Mr. Vesna Dušak

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

UDK: 65.012.122:656

Stručni rad

PREDNOSTI GAP-ALGORITMA U RJEŠAVANJU DEGENERIRANIH TRANSPORTNIH PROBLEMA

Transportni problemi linearnog programiranja često imaju degenerirano bazično rješenje. U radu su prikazani različiti pristupi u rješavanju degeneriranih bazičnih rješenja klasičnog transprotnog problema linearnog programiranja s ograničenim propusnim sposobnostima (2-TP_c problem), s posebnim osvrtom na prednosti primjene General Alternating Path algoritma (GAP-algoritam) u izboru varijable koja omogućuje skok na novo nedegenerirano bazično rješenje

Transportni problem; degeneracija; GAP-algoritam

1.0. TRANSPORTNI PROBLEM LINEARNOG PROGRAMIRANJA S OGRANIČENIM PROPUSNIM SPOSOBNOSTIMA

Jedan od odlučujućih činilaca za izbor tipa transportnog sredstva za neku relaciju upravo su karakteristike tih relacija (usponi, padovi, zavoji, nosivosti...).Ovo se odražava kroz tzv. "propusne sposobnosti" relacija. Stoga je za rješavanje transportnih problema opravdano koristiti transprotni problem linearnog programiranja s ograničenim propusnim sposobnostima koji se matematički notira kao (4):

$$\text{naći skup } X = \{x_{ij}\}, \qquad i \in I = (1,2,\dots,m) \quad j \in J = (1,2,\dots,n),$$

koji minimalizira funkciju

$$L(X) = \sum_{i,j} c_{ij}\, x_{ij}, \qquad \forall (i,j) \in I \times J, \tag{1}$$

uz ograničenja

$$\sum_j x_{ij} = a_i \qquad \forall i \in I, \tag{2}$$

$$\sum_i x_{ij} = b_j \qquad \forall j \in J, \tag{3}$$

$$0 \le x_{ij} \le d_{ij}, \quad \forall (i,j) \in I \times J; \tag{4}$$

gdje je

m - broj ishodišta,
n - broj odredišta,
x_{ij} - količina tereta koja će se prema programu prevoziti na relaciji $i \to j$,
c_{ij} - troškovi prijevoza tereta na relaciji $i \to j$,
a_i - kapacitet i-tog ishodišta,
b_j - potrebe j-tog odredišta,
d_{ij} - propusna sposobnost relacije $i \to j$.

Česta pojava prilikom rješavanja ovog problema je degeneracija. Degenerirano bazično rješenje javlja se u slučajevima kada je broj relacija $i \rightarrow j$ s veličinom prijevoza x_{ij}, $0 < x_{ij} < d_{ij}$, manji od $m+n-1$. Tom prilikom dolazi do iteracija s nultim korakom (funkcija cilja se ne mijenja), odnosno do određenog kruženja, dok se ne pronađe relacija koja će bazično rješenje svesti na jedinstveno drvo i omogućiti skok na bazično rješenje koje će smanjiti vrijednost funkcije cilja (ako postignuto rješenje već nije optimalno).

Da bi se degenerirani transportni problemi riješili, koriste se različite metode. U već tradicionalni pristup spada ε-pristup, te rješavanje pomoćnih problema izvedenih iz degeneriranih bazičnih rješenja, a pomoću principa dozvoljenih smjerova. Ovi pristupi zahtijevaju modificiranje originalnog problema već prilikom njegova postavljanja, odnosno u toku rješavanja.

Noviji pristup rješavanju degeneriranih problema polazi od svojstva transportnog problema da njegova bazična rješenja u grafičkom smislu predstavljaju jedinstveno drvo. Posebnim postupkom, tzv. GAP-algoritmom (The Generalized Alternating Path Algorithm), izabire se za povezivanje ona grana koja eliminira problem degeneracije i kruženja.

Problem (1) - (4) ima rješenje (4) ako je ispunjen uvjet ravnoteže

$$\sum_i a_i = \sum_j b_j, \tag{5}$$

te uvjet

$$\sum_{i \in I'} a_i - \sum_{j \in J'} b_j \leq \sum_{i \in I'} \sum_{j \in J \setminus J'} d_{ij} \quad \forall\, I' \subset I,\ J' \subset J, \tag{6}$$

ili uvjet

$$\sum_i \min\Big(a_i, \sum_{j \in J'} d_{ij}\Big) \geq \sum_{j \in J'} b_j \quad \forall\, J' \subset J. \tag{7}$$

Ovo rješenje je nedegenerirano ako bazične varijable ne poprimaju granične vrijednosti, tj. $0 < x_{ij} < d_{ij}$, $(i,j) \in B$ (B-skup relacija (i,j) koje čine bazu rješenja). Promatrajući dualni oblik problema (1) - (4) -

naći skup $V = \{u_i, v_j, w_{ij}\}$, $i \in I$, $j \in J$, $(i,j) \in I \times J$,

koji će maksimalizirati funkciju

$$L(V) = \sum_i a_i u_i + \sum_j b_j v_j - \sum_{i,j} c_{ij} w_{ij} \tag{8}$$

uz ograničenja

$$u_i + v_j - w_{ij} \leq c_{ij} \quad \forall\, (i,j) \in I \times J, \tag{9}$$

$$w_{ij} \geq 0 \quad \forall\, (i,j) \in I \times J, \tag{10}$$

rješenje je nedegenerirano ako su nebazične komponente koplana[1] prijevoza (δ, B) različite od nule. Pri tom je

$$\delta = A^T y - c, \quad y = [u, v], \tag{11}$$

a komponente pseudoplana glase:

$$\varkappa_{ij} = \begin{cases} 0 & \text{za } \delta_{ij} < 0 \\ d_{ij} & \text{za } \delta_{ij} > 0, \\ 0 \leq \varkappa_{ij} \leq d_{ij} & \text{za } \delta_{ij} = 0. \end{cases} \quad (i,j) \notin B, \tag{12}$$

1 - *Koplan prijevoza predstavlja skup brojeva* $= \{\delta_{ij},\ i \in I,\ j \in J\}$, *takvih da vrijedi* $\delta_{ij} = u_i + v_j - c_{ij}$, *odnosno relacija (11).*

$$\sum_{(i,j)\in B_i} \kappa_{ij} = a_i - \sum_{(i,j)\notin B_i} {}_{ij}\,,$$
$$\sum_{(i,j)\notin B_j} \kappa_{ij} = b_j - \sum_{(i,j)\in B_j} {}_{ij}\,.$$

Bazično rješenje originalnog problema je optimalno ako ocjene relacija, tzv. relativni troškovi, glase:

$$\Delta_{ij} \begin{cases} \leq 0, & x_{ij} = 0 \\ \geq 0, & x_{ij} = d_{ij} \\ = 0, & 0 < x_{ij} < d_{ij} \end{cases}, \quad (i,j)\notin B, \tag{13}$$

$$\begin{aligned} \Delta_{ij} &= u_i + v_j - c_{ij}, \quad (i,j) \notin B, \\ u_i + v_j &= c_{ij}, \quad (i,j) \in B. \end{aligned} \tag{14}$$

U optimalnom slučaju, gledajući dualni oblik problema, i bazične komponente koplana prijevoza zadovoljavaju uvjete (12).

2.0. TRADICIONALNE METODE RJEŠAVANJA DEGENERIRANIH TRANSPORTNIH PROBLEMA

Karakteristika ovih metoda je da se ne rješavaju originalni problemi, već se pristupa određenim modifikacijama. Tako se problem može već u samom početku rješavanja modificirati kroz vektor ograničenja. Pri tome se svakom ishodišnom kapacitetu dodaje dovoljno mali broj ε (2):

$$a'_i = a_i + \varepsilon, \ \forall\, i \in I, \ 0 \leq \varepsilon \leq \varepsilon_0, \quad \varepsilon_0 \leq (1/2m) \tag{15}$$

i radi uvjeta (5)

$$b'_n = b_n + \varepsilon\, m. \tag{16}$$

Ovako modificirani problem rješava se klasičnim metodama, a također i njihovim modifikacijama, namijenjenim za rješavanje "kvazidegeneriranih" problema. Dobiveno optimalno rješenje korigira se za veličinu ε.

Drugi pristup odnosi se na rješavanje pomoćnih, izvedenih problema, koji se formiraju pomoću parametara dobivenih u nekoj iteraciji rješavanja problema kada je nastupila degeneracija. Koristi se princip dozvoljenih smjerova[2].

Kada je rješenje problema (1) - (4) degenerirano, tada postoje komponente baze koje poprimaju kritične vrijednosti (x_{ij} = 0 ili d_{ij}). Stoga je kod provjere optimalnosti rješenja nužno da i bazične i nebazične komponente relativnih troškova udovoljavaju relacijama (13). Ove dvije skupine uvjeta (za bazične i nebazične komponente) objedinjava slijedeći pomoćni problem (2):

naći skup $\{p_{ij}, q_{ij}\}$, $(i,j) \in I\times J$,
koji minimalizira funkciju

$$L(p_{ij},q_{ij}) = \sum_{(i,j)\in B'_N \cup B'_p} f_{ij}\, p_{ij} + \sum_{(i,j)\in B'_V \cup B'_P} g_{ij}\, q_{ij} \tag{17}$$

2 - Traži se vektor $1 = \{1_{ij}, i\in I, j\in J,\}$ za koji će rješenje pomoćnog problema $A(x+\varepsilon 1)=b$, $0\leq x+\varepsilon 1\leq d$, $\varepsilon_0>0$, $0<\varepsilon<\varepsilon_0$, biti rješenje problema (1)-(4).

uz ograničenja

$$
\begin{aligned}
&u_i + v_j - c_{ij} &&= 0 && (i,j) \in B_P,\\
&u_i + v_j - c_{ij} &&\le 0 && (i,j) \in B_N,\\
&u_i + v_j - c_{ij} &&\ge 0 && (i,j) \in B_V,\\
&u_i + v_j - c_{ij} - p_{ij} + q_{ij} &&= 0 && (i,j) \in B'_{P'}\\
&u_i + v_j - c_{ij} - p_{ij} &&\le 0 && (i,j) \in B'_{N'}\\
&u_i + v_j - c_{ij} + q_{ij} &&\ge 0 && (i,j) \in B'_{V'}
\end{aligned}
\tag{18}
$$

$p_{ij} \ge 0 \quad (i,j) \in B'_N \cup B'_{P'}$

$q_{ij} \ge 0 \quad (i,j) \in B'_V \cup B'_{P'}$

gdje je

$B_P = \{(i,j) \in B : 0 < x_{ij} < d_{ij}\}, \quad B'_P = \{(i,j) \notin B, 0 < x_{ij} < d_{ij}\},$

$B_N = \{(i,j) \in B : x_{ij} = 0\}, \quad B'_N = \{(i,j) \notin B, x_{ij} = 0\},$

$B_U = \{(i,j) \in B : x_{ij} = d_{ij}\}, \quad B'_V = \{(i,j) \notin B, x_{ij} = d_{ij}\}.$

Dualni oblik ovog pomoćnog problema predstavlja problem izbora optimalnog smjera za zadatak (1) - 4):

naći skup $\{1_{ij}\}$, $i \in I$, $j \in J$,

koji minimalizira funkciju

$$L(1_{ij}) = \sum_{i,j} c_{ij} 1_{ij} \tag{19}$$

uz ograničenja

$$\sum_j 1_{ij} = 0, \quad \sum_i 1_{ij} = 0 \tag{20}$$

$$0 \le 1_{ij} \le f_{ij}, \quad (i,j) \in B'_{N'} \tag{21}$$

$$g_{ij} \le 1_{ij} \le 0, \quad (i,j) \in B'_{V'} \tag{22}$$

$$-g_{ij} \le 1_{ij} \le f_{ij}, \quad (i,j) \in B'_{P'} \tag{23}$$

$$1_{ij} \ge 0, \quad (i,j) \in B'_{N'} \tag{24}$$

$$1_{ij} \le 0, \quad (i,j) \in B'_V. \tag{25}$$

Rješavanje ovog problema dualnom metodom daje informaciju o koeficijentima p_{ij}, q_{ij}, odnosno 1_{ij}. Rješenje je optimalno kada su vrijednosti $p_{ij}=q_{ij}=0$. Tada komponente novog rješenja glase $x_{novo} = x + \Theta^0 1^0$, gdje je Θ^0 - količina koja se seli, a 1^0 - optimalne vrijednosti vektora 1. $\Theta^0 > 0$ pa dolazi do skoka na novu iteraciju čije bazično rješenje dovodi do sniženja vrijednosti funkcije za $\Theta^0 \sum_{i,j} c_{ij} 1^0_{ij}$, ali pri tome često treba ispitati sve cikluse koje je moguće formirati (po svakoj nenultoj nebazičnoj komponenti vektora 1^0).

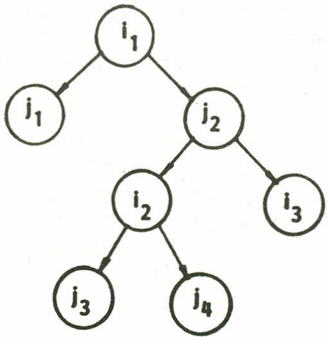
Isti princip može se primijeniti i za rješavanje dualnog problema, s time da se pomoćni problem izvodi pomoću koplana prijevoza, povezujući relacije (12) za bazične i nebazične varijable.

3.0. GAP-ALGORITAM

GAP-algoritam zasniva se na konstrukciji specifičnih baza, tzv. GAP-baza. U grafičkom smislu bazično rješenje transportnog problema s ograničenim propusnim sposobnostima (drvo s korijenom) predstavlja GAP-bazu ako je (1):

1) čvor-korijen ishodišni čvor,
2) sve pozitivne veze $(i \to j)$ imaju tok $x_{ij} > 0$,
3) sve negativne veze $(j \to i)$ imaju tok $x_{ij} < d_{ij}$.

Ovo je identično činjenici da u GAP-bazi svi parni lukovi imaju tok $x_{ij} > 0$, a svi neparni tok $x_{ij} < d_{ij}$, ili obratno (što se vidi iz Slike 1). Odnosno, jedni su tolerantni na povećanje, a drugi na smanjenje toka do kritičnih vrijednosti (0 ili d_{ij}). Značajno svojstvo GAP-baze je da ako postoji optimalno rješenje problema, onda ono ima strukturu GAP-baze.



Slika 1.

Konstrukcija GAP-baze sastoji se od nekoliko koraka (2):

1) Iz promatranja se odstranjuju svi lukovi s tokovima na donjoj ili gornjoj granici:

$$A' = \{(i,j) \in I \times J \ : \ 0 < x_{ij} < d_{ij}\}. \quad (26)$$

Za nedegenerirano rješenje A=B. Kod degeneriranog rješenja graf $G=(I',J',A')$ koji korespondira skupu A' raspada se na odvojena stabla

$$G_l(I'_l, J'_l; A'_l) \subset G(I',J';A'), \ \sum_{l=1}^{r} G_l = 0, \ l = 1,2,\ldots,r \quad (27)$$

i izolirane čvorove $\sigma = \{j_s\}$, $s = 1,2,\ldots,p$;

2) Za svako drvo odredi se jedan izvorišni čvor $i^l_1 \in I'_l$ kao korijen,

$$K = \{ i^l_1, \ i^l_1 \in I'_l\}, \quad l = 1,2,\ldots,r\}; \quad (28)$$

3) Izolirani odredišni čvorovi spajaju se s nekim od stabala lukom s tokom na donjoj granici i dodjeljuje mu se pozitivna orijentacija $(i \to j)$:

$$(i^l_1, j_s) : i^l_1 \in I_l', \quad j_s \in \sigma, \quad x^l_{i1}\, j_s = d^l_{i1}\, j_s \qquad s=1,2,\ldots,p; \quad (29)$$

4) Odvojena stabla međusobno se spajaju u jedinstveno drvo lukom s nultim tokom od korijena jednog stabla do odredišnog čvora drugog stabla, tj. izabire se luk

$$(j_{1'},\ i^1_{1''}) : j_{1'} \in J_1',\ i_{1''} \in I_{1''},\ x^1_{i1''\,j_{1'}} = 0,\quad 1' \neq 1'' = 1,2,\ldots,r \tag{30}$$

s negativnom orijentacijom ($j \rightarrow i$).

Ovim postupkom dobivena je struktura drva koja udovoljava svim zahtjevima GAP-baze. Provodeći zamjenu vektora (lukova) na ovako oformljenoj bazi, postiže se nedegenerirano rješenje. Da bi se opisana procedura mogla provesti, potrebno je da uz ranije opisane uvjete za postojanje rješenja (5) - (7), problem (1) - (4) zadovoljava i relacije

$$\begin{aligned} a_i &= 0, \quad \forall i \in J, \\ b_j &= 0, \quad \forall j \in J. \end{aligned} \tag{31}$$

4.0. OPRAVDANOST PRIMJENE GAP-ALGORITMA ZA RJEŠAVANJE DEGENERIRANIH 2-TP$_c$ PROBLEMA

Prethodno su opisane metode rješavanja degeneriranih transportnih problema - ε-pristup, metoda traženja optimalnih smjerova, GAP-algoritam. Pokazane su prednosti i nedostaci ovih metoda u konstrukciji baznih rješenja dvodimenzionalnih problema linearnog programiranja transportnog tipa s ograničenim propusnim sposobnostima te način konstrukcije jedinstvenog drveta baznog rješenja prema GAP-algoritmu u smislu izbjegavanja degeneracije i kruženja, odnosno bržeg pronalaženja optimalnog smjera kod pojave degeneriranih baznih rješenja. Dok se kod metode traženja optimalnog smjera, kod pojave degeneriranih baznih rješenja 2-TP$_c$ problema, mora riješiti nezgrapan program linearnog programiranja (da bi se osigurao skok na novu nedegeneriranu točku), kod GAP-algoritma to se postiže primjenom specifičnih i strožih pravila u odnosu na promjenu baznog rješenja MODI metodom, a odnose se na izgradnju jedinstvenog drveta baznog rješenja i izbor izlaznog luka.

Zapis problema traženja optimalnog smjera za degenerirano bazno rješenje 2-TP$_c$ problema dano je izrazom (19) - (25). Ograničenja (19), (24) i (25) odnose se na količinu koja će izmijeniti komponente postojećeg baznog rješenja. Ta se količina u zatvorenom ciklusu (koraku izmjene baznog rješenja kod MODI metode) naizmjenično dodaje, odnosno oduzima u sukcesivnom slijedu kako ne bi narušila ravnotežu sustava ograničenja odnosno kapaciteta pojedinih čvorova. Promatrajući matrični zapis istog problema, suma veličina tih promjena po recima i stupcima mora biti 0. Istovremeno se ova količina ne može oduzeti na skupu lukova kojima je pridružena količina $x_{ij} = 0$ (na donjoj granici), tj. ne može se dodati količini x_{ij} saturiranih lukova.

Kod GAP-algoritma bazno rješenje prikazuje se preko stabla s korijenom. U svakoj grani stabla luk ima svoje tzv. prethodnike, kojima se zatvara put prema čvoru-korijenu i nasljednike kojima se zatvara put do zadnjeg čvora u grani. Pri tom se u svakoj grani sukcesivno izmjenjuju lukovi suprotnih orijentacija, odnosno sukcesivno se izmjenjuju ishodišni i odredišni čvorovi (čvor-korijen je ishodišni čvor).

Kod koraka izmjene baznog rješenja GAP-algoritma također se zatvara ciklus koji zadovoljava gornja ograničenja - suma ukupnih promjena količina na ciklusu je nula, a ograničenja $1_{ij} \geq 0$ za lukove kojima su pridružene količine $x_{ij} = 0$, odgovaraju vezama koje dozvoljavaju povećanje toka (upper leeway links), dok ograničenja $1_{ij} \leq 0$ za lukove sa saturiranim tokom odgovaraju vezama koje dozvoljavaju smanjenje toka (lower leeway links). Ovo je naročito važno za nebazične varijable prilikom povezivanja substabala degeneriranog baznog rješenja. Bazične varijable zadovoljavaju oba ova ograničenja (double leeway links).

Kod oba pristupa također su ispunjeni osnovni uvjeti nenegativnosti parametara problema a_i, $i \in I$; b_j, $j \in J$ i d_{ij} $(i\ j) \in I \times J$.

U istom zapisu dalje slijede tzv. normirajući uvjeti:

$1_{ij} \leq f_{ij}$, $(i,j) \in \{(i,j) : x_{ij} = 0\}$ koji dopunjuje zahtjev GAP-baze za vezama koje dozvoljavaju povećanje toka i

$1_{ij} \leq -g_{ij}$, $(i,j) \in \{(i,j) : x_{ij} = d_{ij}\}$ koje dopunjuje uvjet GAP-baze za vezama koje dozvoljavaju smanjenje toka. Uvjet $-g_{ij} \leq 1_{ij} \leq f_{ij}$, $(i,j) \in (i,j) : 0 < x_{ij} < d_{ij}\}$ odgovara karakteristikama baznih varijabli 2-TP$_c$ problema koje dozvoljavaju i povećanje i smanjenje toka do krajnjih granica. Sve ovo zaokružuje se u uvjet za postojanje GAP-baze, a to je da sve $(i \rightarrow j)$[3] veze dozvoljavaju smanjenje veličine toka, a sve $(j \rightarrow i)$ veze povećavanje veličine toka, što omogućuje sukcesivnu izmjenu količina u ciklusu formiranom za izmjenu baze.

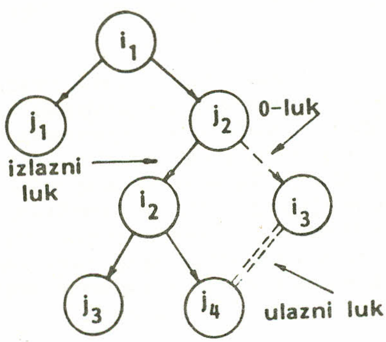
Rješavajući problem (19) - (25), ili je pronađen optimalan program ($p_{ij} = q_{ij} = 0$), ili luk, odnosno bazična komponenta kojom će se omogućiti skok na novo degenerirano bazično rješenje. To su komponente vektora 1^0. Supstitucijom

$$f_{ij} = -d_{ij} - x_{ij} \quad (i,j) \in B_N \cup B_P, \; p_{ij} \geq 0, \; x_{ij} \geq 0 \tag{32}$$

$$g_{ij} = -x_{ij} \quad (i,j) \in B_V \cup B_P, \; q_{ij} \geq 0, \; x_{ij} \leq d_{ij}; \tag{33}$$

omogućen je prijelaz na originalan problem. Sada je moguće provjeriti podudarnost ovih dviju metoda:

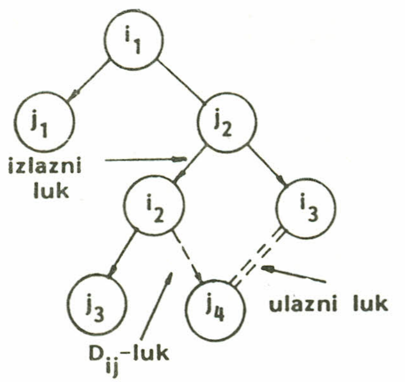
1) **ako je izoliran ishodišni čvor**, tada to znači da u nekoj grani nedostaje luk nasljednik $(j \rightarrow i)$ orijentacije (slika 2). Lukovi negativne orijentacije, prema pravilima GAP-algoritma, trebaju dozvoljavati povećanje veličine toka na njima. Stoga prilikom prevođenja degeneriranog u nedegenerirani problem, izolirani ishodišni čvor treba povezati lukom s 0-tokom i orijentirati ga od odredišta prema ishodištu. Luku, koji je uveden, treba dodijeliti određenu količinu toka $0 \leq 1_{ij} = d_{ij} - x_{ij}$. Količinu 1_{ij} treba dodati svim tokovima na lukovima $(j \rightarrow i)$ orijentacije. Lukovima $(i \rightarrow j)$ orijentacije treba oduzeti količinu $0 \leq 1_{ij} \leq -x_{ij}$. Dodavanje odnosno oduzimanje vrši se sukcesivno i da se ne bi desilo da neka od varijabli bazičnog rješenja bude negativna, moraju biti ispunjeni uvjeti (32) i (33) čime se ne narušavaju gornje i donje granice veličina bazičnih lukova. To se upravo i postiže kada je pronađen optimalni smjer, iz čega proizlazi da se na izbor luka, koji će omogućiti skok na nedegenerirano bazično rješenje, mogu primijeniti pravila izbora ulaznog i izlaznog luka na način kako je to predviđeno GAP-algoritmom (pravila Ia i Ib). /1/



Slika 2.

3 - $(i \rightarrow j)$ - oznaka za pozitivno orijentiranu vezu, neparni luk, koji mora biti osjetljiv na smanjivanje veličine toka.
$(j \rightarrow i)$ - oznaka za negativno orijentiranu vezu, parni luk, koji mora biti osjetljiv na povećavanje veličine toka.

2) **ako je izolirani čvor odredišni čvor**, tada u nekoj grani baznog rješenja nedostaje luk (i → j) orijentacije (slika 3). Luk s takvom orijentacijom mora biti osjetljiv na smanjenje veličine toka. Stoga na njemu treba oduzeti neku veličinu toka, pa će se za povezivanje izabrati ne luk s 0-tokom već nebazični luk sa saturiranim tokom. Na taj će se način moći izgraditi ciklus izmjene baznog rješenja u kojem će se naizmjenično nizati lukovi osjetljivi na smanjenje i lukovi osjetljivi na povećanje toka. Prema tome, lukovima s pozitivnom orijentacijom treba oduzeti količinu $-g_{ij} \le l_{ij} \le 0$, a lukovima s negativnom orijentacijom dodati količinu $0 \le l_{ij} = d_{ij} - x_{ij}$. Pri tom je zadovoljen uvjet nenarušavanja gornjih odnosno donjih ograda veličine tokova na lukovima i pronađen optimalan smjer, a to ukazuje da su na izbor luka koji će omogućiti prijelaz s degeneriranog na nedegenerirano bazično rješenje primjenljiva pravila GAP-algoritma (pravila IIa i IIb). /1/



Slika 3.

5. ZAKLJUČAK

Problem degeneracije kod transportnih problema linearnog programiranja zahtijeva posebnu pažnju jer je ona česta pojava kod ove grupe problema, a često dovodi i do niza iteracija koje ne predstavljaju kvalitativni skok do optimalnog rješenja. Postoji više pristupa za otklanjanje degeneracije. Uglavnom se svode na modifikaciju početnog problema i rješavanje pomoćnih problema, što sve dovodi do većeg broja proračuna, a u slučaju rješavanja problema elektroničkim računalom i do povećanja veličine kapaciteta potrebne memorije. Sve to znatno usporuje proces rješavanja problema.

GAP-algoritam omogućuje otklanjanje degeneracije "u hodu", bez potrebe vraćanja na početno bazično rješenje i njegovu modifikaciju. Dodavanjem potprograma za strukturiranje GAP-baze kod degeneriranih bazičnih rješenja u set programa za rješavanje transportnog problema s ograničenim propusnim sposobnostima, omogućuje se nastavak rješavanja problema od onog koraka na kome je degeneracija nastupila. Nije potrebno vraćanje na početno bazično rješenje, kao ni modifikacija problema na originalni oblik nakon postignutog optimalnog rješenja. Usporedba principa izgradnje jedinstvenog drveta bazičnog rješenja transportnog problema linearnog programiranja s ograničenim propusnim sposobnostima s metodom traženja optimalnog smjera, pokazala je sve prednosti primjene GAP-algoritma:

- jednostavnost postupka,
- kratkoću postupka,
- izbjegavanje kruženja i "nultih" iteracija.

LITERATURA

1. R.S.Barr, F.Glover, D.Klingman: The Generalized Alternating Path Algorithm for transportation problems, EJOR2, 1978, 134-144.
2. V.Dušak: Neke mogućnosti rješavanja degeneriranih transportnih problema, Zbornik radova SYM-OP-IS '86, 123-130.

3. V.Dušak: Primjena GAP-algoritma za rješavanje degeneriranih 3-TP_c problema, Zbornik radova SYM-OP-IS '87, 189-196.

4. V.A.Emeličev, M.M.Kovalev, M.K.Kravcov: Mnogogranniki, grafy, optimizacija, Nauka, Moskva, 1981.

5. M.S:Hung, W.O.Rom, A.D.Waren: Degeneracy in transportation problems, Discrete Applied Mathematics, 13 (1986), 223-237.

Primljeno: 1988-09-09

Dušak V. The advantages of the implementation of the GAP-algorithm to degenerated transportation problems

S U M M A R Y

Transportation problems of linear programming often have a degeneratedbasis solution. It provides the circulation and the decomposiotion of the basic spanning tree. The GAP-algorithm establishes the unique basic tree in the way zero-steps are avoided when the basis solution is degenerated.Also it isn't necesary to come back to begining of the procedure, or the modification of the original problem.

The comparation of principles of the construction the unique basic tree and crossing ower to the new nondegenerated basis solution of the 2-TP_C problem, to method for searching the optimal direction of a degenerated basis solution, reprezents all the advantages of implementation the GAP-algorithm: the simplicity and the efficiency of the algorithm, and the avoidance of zero-steps.