u^q, za koji je $z_q = z^{*(s)}$ optimalno rješenje i $z^{*(s)}$ je minimum što ga funkcija cilja postiže za to rješenje.

5b. Ako je $N_k^s \neq \emptyset$ za $k=k_1$, $J_k \subset J_s$, preći na

Korak 6: Ispitivanje relacije

$$\sum_{j \in N_k^s} a_{ij}^- \leq y_i^k \qquad (i/\ y_i^k < 0) \tag{27}$$

za $k=k_1$.

6a. Ako i jedna od relacija (27) nije ispunjena za $k=k_1$, brisati $v_j^{k_1}$ za svaki $j \varepsilon N_{k_1}^s$ i ponoviti korak 5 za $k< k_1$, pišući k_2 umjesto k_1 u koracima 5 i 6. Uvijek kada se korak 5 ponavlja za $k< k_\alpha$, treba pisati $k_{\alpha+1}$ umjesto k u koracima 5 i 6. Ako relacije (27) nisu zadovoljene za svaki k takav da je $N_k^s \neq \emptyset$, algoritam je okončan, s istim zaključcima kao u 5a.

6b. Ako su za $k=k_\beta$ sve relacije (27) zadovoljene kao stroge nejednakosti, treba izabrati j_{s+1} takav da je

$$v_{j_{s+1}}^{k} = \max_{j \varepsilon N_{k_\beta}} v_j^{k_\beta} \tag{28}$$

poništiti $v_{j_{s+1}}^{k_\beta}$ i preći na korak 8.

6c. Ako su za $k = k_\beta$ sve relacije (27) zadovoljene, i postoji podskup $M_{k_\beta}^s$ od M takav da su relacije (27) zadovoljene kao jednadžbe za $i \varepsilon M_{k_\beta}^s$, preći na

Korak 7: Ispitivanje relacije

$$\sum_{j \,\varepsilon\, F_k^s} c_j < z^{*(s)} - z_{k_\beta} \tag{29}$$

gdje je $F_{k_\beta}^s$ skup takvih $j \varepsilon N_{k_\beta}^s$ za koje je $a_{ij}<0$ za najmanje jedan $i \varepsilon M_{k_\beta}^s$.

7a. Ako vrijedi (29), poništit $v_j^{k_\beta}$ za svaki $j \varepsilon F_{k_\beta}^s$. Skup $J_{s+1} = J_{k_\beta} \cup F_{k_\beta}^s$. Izračunati vrijednost funkcije cilja

$$z_{s+1} = z_{k_\beta} + \sum_{j \varepsilon F_{k_\beta}^s} c_j \tag{30}$$

i oslabljenih varijabli

$$y_i^{s+1} = y_i^{k_\beta} - \sum_{j \in F^s_{k_\beta}} a_{ij} \quad (i \in M) \qquad (31)$$

za novo rješenje u^{s+1} i preći na novu iteraciju (startati ponovo s korakom 1).

7b. Ako ne vrijedi (29), poništiti vrijednosti $v_j^{k_\beta}$ za svaki $j \in N^s_{k_\beta}$ i ponoviti korak 5 za svaki $k < k_\beta$. Ako takav $k < k_\beta$ ne postoji, npr. ako je $k_\beta = 0$, algoritam je okončan s istim zaključcima kao u 5a.

Korak 8: Skup $J_{s+1} = J_s \cup \{j_{s+1}\}$. Izračunati vrijednosti funkcije cilja i oslabljenih varijabli za novo rješenje u^{s+1}, u skladu s formulom

$$z_{s+1} = z_p + c_{j_{s+1}} = \sum_{j \in J_{s+1}} c_j \qquad (32)$$

$$\text{i} \quad y_i^{s+1} = y_i^p - a_{ij_{s+1}} = b_i - \sum_{j \in J_{s+1}} a_{ij} \quad (i \in M) \qquad (33)$$

gdje je p definiran s posljednjom izbrisanom vrijednošću v_j^p. Preći na iduću iteraciju (npr.startati ponovo s korakom 1).

Algoritam prestaje kada je dobiveno rješenje u^t za koje je

(I) dobivena situacija 5a ili
(II) dobivena situacija 6a i (27) ne vrijedi ni za jedan k takav da je $N^s_k \neq \emptyset$ ili
(III) dobivena je situacija 7b i $k_\beta = 0$.

Treba napomenuti da aditivni algoritam opisuje dobivanje jednog optimalnog rješenja (ako takvo postoji). No, ako postoji više optimalnih rješenja za dani problem, ona se mogu postići uz modifikaciju izloženog algoritma u kojoj se u (14) i (17) stavlja znak $>$ umjesto $>$, a u (24) i (29) $\leqslant$ umjesto $<$.

Nadalje, moguće je reducirati broj rješenja koja se direktno ispituju tokom izvodjenja algoritma. Cijena za to je nešto složenija numerička procedura u svakoj iteraciji, a koja se zasniva na promjeni algoritamskih koraka 3b, 3c, 6b, 6c kako slijedi;

3b. Ako vrijedi (22) ispitati relacije

$$\sum_{j\varepsilon N_s} \bar{a}_{ij} - \bar{a}_{ij_*} \leq y_i^s \quad (i/\ y_i^s < 0) \qquad (22a)$$

gdje je $\bar{y}_{ij_*} = \max\limits_{j\varepsilon N_s} \bar{a}_{ij}$. Ako sve relacije (22a) vrijede, izračunati vrijednosti v_j^s za svaki $j\ \varepsilon\ N_s$, izabrati j_{s+1} takav da je $v_{j_{s+1}}^s = \max\limits_{j\varepsilon N_s} v_j^s$

Poništiti $v_{j_{s+1}}^s$ i preći na korak 8.

3c. Ako vrijede relacije (22), i postoji podskup M^s od M takav da relacije (22a) ne vrijede za $i\ \varepsilon\ M^s$, preći na korak 4.

Odgovarajuće promjene treba u cijelosti učiniti i u koracima 6b i 6c.

4. PRIMJERI

U ovom dijelu rada prikazano je rješavanje dvaju primjera iznesenom metodom. Prvi problem, problem minimuma, je u potpunosti riješen uz navodjenje svih algoritamskih koraka. Za drugi problem navedeni su samo algoritamski koraci kroz koje se prolazi tokom rješavanja, a čitalac može lako pratiti postupak rješavanja kombiniranjem ovog s prethodnim zadatkom.

Za problem maksimuma nije dan primjer jer se on rješava jednostavnim svodjenjem na problem minimuma, kako je prikazano na početku iznošenja algoritma.

Primjer 1: Odrediti minimalnu vrijednost funkcije

$$z' = -2x_1' + 3x_2' + x_3' + 5x_4' - 4x_5'$$

uz uvjete

$$\begin{aligned} x_1' + 2x_2' - 4x_3' + 4x_4' + 3x_5' &\geq 4 \\ 2x_1' - 3x_2' - x_3' + 5x_4' + 6x_5' &\leq 8 \\ 4x_1' + x_2' \qquad - 3x_4' - 2x_5' &\leq 0 \end{aligned}$$

$$x_j' = 0 \text{ ili } 1 \quad \text{za } j = 1,\ldots,5$$

Da bi se ovaj problem transformirao u oblik na koji se može primijeniti aditivni algoritam (oblik koji odgovara problemu P iz prethodnog prikaza aditivnog algoritma), treba uraditi slijedeće:

- prvu nejednadžbu pomnožiti s -1 da bi se svela na oblik
- izvršiti zamjenu varijabli kako slijedi

$$x_j = \begin{cases} x_j' & \text{za } j = 2,4 \\ 1 - x_j' & \text{za } j = 1, 3, 5 \end{cases}$$

- uvesti dopunske varijable

Transformirani problem u tom slučaju glasi:

Odrediti minimum funkcije

$$z = 2x_1 + 3x_2 + x_3 + 5x_4 + 4x_5$$

uz uvjete

$$\begin{aligned} x_1 - 2x_2 - 4x_3 - 4x_4 + 3x_5 + y_1 \qquad\qquad &= -4 \\ -2x_1 - 3x_2 + x_3 + 5x_4 - 6x_5 \qquad + y_2 \qquad &= 1 \\ -4x_1 - x_2 \qquad - 3x_4 + 2x_5 \qquad\qquad + y_3 &= -2 \end{aligned}$$

$$x_j = 0 \text{ ili } 1 \quad \text{za } j = 1,\ldots,5$$

Rješavanje ovog zadatka može se pratiti kroz tabelu 1. U daljnjem tekstu bit će navedeni algoritamski koraci koji se poduzimaju u svakoj iteraciji, uz prikaz izračunavanja pojedinih veličina i formiranja skupova indeksa koji su potrebni za rad algoritma.

Broj reda	s	J_s	z_s	N_s		i: 1	2	3	v_j^s	C^s	D_s	E_s	F_s
1	0	Ø	0		y_i^o	-4	1	-2		Ø	Ø		
2					$\sum_{j\varepsilon N_o} a^-_{ij}$	-10		-7					
3				1	$y_i^o - a_{i1}$	-5			-5^7				
4				2	$y_i^o - a_{i2}$	-2		-3	-5^8				
5				3	$y_i^o - a_{i3}$			-2	-2^1				
6				4	$y_i^o - a_{i4}$		-4		-4^6				
7	1	3	1		y_i^1	0	0	-2		3	Ø	2,5	
8					$\sum_{j\varepsilon N_1} a^-_{ij}$			-7					
9				1	$y_i^1 - a_{i1}$	-1			-1^2				
10				4	$y_i^1 - a_{i4}$		-5		-5^4				
11	2	3,1	3		y_i^2	-1	2	2		3,1	Ø	5	
12					$\sum_{j\varepsilon N_2} a^-_{ij}$	-6							
13				2	$y_i^2 - a_{i2}$				0^3				
14				4	$y_i^2 - a_{i4}$		-3		-3^5				
15	3	3,1,2	6		y_i^3	1	5	1					
16					$\sum_{j\varepsilon N_1^3} a^-_{ij}$			0					
17					$\sum_{j\varepsilon N_o^3} a^-_{ij}$	-6		-7					
18	4	3,1,2,4	11		y_i^4	5	0	4					
19					$\sum_{j\varepsilon N_o^4} a^-_{ij}$	-2							

Tabela 1

Algoritam počinje s odredjivanjem vrijednosti $s = 0$, $J_o = \emptyset$, $z_o = 0$ za početno rješenje u^o (0, b). Za ovo početno rješenje su vrijednosti oslabljenih varijabli

$$y_1^o = -4,\ y_2^o = 1,\ y_3^o = -2$$

I Iteracija:

1. korak: $y_i^o < 0$ za $i = 1, 3$, prema 1b prelazi se na
2. korak: Traženje poboljšavajućeg vektora za rješenje u^o, formiranjem skupa

$N_o = N - (C^o \cup D_o \cup E_o)$

$C_o = \emptyset$, $D_o = \emptyset$ (zbog $Z_o = \emptyset \rightarrow z^{x(o)} = \infty \rightarrow$ ne postoji j takav da bi vrijedilo $c_j \geq z^{x(o)} - z_o$)

$E_o = \{5\}$ (vidi 1. red tabele 1)

$N_o = \{1,2,3,4,5\} - \{5\} = \{1, 2, 3, 4\}$

Zbog $N_o \neq \emptyset$ ovo je situacija 2b, pa se prelazi na

3. korak: Ispitivanje relacije (22) za $i = 1,3$ (2.red u tabeli 1)

$$\sum_{j\in N_o} a_{1j}^- = a_{12}^- + a_{13}^- + a_{14}^- = -2 -4 -4 = -10 < y_1^o = -4$$

$$\sum_{j\in N_o} a_{31}^- = a_{34}^- = -4 -3 = -7 \quad y_3^o = -2$$

Kako su nejednakosti (22) strogo zadovoljene, ovo je slučaj 3b, pa se izračunavaju vrijednosti v_j^o definirane s (11) i (12) za $j\in N_o$ (to je prikazano u 3., 4., 5. i 6. redu tabele 1):

$$v_1^o = \sum_{i\in M_1^{o-}} (y_i^o - a_{i1}) = -4 -1 = -5$$

$$v_2^o = \sum_{i\in M_2^{o-}} (y_i^o - a_{i2}) = (-4 + 2) + (-2 - 1) = -5$$

$$v_3^o = \sum_{i\in M_3^{o-}} (y_i^o - a_{i3}) = -2$$

$$v_4^o = \sum_{i\varepsilon M_4^{o-}} (y_i^o - a_{i4}) = 1 - 5 = -4$$

Vidi se da je $\max_{j\varepsilon N_o} v_j^o = v_3^o$, pa se prema 3b poništava v_3^o (kako je algoritam važno kojim se redom poništavaju vrijednosti v_j^s, to se redoslijed poništavanja tih vrijednosti bilježi dodavanjem numeracije; npr. uz $v_3^o = -2$ doda se 1 (-2^1) - vidi 5. red tabele 1) i prelazi na

8. korak: $J_1 = J_o \cup \{3\} = \emptyset \cup \{3\} = \{3\}$

$z_1 = z_o + c_3 = 3$

$y_1^1 = y_1^o - a_{13} = -4 + 4 = 0$

$y_2^1 = y_2^o - a_{23} = 1 - 1 = 0$

$y_3^1 = y_3^o - a_{33} = -2 - 0 = -2$

(sve ove vrijednosti upisane su u sedmi red tab.1)

Prelazi se na iduću iteraciju.

II Iteracija:

1. korak: $y_3^1 < 0$, prema 1b prelazi se na
2. korak: $N_1 = N - (C^1 \cup D_1 \cup E_1)$

   $C_1 = \{3\}$, $D_1 = \emptyset$, $E_1 = \{2,5\}$, (vidi 7.red tab.1)

   $N_1 = \{1,2,3,4,5\} - \{\{3\} \cup \{2,3,5\}\} = \{1,4\}$

   Zbog $N_1 \neq \emptyset$, prema 2b prelazi se na
3. korak: Ispitivanje relacije (22) za i=3 (prikazano u 8. redu tabele 1)

   $$\sum_{j\varepsilon N_1} a_{3j}^- = a_{31}^- + a_{34}^- = -4 -3 = -7 < y_3^1 = -2$$

   Kako je ovo slučaj 3b, treba izračunati vrijednosti v_j^1 za j = 1,4 (vidi 9.i 10.red tabele 1):

   $$v_1^1 = \sum_{i\varepsilon M_1^{1-}} (y_i^1 - a_{i1}) = 0 - 1 = -1$$

   $$v_4^1 = \sum_{i\varepsilon M_4^{1-}} (y_i^1 - a_{i4}) = 0 - 5 = -5$$

Kako je $v_1^1 = \max_{j \in N_1} v_j^1$, briše se v_1^1 i prelazi na

8. korak: $J_2 = J_1 \cup \{1\} = \{3,1\}$

$z_2 = z_1 + c_1 = 1 + 2 = 3$

$y_1^2 = y_1^1 - a_{11} = 0 - 1 = -1$

$y_2^2 = y_2^1 - a_{21} = 0 + 2 = 2$

$y_3^2 = y_3^1 - a_{31} = -2 + 4 = 2$

(vidi 11. red tabele 1)

Prelazi se na novu iteraciju.

III Iteracija:

1. korak: $y_1^2 < 0$, pa se prema 1b prelazi na

2. korak: $N_2 = N - (C^2 \cup D_2 \cup E_2)$

$C^2 = \{3,1\}$, $D_2 = \emptyset$, $E_2 = \{5\}$ (vidi 11.red tabele 1)

$N^2 = N - (c^2 \cup D_2 \cup E_2) = \{1,2,3,4,5\} - \{\{3,1\} \cup \{5\}\} = \{2,4\}$

Zbog $N_2 \neq \emptyset$ prelazi se, prema 2b, na

3. korak: Ispitivanje relacije (22) za i=1 (prikazano u 12. redu tabele 1)

$$\sum_{j \in N_2} a_{1j}^- = a_{12}^- + a_{14}^- = -2 - 4 = -6 < y_1^2 = -1$$

Očito se radi o slučaju 3b, pa treba izračunati vrijednosti v_j^2 za j = 2,4 (vidi 13., 14., 15. red u tabeli 1)

$$v_2^2 = \sum_{j \in M_2^{2-}} (y_i^2 - a_{i2}) = 0$$

$$v_4^2 = \sum_{j \in M_4^{2-}} (y_i^2 - a_{i4}) = 2 - 5 = -3$$

Vidi se da je $v_2^2 = \max_{j \in N_2} v_j^2$, pa se briše v_2^2 i prelazi na

8. korak: $J_3 = J_2 \cup \{2\} = \{3,1,2\}$

$z_3 = z_2 + c_2 = 3 + 3 = 6$

$y_1^3 = y_1^2 - a_{12} = -1 + 2 = 1$

$y_2^3 = y_2^2 - a_{22} = 2 + 3 = 5$

$y_3^3 = y_3^2 - a_{32} = 5 - 1 = 4$

(vidi 15. red tabele 1)

Prelazi se na

IV Iteraciju:

1. korak: Kako je $y_i^3 > 0$ za sve i, ovo je slučaj 1a.

$Z_3 = \{z_3\}$, $z^{*(3)} = z_3 = 6$

Treba formirati skupove D_o^3, D_1^3 i D_2^3:

$D_o^3 = \{j/\ j\varepsilon(N_o - C_o^3),\ c_j \geqslant (6-0)\} = \emptyset$ (jer je $c_j < 6\ \forall\ j$)

$D_1^3 = \{j/\ j\varepsilon(N_1 - C_1^3),\ c_j \geqslant (6-1)\}$

$N_1 = \{1,4\}$, $C_1^3 = \{1\} \Rightarrow D_1^3 = \{4\}$

$D_2^3 = \{j/\ j\varepsilon(N_2 - C_2^3),\ c_j > (6-3)\}$

$N_2 = \{2,4\}$, $C_2^3 = \{2\} \Rightarrow D_2^3 = \{4\}$

Dalje se, prema 1a, brišu v_4^1 (zbog $D_1^3 = \{4\}$) i v_4^2 (zbog $D_2^3 = \{4\}$ i prelazi se na

5. korak: Ispituje se da li postoji poboljšavajući vektor za rješenje u^3, ispitivanjem skupova N_k^3 (definiranih s (18)) za k = 2,1,0.

$N_2^3 = N_2 - (C_2^3 \cup D_2^3)$

$N_2 = \{2,4\}$, $C_2^3 = \{2\}$, $D_2^3 = \{4\} \rightarrow N_2^3 = \emptyset$

$N_1^3 = N_1 - (C_1^3 \cup D_1^3)$

$N_1 = \{1,2,3,4\}$, $C_1^3 = \{1\}$, $D_1^3 = \{4\} \rightarrow$ $N_1^3 = \{2,3\}$

Kako je $N_1^3 \neq \emptyset$, ovo je situacija 5b, pa se prelazi na

6\. korak: Ispituje se relacija (27) za k = 1, i=3 (jer je $y_3^1 < 0$) (vidi 16. red tabele 1)

$$\sum_{j\in N_1^3} a_{3j}^- = 0 \not< y_3^1 = -2$$

Prema tome, ovo je slučaj 6a. Brišu se v_2^1 i v_4^1 (već su brisani) i ponavlja se 5. korak za k=5.

5\. korak: $N_o^3 = N_o - (C_o^3 \cup D_o^3)$

$N_o = \{1,2,3,4\}$, $C_o^3 = \{3\}$, $D_o^3 = \emptyset \rightarrow N_o^3 = \{1,2,4\}$

Zbog $N_o^3 \neq \emptyset$, prema 5a, prelazi se na

6\. korak: Ispituje se relacija (27) za k = 0, i=1,3

$$\sum_{j\in N_o^3} a_{1j}^- = -2 -4 = -6 < y_1^o = -4$$

$$\sum_{j\in N_o^3} a_{3j}^- = -4 -3 = -7 < y_3^o = -2$$

Ovo je slučaj 6b. Kako je $\max\limits_{j\in N_o} v_j^o = v_4^o$ poništi se v_4^o i prelazi na

8\. korak: $J_4 = J_3 \cup \{4\} = \{3, 1, 2, 4\}$

$z_4 = z_3 + c_4 = 6 + 5 = 11$

$y_1^4 = y_1^3 - a_{14} = 1 + 4 = 5$

$y_2^4 = y_2^3 - a_{24} = 5 - 5 = 0$

$y_3^4 = y_3^3 - a_{34} = 1 + 3 = 4$

(vidi 18. red u tabeli 1)

prelazi se na

V Iteraciju:

1\. korak: Zbog $y_i^4 \geq 0$ (i=1,2,3) ovo je slučaj 1a.

$Z_4 = \{6,11\}$ $z^{*(4)} = 6$

Formiraju se skupovi D_k^4 za k = 0, 1, 2, 3;

$D_o^4 = \{j/\ j\varepsilon(N_o - C_o^4),\ c_j \geqslant 6\} = \emptyset$

$D_1^4 = \{j/\ j\varepsilon(N_1 - C_1^4),\ c_j \geqslant 5\}$

$N_1 = \{2,\ 4\},\ C_1^4 = \{1,\ 4\} \rightarrow D_1^4 = \emptyset$

$D_2^4 = \{j/j \in (N_2 - C_2^4),\ c_j \geqslant 3\}$

$N_2 = \{2,\ 4\},\ C_2^4 = \{2,\ 4\} \rightarrow D_2^4 = \emptyset$

$D_3^4 = \emptyset$ (zbog $N_3 = \emptyset$)

Prelazi se na

5. korak: $N_3^4 = N_3 - (C_3^4 \cup D_3^4) = \emptyset$

$N_2^4 = N_2 - (C_2^4 \cup D_2^4)$

$N_2 = \{2,\ 4\},\ C_2^4 = \{2,\ 4\} \rightarrow N_2^4 = \emptyset$

$N_1^4 = N_1\ (C_1^4 \cup D_1^4)$

$N_1 = \{1,\ 4\},\ C_1^4 = \{1,\ 4\} \rightarrow N_1^4 = \emptyset$

$N_o^4 = N_o - (C_o^4 \cup D_o^4)$

$N_o = \{1,2,3,4\},\ C_o^4=\{3,4\},\ D_o^4=\emptyset \rightarrow N_o^4 = \{1,2\}$

Prema 5b prelazi se na

6. korak: Ispitivanje relacije (27) za k = 0, i = 1,3

$$\sum_{j\varepsilon N_o^4} a_{1j}^- = -2 \nleq y_1^o = -4$$

Ovo je slučaj 6a, pa treba brisati v_1^o i v_2^o.

Kako se relacija (27) dalje ne može ispitivati (ispitana je za sve vrijednosti indeksa k), algoritam je okončan.

Prema 5a, optimalno je rješenje u^3 jer je $z_3 = z^{*(4)} = 6$.

Prema tome, funkcija z iz problema P postiže minimum z=6 za vrijednosti varijabli

$$x_1 = 1,\ x_2 = 1,\ x_3 = 1,\ x_4 = 0,\ x_5 = 0.$$

Odgovarajuće rješenje početnog zadatka je

$$x_1' = 0,\ x_2' = 1,\ x_3' = 0,\ x_4' = 0,\ x_5' = 1$$

a vrijednost funkcije cilja min $z' = -1$.

U tabeli 2 pregledno je prikazana provedba aditivnog algoritma za primjer 1, po algoritamskim koracima. U stupcima tabele su redom napisani; iteracija, skup J_s za odgovarajuću iteraciju, algoritamski koraci u svakoj iteraciji.

Niz rješenja		Algoritamski koraci
s	J_s	
0	∅	1b, 2b, 3b, 8
1	3	1b, 2b, 3b, 8
2	3,1	1b, 2b, 3b, 8
3	3,1,2	1a, 5b, 6a, 5a, 6b, 8
4	3,1,2,4	1a, 5b, 6a, 5a.

Tabela 2.

Na slici 1, danoj u sklopu aditivnog algoritma, može se pratiti rješavanje primjera 1. Starta se s rješenjem

u^o za koje je $x_j = 0$ $(j = 1,\dots,5)$.

Potom se dobije rješenje

$$u^1 \text{ ta koje je } x_j = \begin{cases} 1 \text{ za } j = 3 \\ 0 \text{ za } j = 1,2,4,5 \end{cases}$$

(na slici tom rješenju odgovara čvor T_1).

Iduće rješenje koje daje algoritam je

$$u^2 \quad s \quad x_j = \begin{cases} 1 \text{ za } j = 1,3 \\ 0 \text{ za } j = 2,4,5 \end{cases}$$

Na slici se vidi kojom granom treba ići da bi se došlo u čvor T_2 koji odgovara tom rješenju.

Rješenje koje slijedi je

$$u^3 \text{ za koje je } x_j = \begin{cases} 1 \text{ za } j = 1,2,3 \\ 0 \text{ za } j = 4,5 \end{cases}$$

Iako se na kraju pokaže da je ovo optimalno rješenje, prema algoritmu, potrebno je ispitati još i rješenje

$$u^4 \quad s \quad x_j = \begin{cases} 1 \text{ za } j = 1,\ 2,\ 3,\ 4 \\ 0 \text{ za } j = 5 \end{cases}$$

ali se ustanovi da grana kroz pripadni čvor T_4 ne vodi do optimalnog rješenja, odnosno da je optimalno rješenje već prije postignuto.

Algoritam se zaustavlja "dolaskom" na posljednji, peti nivo u čvor T_5 - taj čvor odgovara rješenju u^3 jer se u njega dodje iz čvora T_3 spuštanjem po lijevim granama.

Primjer 2: Odrediti minimum funkcije

$$z' = 4x_1' - 3x_2' - 12x_3' + x_4' + 8x_5'$$

uz uvjete

$$\begin{aligned} -x_1' + 2x_2' - 5x_3' - 4x_4' + 6x_5' &\leq 2 \\ 4x_1' - 5x_2' - x_3' + 4x_4' - 8x_5' &\geq 5 \\ 2x_1' - x_2' + 4x_3' - x_4' + 2x_5' &\leq 0 \\ x_j' = 0 \text{ ili } 1 \ (j = 1,\ldots,5) & \end{aligned}$$

Da bi se mogao primijeniti Balasov algoritam na rješavanje ovog zadatka, potrebno je provesti slijedeće transformacije:

- drugu nejednadžbu pomnožiti s -1
- zamijeniti varijable kako slijedi

$$x_j = \begin{cases} x_j' & j = 1,4,5 \\ 1 - x_j' & j = 2,3 \end{cases}$$

Nakon toga zadani problem poprima oblik:

Odrediti minimum funkcije

$$z = 4x_1 + 3x_2 + 12x_3 + 4x_4 + 8x_5$$

uz uvjete

$$-x_1 - 2x_2 + 5x_3 - 4x_4 + 6x_5 + y_1 = 5$$

$$-4x_1 - 5x_2 - x_3 - 4x_4 + 5x_5 + y_2 = -11$$

$$2x_1 + x_2 - 4x_3 - x_4 + 2x_5 + y_3 = -3$$

$$x_j = 0 \text{ ili } 1 \ (j=1,....,5)$$

Ovaj problem, pa stoga i polazni, nema rješenje. Tok rješavanja tog problema, napisan po algoritamskim koracima, vidljiv je iz slijedeće tabele (tabela 3):

Niz rješenja		Koraci
s	J_x	
0	Ø	1b,2b,3b,8
1	4	1b,2b,3b,8
2	4,2	1b,2b,3b,8
3	4,2,3	1b,2b,3b,8
4	4,2,3,1	1b,2a,5b,6b
5	4,2,3,1	1b,2a,5b,6b
6	4,2,3,1	1b,2a,5b,6a,5b,6a,5a

Tabela 3

L I T E R A T U R A

1. *Lj.Martić, Matematičke metode za ekonomske analize II, Narodne novine, Zagreb, 1972, str.256-257.*

2. *Lj.Martić, Kvantitativne metode za financijske i računovodstvene analize, Ekonomski fakultet, Zagreb,1978.*

3. *E.Balas, An additive algorithm for solving linear programs with zero-one variables, Operations Research, Vol. 13, 1965, str.517-546.*

4. *S.Ashour and A.R.Char, Computational Experience on Zero-One Programming Approach to Verious Combinatorial Problems, Journal of the Operations Research Society of Japan, Vol.13, No.2, 1970, str.78-108.*

Primljeno: 1980-11-01

Hunjak T. A Description of Balas's Algorithm for Zero-One Programming

S U M M A R Y

In this paper the author presents an algorithm for solving linear programs with zero-one variables.

In the introductory part of the paper attention is called to the importance of zero-one programming problems.

The paper itself is divided into thres parts: the first part discusses the basic ideas of the algorithm, the second contains a description of the algorithm, while the third provides the solutions of two numerical examples.