EKONOMSKA ANALIZA
3, XVIII (1984), 259—267

SADAŠNJE STANJE I PRAVCI RAZVOJA CJELOBROJNOG PROGRAMIRANJA

*Ljubomir MARTIĆ**

Pratim razvoj cjelobrojnog programiranja gotovo od samog njegovog početka. Pregled prvih dostignuća u tom području objavio sam 1966. godine pod naslovom »Aktualna situacija u cjelobrojnom programiranju«. Iduće godine izašao je moj rad o problemu trgovačkog putnika koga će C. Kastning 1976. godine uvrstiti u svoju klasificiranu bibliografiju radova iz cjelobrojnog programiranja i povezanih područja. Temama iz cjelobrojnog programiranja vraćao sam se još nekoliko puta. A 1979. godine u Sveučilišnom računskom centru »Srce« u okviru Seminara iz matematičkog programiranja održao sam predavanje pod naslovom »Dvadeset godina razvoja cjelobrojnog programiranja«. To dosad neobjavljeno predavanje koncipirano je kao uvod u niz predavanja i diskusija u Seminaru sa temama iz cjelobrojnog programiranja. Nije se radilo o nekom sveobuhvatnom pregledu programiranja u cijelim brojevima. Namjera mi je bila da spomenem samo ponešto od onoga što se u cjelobrojnom programiranju postiglo do kraja 1978. godine. Ovdje ću proširiti i dopuniti to moje predavanje najnovijim rezultatima. Posebno ću istaknuti neke rezultate i naznačiti neke teme i otvorene probleme koji mogu biti predmetom našeg zajedničkog rada.

Iz popisa literature vidi se da je prvi rad u kojem je utvrđen predmet cjelobrojnog programiranja objavio Ralph Gomory 1958. godine. U njemu je zapisan ovaj problem:

$$Max\, z\,(x, v) = c_1 x + c_2 v, \qquad (x, v) \in S, \tag{1}$$

gdje je

$$S = \{(x, v) \mid A_1 x + A_2 v = b, \quad x \geq 0 \text{ cijeli}, \quad v \geq 0\}$$

a dani su c_1 kao n_1 — vektor, c_2 kao n_2 — vektor, A_1 matrica reda (m, n_1), A_2 matrica reda (m, n_2). To je problem mješovitog cjelobrojnog i kontinuiranog linearnog programiranja. Kad je $n_2 = 0$, imamo

* Ekonomski fakultet, Zagreb.

slučaj čistog cjelobrojnog programiranja a kad je $n_1 = 0$ slučaj običnog linearnog programiranja. Cjelobrojno programiranje bavi se, dakle, problemima matematičkog programiranja u kojima se zahtijeva da su sve ili samo neke varijable cjelobrojne.

Ako ostanemo u području linearnog programiranja (LP), problem čistog cjelobrojnog programiranja (ILP) može se zapisati ovako:

$$Max\, c'x, \quad x \in S = \{x \mid Ax = b, \quad x \geq 0 \quad cijeli\}, \tag{2}$$

gdje je $A = [a_{ij}]$ matrica reda (m, n). Svi parametri su cjelobrojni.

Ako je S konačan skup, tada postoji vektor u od kojeg su komponente dovoljno veliki cijeli brojevi tako da je

$$x \in S \Rightarrow x \leq u.$$

Problem (2) sa konačnim S može se transformirati u linearni ILP problem ovako:

$$x_j = \sum_{k=0}^{t_j} 2^k \delta_{kj} \quad \text{za svako } j \tag{3}$$

$$\delta_{kj} = 0,\ 1 \quad \text{za svaki par} \quad k, j,$$

gdje je

$$2^{t_j} \leq u_j < 2^{t_j + 1}$$

Ti 0—1 programi imaju neka zgodna svojstva. Skup mogućih rješenja je konačan. Njegovi elementi su vrhovi hiperkocke. Ima ih, dakle, najviše 2^n.

Prilično začuđujući rezultat postignut je u 1970. i 1971. godini. Tada je, naime, dokazano da se svaki konačni ILP može transformirati u binarni knapsack-problem:[1]

$$Max\, z(x) = \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_j x_j \leq b \tag{4}$$

$$x_j = 0{,}1$$

To je problem s jednim ograničenjem a uzima se da je $a_j > 0$, $c_j > 0$ i $b > 0$. Problem se sastoji u tome da se ranac, ruksak (Rucksack-

[1] S. E. Elmaghraby i M. K. Wig agregirali su više jednadžbi u jednu jedinu, primijenivši teorem Mathewsa iz 1897. godine i to rekurzivno, svaki put na dvije jednadžbe. Nedostatak njihovog procesa svođenja više ograničenja na jedno jest u tome što koeficijenti agregirane jednadžbe postaju značajno veliki.

problem) napuni predmetima težine a_j tako da težina čitavog ranca ne pređe b kilograma a da sadržaj ruksaka bude što veće vrijednosti, pri čemu c_j ne mora biti novčana vrijednost predmeta j. Umjesto punjenja ranca, može se razmatrati ukrcavanje tereta u brod i slično. Problem je skrenuo još jednom pažnju na sebe kad je upotrebljen u razvoju jednog algoritma dekompozicije za dobro poznati problem rezanja (trim loss or cutting stock problem). Za problem o rancu nađeno je podosta algoritama a u zadnje vrijeme razmatraju se razna njegova poopćenja. Tako je objavljeno nekoliko radova o generaliziranom knapsack-problemu. U jednom od njih a_j se razmatra kao varijabilni parametar y_j, a svaki c_j je zamijenjen kontinuiranom funkcijom $f_j(y_j)$. U više praktičnih problema ta funkcija je neopadajuća i konkavna nad intervalom $[b_j^-, b_j^+]$ u kojem se kreće y_j.[2]

Ne samo u citiranom časopisu Društva za matematičko programiranje (osnovanog 1971) već i u časopisima iz operacijskog istraživanja sve češće se javljaju radovi iz cjelobrojnog programiranja. Tako su u francuskom časopisu »Recherche opérationelle«, u februarskom broju 1979. godine, prva tri rada iz bivalentnog programiranja. U naslovu drugoga članka istaknuta su tri poznata problema: particije, prekrivanja i uvezivanja (engleski: partitioning, set covering, packing). Ti problemi spadaju zapravo u teoriju grafova a zajedničko im je to da je A binarna matrica dok je b vektor jedinica. Binarnu matricu koeficijenata ima i poznati problem transporta i neke njegove generalizacije i specijalizacije kao što su problem trgovačkog putnika i problem asignacije. Ovaj posljednji je dobro poznat primjer za problem particije. Postoji bogata literatura o tim problemima, posebno o problemu trgovačkog putnika.

Vrlo rano se pokušalo i uspjelo da se transformira mješoviti u čisti problem cjelobrojnog programiranja. No, pod koju cijenu? Broj ograničenja enormno je porastao. Što se tiče transformacije čisto cjelobrojnog linearnog problema u obični LP problem, moguće je svaki konačni ILP, bar u principu, zapisati kao LP problem, dodavajući skupu $S' = \{x \mid Ax = b,\ x \geq 0\}$ ona ograničenja koja definiraju konveksnu ljusku mogućih rješenja problema (2). No, samo za neke vrlo specijalne probleme eksplicitno je dana ta konveksna ljuska.

U literaturi se može naći nešto više od pedeset algoritama za probleme cjelobrojnog programiranja, uključujući i one specijalne. Četiri su glavne grupe algoritama:

1. algoritmi odsijecajućih ravnina,
2. algoritmi enumeracije,
3. algoritmi grananja i ograđivanja i
4. aproksimativne metode.

Rad pod brojem 1 iz popisa literature sadrži prvi algoritam odsijecajućih ravnina — Gomoryjevu metodu cjelobrojnih formi.

Gomoryjev algoritam sastoji se iz tri koraka. U prvom se relaksira problem (2), tako da se maksimum ne traži nad skupom S' svih točaka definiranih ograničenjima. U početku se, dakle, ignorira uvjet

[2] Vidi »Mathematical Programming«, Vol. 15, 1978, №. 2, str. 162—176.

cjelobrojnosti. U drugom koraku testira se optimalno rješenje dobiveno običnom simpleks metodom. Ako je to rješenje cjelobrojno, tada je problem riješen. Ako nije, proslijedi se na treći korak u kojem se necjelobrojni vrh odreže tako da se skupu ograničenja doda je još jedno u obliku linearne nejednadžbe. Zapravo u simpleks tabelu to ograničenje ulazi kao jednadžba, pošto se uvede dodatna varijabla. Ova linearna jednadžba predstavlja tu ravninu (hiperravninu) koja odsijeca onaj necjelobrojni vrh.

Evo kako se generira ta ravnina. Uzme se i-ti redak u finalnoj simpleks tabeli koji sadrži koeficijente ove jednadžbe:

$$x_i^B = t_{io} - \sum_{j \in R} t_{ij} x_j, \tag{5}$$

gdje je x_i^B oznaka za i-tu baziČnu varijablu, t_{io} je tablični element u nultom stupcu a R je skup indeksa nebaziČnih varijabla. Uzima se onaj redak i u kojem t_{io} nije cio broj. Jednadžbu (5) napisat ćemo ovako:

$$x_i^B + \sum_{j \in R} t_{ij} x_j = t_{io}. \tag{6}$$

Tada $x \geq 0$ implicira

$$x_i^B + \sum_{j \in R} [t_{ij}]\, x_j \leq t_{io}, \tag{7}$$

gdje je $[t_{ij}]$ cio broj manji ili jednak t_{ij}. Budući da x mora biti cjelobrojan, lijeva strana od (7) je cio broj pa je stoga

$$x_i^B + \sum_{j \in R} [t_{ij}]\, x_j \leq [t_{io}]. \tag{8}$$

Ako sada od (6) oduzmemo (8) imamo

$$\sum_{j \in R} (t_{ij} - [t_{ij}])\, x_j \geq t_{io} - [t_{io}]. \tag{9}$$

Razlika u zagradi je zapravo razlomački dio od t_{ij}, to jest

$t_{ij} - [t_{ij}] = f(t_{ij})$ ili kraće: f_{ij}.

Slično $t_{io} - [t_{io}] = f_{io}$. Sada (9) možemo pisati ovako:

$$\sum_{j \in R} f_{ij} x_j \geq f_{io}. \tag{10}$$

Budući da smo (10) izveli uz pretpostavku nenegativnosti i cjelobrojnosti varijabli, nejednadžba (10) ne isključuje iz skupa mogućih rješenja relaksiranog problema nijednu cjelobrojnu točku. Napokon (10) pišemo kao jednadžbu ovako:

$$s = -f_{io} + \sum_{j \in R} f_{ij} x_j, \qquad (s \geq 0), \tag{11}$$

gdje je s dodatna varijabla. Budući da je $f_{io} > 0$ (što slijedi iz pretpostavke da t_{io} nije cio broj), bazično rješenje nije moguće tj. nije dopustivo rješenje. Naime, još smo u onom necjelobrojnom vrhu koji je sada izvan suženog skupa mogućih rješenja relaksiranog problema. Stoga sada djelujemo izvan toga skupa dualnom simpleks metodom da dođemo do optimalne ekstremne točke. Kad je dokučimo, prelazimo na drugi korak itd.

Uzgred spominjem da je Gomory pokazao da dodatne nejednadžbe (10) čine jednu aditivnu grupu. Naime, adicijom modulo 1 vektora koeficijenata $(f_{io}, f_{i1}, f_{i2}, \ldots, f_{in})$ ne izlazi se iz njihovog skupa. Drugim riječima, kad dva takva vektora zbrojimo, komponentu po komponentu, i odbacimo cijeli broj sadržan u sumi, dobijemo opet takav vektor koeficijenata odsijecajuće ravnine. Teorija grupa na taj način ušla je u područje cjelobrojnog programiranja.

Opisana metoda je, na žalost, spora iako je konačna.[3] Dokazano je, 1970. godine, da se ne može postaviti nikakva gornja ograda na broj iteracija. Ta metoda je vrlo osjetljiva na greške od zaokruživanja, jer ona mora da besprijekorno raspoznaje cijele brojeve. To je navelo Gomoryja da razvije svecjelobrojnu metodu.[4] No budući da je i ona dualna u smislu da preko rješenja duala dolazi, ali tek na kraju, do optimalnog rješenja primarnog problema, išlo se na razvijanje primarnog (direktnog) svecjelobrojnog algoritma. Prvi takav algoritam u rudimentarnoj formi objavili su Ben-Israel i Charnes (asistent i njegov profesor, tada) 1962. godine. No, Young je 1965. godine prvi razvio konačan direktni algoritam. Još jedan algoritam toga tipa, pojednostavljeni svecjelobrojni, objavio je Young tri godine kasnije. Iste godine pojavio se Gloverov simplificirani direktni algoritam.

Ideja implicitne enumeracije ili parcijalnog (djelomičnog) nabrajanja sastoji se u tome da se skup svih mogućih rješenja pretraži tako da se relativno mali broj rješenja eksplicitno razmotri dok se sva ostala implicite odbacuju. Prvu metodu implicitne enumeracije saopćio je E. Balas 1965. godine. Ta metoda, nazvana aditivnom, razvijena je za rješavanje linearnog 0—1 problema. Glover je nešto kasnije iste godine poboljšao Balasovu shemu nabrajanja tako da je zamijenio sva ograničenja jednim, tzv. surogatnim ograničenjem. Na

---

[3] Nije pomoglo ni to što je odsijecanje usavršeno. Druge načine odsijecanja uveli su Danzig 1959. godine, Glover 1965. godine, Bowman i Nemhauser 1970. godine, Young 1971. godine, Balas 1971. godine, Balas i Jeroslow 1971. godine i opet Glover 1973. godine. Sva ta odsijecanja ugrađena su u algoritme dualnog tipa. Treba napomenuti da nijedan od postojećih načina odsijecanja ne vodi računa o funkciji cilja (kako bi se što više uvećala njezina vrijednost) već svaki od njih ide za tim da odsiječe što veći komad skupa mogućih rješenja relaksiranog problema. Sva nastojanja kod konstrukcije reza išla su, dakle, za tim da rez bude što dublji bez obzira u kojem smjeru.

[4] Gomoryjev rez (cut), tj. način konstrukcije odsijecajuće ravnine u prvom njegovom algoritmu, ponekad se naziva frakcionalnim ili razlomačkim rezom (f-rez) zato što su svi koeficijenti u njoj frakcije ili razlomci. U njegovom svecjelobrojnom algoritmu svojstvo cjelobrojnosti prve simpleks tabele zadržano je u svim ostalima.

bazi Gloverovog rada, originalnu Balasovu shemu dalje su modificirali Geoffrion 1967. i 1969, te sâm Balas 1967. godine i drugi.

Metode implicitne enumeracije zajedno s metodama grananja i ograđivanja čine skupinu metoda traženja ili pretraživanja. Metode implicitne enumeracije u suštini su metode grananja i ograđivanja. Prve se primjenjuju na probleme 0—1 programiranja dok su one druge razvijene za opći problem cjelobrojnog programiranja. Inače obe počivaju na ideji inteligentnog, više implicitnog nego eksplicitnog, pretraživanja skupa mogućih rješenja. Svaki ciklus obuhvata dvije operacije: grananje i ograđivanje. U prvoj se razbija skup rješenja na disjunktne podskupove a u drugoj operaciji računaju se ograde (donje u problemu minimuma a gornje u problemu maksimuma) na vrijednost funkcije cilja u svakom od podskupova. Particija skupa rješenja grafički se predstavlja kao grananje drveta. Skupovi su reprezentirani čvorovima drveta, a ograda dolazi kao neka vrsta etikete na svakom čvoru. Grananje se nastavlja iz čvora sa najmanjom (u slučaju minimuma), odnosno sa najvećom ogradom (u slučaju maksimuma). Broj čvorova u problemima većih dimenzija zanemarivo je malen prema broju cjelobrojnih točaka, tj. svih mogućih rješenja.

Prvu metodu grananja i ograđivanja objavili su Land i Doig 1960. u časopisu »Econometrica« (Vol. 28, str. 497—520). Više poboljšanja i modifikacija izvršili su Dakin (1965), Beale i Small (1965), Bertier i Roy (1965), Balas (1968), Mitten (1970), Taha (1971), Moore (1974) i drugi. Njihove procedure razlikuju se u načinu grananja, tj. particije skupa rješenja, i u postupku određivanja ograda. U posljednjoj dekadi razvijene su brojne heuristike u prvom redu za rješavanje problema velikih dimenzija (Shaw, Tomlin, Forrest i dr). Nađeno je nekoliko algoritama za specijalne probleme kao što su problem trgovačkog putnika i drugi.

Aproksimativne metode (heuristike) najčešće generiraju skup lokalnih optimalnih rješenja. Pođe se od nekog cjelobrojnog x^o iz S pa se traži maksimum od $c'x$ u nekoj okolini $N(x^o)$, točnije u $S \cap N(x^o)$. Neka je x^1 lokalni maksimum. Tada se traži max $c'x$ u okolini od x^1, tj. u skupu $S \cap N(x^1)$ itd. Efikasnost metode zavisi o izboru okolina. Jasno, lokalni optimum, dobiven u svakoj iteraciji, zavisi i o startnoj točki. U pravilu aproksimativne metode primjenjuju se na realne probleme velikih dimenzija. Ponekad takvom metodom dođe se brzo i jeftino do dobre početne točke za neki algoritam enumeracije. Od heuristika u cjelobrojnom programiranju naročito dobro su poznate one od Echolesa i Coopera iz 1968. godine i Hilliera iz 1969. godine.

U novije vrijeme razvija se cjelobrojno ciljno i cjelobrojno višekriterijalno programiranje. Pokazalo se da neki teoremi iz kontinuiranog višekriterijalnog programiranja nisu valjani u uvjetima cjelobrojnosti. Na primjer, svako efikasno rješenje problema Max Cx, $Ax \leq b$, $x > 0$, gdje je C matrica reda (p, n), optimalno je rješenje problema Max λCx, $Ax \leq b$, $x \geq 0$ za neki vektor parametra $\lambda > 0$. Da taj rezultat ne stoji u cjelobrojnom višekriterijalnom programiranju, dovoljno je navesti slijedeći primjer. Točka $(0, 0)$ efikasno je rješenje problema:

$$Max\ z_1 = x_1 - 2x_2$$

$$Max\ z_2 = -x_1 - 3x_2$$

$$x_1, x_2 = 0 \text{ ili } 1.$$

Međutim, ne postoji parametar λ iz $[0, 1]$ tako da je $(0, 0)$ optimalno rješenje problema:

$$Max\ z = \lambda_1 z_1(x) + (1 - \lambda) z_2(x)$$

$$x_1, x_2 \in \{0, 1\}.$$

Naime, $z = (2\lambda - 1)x_1 + (3 - 5\lambda)x_2$. Za tu linearnu funkciju točka $(0, 0)$ mogla bi biti maksimalna, ako i samo ako je $2\lambda - 1 \leq 0$, $3 - 5\lambda \leq 0$. No, taj sistem nije konzistentan, budući da iz prve nejednadžbe izlazi da je $\lambda \leq \frac{1}{2}$ a iz druge nejednadžbe da je $\lambda \geq \frac{3}{5}$.

Na kraju, može se reći da ne postoji unificirana teorija cjelobrojnog programiranja, tj. teorija koja bi pokrivala sve probleme iz tog područja. No, ona je u izgrađivanju. Za specijalne klase problema, kao što su spomenuti 0—1 programi, postoji, može se reći, čak elegantna teorija. Mnogo se učinilo na razvoju metoda i algoritama. Ipak, usprkos velikog broja radova (u Kastningovoj bibliografiji[5] sakupljeno je više od 4.000!) progres u numeričko-računskom aspektu cjelobrojnog programiranja nije ni izdaleka tako impresivan kako se moglo očekivati.

Zašto su sada problemi cjelobrojnog programiranja tako reći u prvom planu? Dva su glavna razloga za to pojačano interesiranje. Jedan su zahtjevi koje postavljaju primjene, u prvom redu ekonomske, a drugi je što ovi problemi kao teži i kompliciraniji dolaze sada na red, pošto su riješeni ili se prilično odmaklo u rješavanju glavnih problema kontinuiranog programiranja.

Primljeno: 28. 12. 1983.
Prihvaćeno: 30. 01. 1984.

CJELOBROJNO PROGRAMIRANJE

Prvi rad:

1. R. E. Gomory, »Outline of an Algorithm for Integer Solution to Linear Programs«, *Bulletin of the American Mathematical Society* 64, 5 (1958), 275—278.

[5] »Deadline« (neprekoračiva granica) ove bibliografije je konac 1975. godine. Bibliografija pokriva slijedeće teme: teorija i metode općeg cjelobrojnog programiranja, problemi kombinatorne i graf-teoretske optimalizacije, te primjene cjelobrojnog programiranja. Bibliografija sadrži 4.704 različite publikacije od 6.767 autora.

Knjige:

2. T. C. Hu, *Integer Programming and Network Flows*, Reading (Mass.), 1969.
3. T. L. Saaty, *Optimization in Integers and Related Extremal Problems*, New York, 1970. (Ruski prijevod: *Celočislennye metody optimizacii i svjazaannye s nimi ekstremalnye problemy*, Moskva 1973.)
4. H. Greenberg, *Integer Programming*, New York, 1971.
5. A. A. Korbut und J. J. Finkelstein, *Diskrete Optimierung*, Berlin, 1971.
6. D. R. Plane and C. Mc Millan, *Discrete Optimization: Integer Programming and Network Analysis for Management Decision*, Englewood Cliffs, 1971.
7. R. S. Garfinkel and G. L. Nemhauser, *Integer Programming*, New York, 1972.
8. R. E. Burkhard, *Methoden der ganzzahligen Optimierung*, Wien, 1972.
9. P. Brucker, *Ganzzahlige lineare Programmierung mit ökonomischen Anwendungen, Mathematical Systems in Economics* 16, Meisenheim am Glan, 1975.
10. H. M. Salkin, *Integer Programming*, New York, 1975.
11. H. A. Taha, *Integer Programming. Theory, Applications, and Computations*, New York, 1975.
12. Arnold Kaufmann and Arnold Henry-Labordère, *Integer and Mixed Programming: Theory and Applications*, New York, 1977.

Pregledi:

13. E. M. L. Beale, Survey of Integer Programming, *Operational Research Quarterly*, Vol. 16, 1965.
14. M. L. Balinski, Integer Programming: methods, uses, computation, *Management Science*, 12, 1965. 253—313.
15. Y. Y. Finkelstein, Celočislennoe linejnoe programmirovanie, *Ekonomika i matematičeskie metodi*, Tom 2, №. 1, 1966, 115—124.
16. E. L. Lawler and D. E. Wood, Branch and bound methods: a survey, *Operations Research*, Vol. 14, 1966.
17. B. Korda, Celočíslené programování, *Ekonomicko-matematicky obzor*, R. 3, č. 2, 1967, 137—155.
18. M. L. Balinski, On Recent Development in Integer Programming, Proceedings of the International Symposium on Mathematical Programming, Princeton, 1967.
19. M. L. Balinski and K. Spielberg, Methods for Integer Programming: Algebraic, Combinatorial and Enumerative, in J. Aronofsky, *Progress in Operations Research*, Vol. 3, New York, 1969, 195—292.
20. A. M. Geoffrion and R. E. Marsten, Integer programming algorithms; a framework and state of the art survey, *Management Science*, 18, Series A, 1972, 465—491.
21. R. S. Garfinkel and G. L. Nemhauser, A Survey of Integer Programming Emphasizing Computation and Relations among Models, in Hu-Robinson (ed.), *Mathematical Programming*, New York, 1973.

Bibliografije:

22. C. Kastning (edit.), Integer Programming and Related Areas. A Classified Bibliography, *Lecture Notes in Economics and Mathematical Systems, 128*, Berlin, 1976.
23. D. Hausman, Integer Programming and Related Areas. A Classified Bibliography, *Lecture Notes in Economics and Math. Systems*, Vol. 160, Berlin, 1978.