Mészáros Katalin

NÉHÁNY SPECIÁLIS EGÉSZÉRTÉKŰ PROGRAMOZÁSI MODELLSZERKEZET ÉS MEGOLDÁSI ELJÁRÁS

Bevezető

A mindennapi életben felmerülő optimalizálási problémák jelentős része olyan programozási modellhez vezet, ahol a változók csak egész számértéket vehetnek fel. Ez annak következménye, hogy sok közgazdasági feladatnál a vizsgált változók nem oszthatók tetszőlegesen, s ezért a modellalkotásnál egészértékű jellegüket figyelembe kell venni. Ilyen problémák adódnak munkaerő vagy gépterhelés tervezésekor, ahol csak egész számú munkást vagy gépet lehet igénybe venni. Beruházástervezésnél a modell változói egész számú gépeket vagy berendezéseket jelölnek. Bizonyos változók oszthatatlansága miatt jelentkezik a modell egészértékűséget megkövetelő problémaként.

Az egészértékű programozás ma még nem vált az operációkutatási alkalmazások olyan sokoldalúan felhasznált eszközévé, mint a lineáris programozás, azonban intenzív fejlődési szakaszban van. Sok időszerű, érdekes feladat fogalmazható meg az egészértékű programozás nyelvén. Sokat ígérő, gyorsan fejlődő terület, amely viszonylag hosszú múltra tekint vissza. Születési éve 1958. Gomory ekkor fejlesztette ki az egészértékű optimalizálási feladatok megoldására az első egzakt algoritmust. Azóta számos módszer született az egészértékű problémák megoldására, de a kidolgozott módszerek még általában nem elég hatékonyak. E módszerek felhasználása során főleg számítástechnikai, de részben elvi nehézségek is adódnak. Csak közepesnél nem nagyobb méretű feladatok esetén sikerül az egzakt megoldások előállítása. A megoldási nehézségek arányosak az egészértékű változók számával, amely ha egy bizonyos méretet túlhalad, akkor a feladat általában nem oldható meg kifizetődő számítástechnikai ráfordítással. Jelentős sikereket érünk el azonban nagyméretű feladatok esetén, ha nem az egzakt, hanem csak egy közeli optimális megoldást akarunk keresni (annak az egzakt megoldástól való eltérésének becslésével együtt).

Nagy jelentőségük van az ún. 0—1 problémáknak. Itt az egészértékűségi feltétel úgy módosul, hogy a változók csak 0 vagy 1 értékűek lehetnek. Az ilyen változók számítástechnikai kezelése gépi feldolgozás szempontjából egyszerűbb. A 0—1 értékűség nem jelent lényeges megszorítást, mert bármely egészértékű változó felírható 0—1 értékű változók segítségével. Ez esetben azonban megnövekedik a változók száma.

Jelenlegi ismereteink szerint egy egészértékű programozási feladat megoldására a legjobb stratégia nem egy konkrét módszer kiválasztása és annak alkalmazása, hanem a módszerek együttes alkalmazása, kombinálása a feladat jellegétől függően. Esetleg menet közben beavatkozva kell stratégiát változtatni. A helyzet nem olyan kedvező és egyértelmű, mint a lineáris programozási feladatoknál, ahol egy általános algoritmus, a szimplex algoritmus segítségével bármely lineáris programozási feladat megoldható.

A hatékonyság növelésének másik módja a speciális egészértékű problémák szerkezetének vizsgálata, s a sajátos szerkezetnek megfelelő speciális, az általánosnál jóval hatékonyabb algoritmusok kifejlesztése. Jó eredményeket értek el ezen a téren. Hiszen a hátizsák és utazóügynök problémák megoldására rendelkezésre álló algoritmusok sokkal hatékonyabbak az általános algoritmusoknál vagy azok kombinációinál.

A fenti megjegyzések bizonyítják, hogy nem mondhatunk le az egészértékű programozás speciális módszereiről, amelyek egy-egy szűk feladatcsoport sajátosságain alapulnak.

Az egészértékű programozás elméletét ma lényegében a számítási megoldási eljárások elmélete képezi. Ennek okát a számítási megoldásnál felmerülő rendkívüli nehézségekben kereshetjük.

A hátizsák probléma

Azt a tiszta egészértékű feladatot, amelynek pontosan egy feltételi egyenlőtlensége van, hátizsák problémának (angolul knapsack problem) nevezzük. Matematikailag a következő alakban írható fel:

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

$$\sum_{j=1}^{n} a_j x_j \leq k$$

$$0 \leq x_j \leq d_j, \qquad x_j = \text{egész} \qquad a_j,\ c_j,\ d_j,\ k = \text{egész} \qquad (j = 1,2,3,\ldots,n).$$

A hátizsák probléma elnevezés a következő tréfás interpretációból ered: egy kiránduló mielőtt gyalogtúrára megy n fajta különböző tárgyból választhat különböző darabszámot, de legfeljebb d_j-t. A j-edik tárgy súlya a_j, használati értéke pedig c_j. A túrázó legfeljebb k súlyt tud cipelni. Célja, hogy úgy válassza ki a tárgyakat, hogy azok a lehető legnagyobb értéket képviseljék.

Felírható a hátizsák probléma 0—1 változós alakja is:

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

$$\sum_{j=1}^{n} a_j x_j \leq k$$

$$0 \leq x_j \leq 1, \quad x_j = \text{egész} \quad a_j, c_j, k = \text{egész} \quad (j=1,2,3,\ldots,n).$$

A feladat tréfás interpretációja most a következőképpen módosul: egy kiránduló n tárgy közül választhat, mégpedig úgy, hogy az a lehető legnagyobb értéket képviselje. Korlátozó feltételként szerepel, hogy k súlynál többet nem képes magával vinni. A j-edik tárgy súlya a_j, használati értéke pedig c_j.

Az (1)-gyel jelölt hátizsák probléma egyenlőséggel korlátozott változata a következő alakú:

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

$$\sum_{j=1}^{n} a_j x_j \leq k$$

$$0 \leq x_j \leq d_j, \quad x_j = \text{egész} \quad a_j, c_j, d_j, k = \text{egész} \quad (j=1,2,3,\ldots,n).$$

A (3) is hasonlóképpen interpretálható, mint az (1), azzal a különbséggel, hogy most a hátizsákot kötelezően tele kell tölteni. Az (1)-et x_{n+1} kiegészítő változó bevezetésével (3)-má alakíthatjuk, ha $a_{n+1} = 1$ és $c_{n+1} = 0$ értéket vesz fel.

A (2) feladat egy nagyon egyszerű beruházási döntési probléma matematikai modelljéül is szolgálhat, ahol a beruházási keret k pénzegységet tesz ki, az egyes beruházási javaslatok a_j összeget igényelnek és c_j hozamot ígérnek. A (2) feladat optimális megoldása biztosítja a maximális összhozamot a beruházási kereten belül.

A hátizsák probléma még speciálisabbnak tűnő változata a csík leszabás problémája, mivel ebben az esetben a célfüggvény és a feltételi egyenlőtlenség együtthatói azonosak. A feladat pedig így hangzik: van egy k szélességű, meglehetősen hosszú téglalap alakú csíkunk. Ezt kell kevésbé széles $a_1 < a_2 < \ldots < a_n < k$ széles csíkokra szabdalnunk. Az a_j szélességű csíkok iránti igény nem haladja meg az ismert d_j értéket ($j = 1, 2, \ldots, n$). Célunk: elkészíteni a minimális hulladékkal járó kiszabási tervet. Jelölje x_j a leszabásra kerülő a_j szélességű csíkok ismeretlen számát. Ekkor szabástervben szereplő csíkok teljes szélessége $a_1 x_1 + a_2 x_2 + \ldots + a_n x_n + \sum_{j=1}^{n} a_j x_j$-vel egyenlő. A hulladékként jelentkező csík szélessége tehát $k - \sum_{j=1}^{n} a_j x_j$. A következő egészértékű lineáris típusú modellhoz jutottunk:

$$k - \sum_{j=1}^{n} a_j x_j \rightarrow \min$$

$$\sum_{j=1}^{n} a_j x_j \leq k$$

$$0 \leq x_j \leq d_j, \quad x_j = \text{egész} \quad a_j, d_j, k = \text{egész} \quad (j = 1,2,3,\ldots,n).$$

Másképpen írva:

$$\sum_{j=1}^{n} a_j x_j \rightarrow \max$$

$$\sum_{j=1}^{n} a_j x_j \leq k$$

$$0 \leq x_j \leq d_j, \quad x_j = \text{egész} \quad a_j, d_j, k = \text{egész}$$

Ez a feladat matematikailag igen hasonlít az (1)-es problémához.

Mivel a hátizsák probléma egészértékű lineáris típusú programozási feladat, így minden további nélkül megoldható azokkal a módszerekkel, amelyek az ilyen problémák megoldására szolgálnak. Ide tartoznak a metszési módszerek, a kombinatorikus módszerek közül a közvetlen leszámlálás módszere, az implicit leszámlálás módszere, valamint a szétválasztás és korlátozás módszere. Azonban a hátizsák probléma igen speciális szerkezetű, ezért az általános eljárásoknál sokkal hatékonyabb algoritmusok ismeretesek számára. Ezek közül kettőt emelnénk ki. Az egyik a dinamikus programozás segítségével oldja meg a (2)-es típusú hátizsák problémát. Mivel alkalmas transzformációkkal az (1)-es típusú egészértékű változós probléma, mint már említettük, átalakítható (2)-es típusú 0—1 változós problémává, így a dinamikus programozás segítségével az is megoldható. A másik módszer egy gráfelméleti módszer, amely a legrövidebb út keresésére alapozza eljárását. Így a (3)-as típusú probléma oldható meg. Természetesen az (5)-ös típusú probléma is a dinamikus programozással oldható meg.

Mielőtt a hátizsák problémára alkalmaznánk a dinamikus programozás optimalitási elvét, felelevenítjük azt: az optimális döntéssorozat rendelkezik azzal a tulajdonsággal, hogy bármilyen is a kezdeti állapot és kezdeti döntés, a további döntések optimális döntéssorozatot alkotnak az első döntés következtében kialakult helyzetben. Az egészértékű programozási feladat egyben döntési feladat is, hiszen a cél figyelembevételével el kell döntenünk a változók optimális értékét a megadott feltétel teljesülése esetén. Már az előbbiekben láthattuk, hogy a hátizsák probléma a következő alakban írható fel:

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

$$\sum_{j=1}^{n} a_j x_j \leq k$$

$$0 \leq x_j \leq 1, \quad x_j = \text{egész} \quad a_j, c_j, k = \text{egész} \quad (j = 1,2,3, \ldots, n).$$

Amennyiben a $c_j x_j = f_j (x_j)$ jelölést alkalmazzuk a (6) feladathoz n számú változó alapján n döntésből álló döntéssorozatot rendelünk, amelyben ξ tetszőleges változó szerepel, melynek értéke a [o, k] intervallumban bármely egész szám. A döntéssorozat függvényei rekurzív módon állíthatók elő. Általános alakjuk $1 = 1, 2, 3, \ldots, n-1$ esetén

$$\Delta_\varphi(\xi) = \max_{x_\varphi} \left\{ f_\varphi(x_\varphi) + \Delta_{\varphi-\varphi}(\xi - a_\varphi x_\varphi) \right\}.$$

Ha $1 = n$, akkor a $\Delta_1 (\xi)$ függvény értékét elég $\xi = k$-ra kiszámítani. Az n-edik döntésnél nyerjük az n-edik változó, x_n optimális értékét.

Sorrendben visszafelé haladva megkapjuk a többi változó x_{n-1}, x_{n-2}, x_{n-3}, ..., x_2, x_1 optimális értékeit is.

A fenti módon az n-változós hátizsák problémát alkalmas transzformációkkal n fázisos dinamikus programozási feladattá alakítottuk át. Az algoritmus lépéseinek definiálása után programot írtunk rá, hogy konkrét számítástechnikai vizsgálatok alapján állapítsuk meg a hátizsák probléma dinamikus programozással való megoldásának hatékonyságát. Nagyszámú mérést végezve a következő eredményeket kaptuk:

Az egészértékű változók száma	CPU TIME[1]
3	64
4	70
5	84
10	250
15	435
20	460
25	519
30	798

Az elvégzett mérések azt mutatják, hogy a dinamikus programozás segítségével a hátizsák probléma igen hatékonyan megoldható, mert átlagosan rövid idő alatt és kicsi memóriaigénnyel oldja meg az adott típusú feladatot.

A (3)-as típusú probléma megoldása visszavezethető egy gráfon definiált optimalizálási feladatra, pontosabban egy legrövidebb út keresésére a megfelelően definiált gráfban. E módszer alkalmazási feltétele az, hogy minden a_j-nak különbözőnek kell lennie, továbbá az általánosság korlátozása nélkül feltételezhetjük, hogy minden a_j és a k értékei pozitívak (mivel $k < 0$ nyilvánvalóan a feladat megoldhatatlanságát jelentené). A (3)-as típusú problémához rendelt $\Gamma = (V, E)$ gráf a következő $V = \{0, 1, 2, \ldots, k\}$; az i csúcsból a k csúcsba irányított él mutat, ha $k - i = a_j$, ez esetben az (i, k) élhez — c_j élhosszt rendeljük.

Tétel[2]. A (3) lehetséges megoldásai és a gráf azon útjai között, amelyek a 0 és k pontokat kötik össze, olyan kölcsönös és egyértelmű megfeleltetés létesíthető, amely a következő tulajdonsággal rendelkezik: x a legrövidebb út hossza egyenlő a célfüggvény optimális értékének (—1)-szeresével. Az algoritmus lépéseinek részletes kifejezése után elkészült a megfelelő program[3] is, melynek számítástechnikai elemzése a következő eredményeket adta:

Az egészértékű változók száma	CPU TIME
3	63
4	73
5	116
10	626
15	1895
20	3944
25	4837
30	9351

A konkrét számítástechnikai tapasztalatok alapján erről az algoritmusról is elmondható, hogy hatékony, habár valamivel hosszabb idő alatt oldja meg a kitűzött feladatot.

A hátizsák probléma önmagában véve nem jelentős. Jelentőségre a filter-módszer révén tett szert. Viszonylag egyszerű megoldhatósága miatt alkalmazzák az általános egészértékű feladatok megoldásakor, ahol a számításba nem jöhető megoldások kiszűrésére szolgál. Jelentős továbbá az a tény, hogy az egészértékű lineáris típusú problémák széles osztálya visszavezethető egy egyenlőséggel korlátozott hátizsák problémára, azaz a (3)-sal jelölt problémára.

A feladat a következőképpen fogalmazható meg: egy utazó n városban szeretné meglátogatni ügyfeleit, de mindegyik városba csak egyszer akar elmenni. Az 1. városból indul, ide is akar visszatérni. Ismert a városok egymás közti távolsága. Az egész körutat úgy szeretné megtenni, hogy a megtett össztávolság minimális legyen. Ezt a kombinatorikus feladatot az utazó ügynök problémájának v. körutazási problémának (angolul traveling salesman problem) nevezik, amely hosszú időn át állította és állítja még ma is komoly feladat elé a matematikusokat, számítástechnikusokat és gyakorlati szakembereket.

$(n-1)!$ lehetséges sorrend van. Direkt végigszámolni őket lehetetlen lenne, még kis számú n esetén is. Ezért az optimumszámítási, s ezen belül a programozási módszereket kell segítségül hívni.

Matematikailag a feladat többféleképpen is megfogalmazható. Lényege az, hogy magába foglaljon egy hozzárendelési feladatot, valamint a körutazási feltételt. Valahogy sorszámozzuk a városokat. Mindegy, hogy miként. Jelölje c_{ij} az i-edik városnak a j-edik várostól való távolságát. Minden lehetséges utazáshoz rendeljünk hozzá egy változót, amely 0 vagy 1 értéket vesz fel a következő értelmezés szerint:

$$x_{ijt} = \begin{cases} 1, & \text{ha az i-edik városból a j-edikbe megy az} \\ & \text{utazó a t-edik lépésben} \\ 0, & \text{ha nem} \end{cases}$$

$$c_{ij} = c_{ji}$$

$$x_{jjt} = 0$$

$$i, j, t = 1, 2, \ldots, n$$

A változók ilyen értelmezése mellett a célfüggvény:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{t=1}^{n} c_{ij} x_{ijt} = \min$$

alakú.

A feltételek pedig a következőképpen írhatók fel:

$$\sum_{j=1}^{n} x_{1j1} = 1, \qquad x_{1jt} = 0, \qquad t \neq 1$$

$$\sum_{i=1}^{n} x_{i1n} = 1, \qquad x_{i1t} = 0 \qquad t \neq n$$

$$\sum_{t=1}^{n} x_{ijt} \leq 1 \qquad i,j = 1,2,\ldots,n$$

$$\sum_{i=1}^{n} x_{ijt} = 1 \qquad j,t = 1,2,\ldots,n$$

$$\sum_{j=1}^{n} x_{ijt} = 1 \qquad i,t = 1,2,\ldots,n$$

$$\sum_{i=1}^{n} x_{ijt} = \sum_{j'=1}^{n} x_{jj't+1} \qquad j,t = 1,2,\ldots,n$$

A feladat n^3 változót tartalmaz. Így még nem túl nagy n esetén is igen sok változója van. Az általános módszerek csak igen alacsony hatékonysággal oldják meg.

Az utazó ügynök probléma felírható közönséges egészértékű programozási feladatként (nem tiszta 0—1 értékű változókkal) és megoldható például a metszési módszerrel. Ez esetben bevezetjük a következő változókat:

$$x_{ij} = \begin{cases} 1, & \text{ha az ügynök i városból j városba utazik} \\ 0, & \text{ha nem} \end{cases}$$

Egészértékű segédváltozókat képviselnek az $u_1, u_2, \ldots, u_n$ szimbólumokkal jelölt változók, ahol u_i értéke azt jelenti, hogy az i-edik várost hányadiknak látogatta meg az utazó ügynök. Felhasználva a fentiekben bevezetett jelöléseket az utazó ügynök probléma a következő matematikai alakot ölti:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \equiv \min$$

$$\sum_{i=1}^{n} x_{ij} = \sum_{j=1}^{n} x_{ij} = 1 \qquad i \neq j \quad i,j = 1,2,\ldots,n$$

$$u_i - u_j + (n-1)x_{ij} \leq n-2 \qquad i \neq j \qquad i,j = 2,3,\ldots,n$$

Kifogásolható az utazó ügynök probléma fenti matematikai modellbe foglalása, mert kis feladat esetén is nagymértékűvé válik a feladat. A változók száma n^2+n. A feltételek száma pedig $2n+(n-1)(n-2)$. A metszési módszer nem bizonyul alkalmasnak az így megfogalmazott utazó ügynök probléma megoldására. Ez esetben két nyomós kifogás hozható fel ellene:

1. a tipikusan kombinatorikus feladatoknál — ilyen az utazó ügynök probléma is — erőltetett az ilyenfajta megoldás, sőt nem is vezet minden esetben eredményhez;

2. problémát jelent a kerekítési hibákból származó instabilitás.

Az utazó ügynök probléma kérdésfeltevési módja arra jogosít, hogy az egészértékű modellek speciális esetének tekintsük. Ez indokolja azt, hogy a feladat sajátos szerkezetéhez alkalmazkodó speciális módszereket fejlesztettek ki. Ilyen speciális módszer például Held és Karp által kidolgozott, a gyakorlatban jól bevált, a szétválasztás és korlátozás elvén alapuló módszer. Ez a módszer a gráfelmélet segítségével oldja meg az utazó ügynök problémát a következőképpen: az utazó ügynök problémához G_n gráfot rendeljük, melynek csúcsai az $\{1, 2, 3, \ldots, n\}$ halmaz elemei. Az i és j csúcsok között él van, ha a megfelelő városok között van összeköttetés, nincs él, ha nincs köztük összeköttetés. A G_n gráf (i, j) éléhez $c_{ij} = c_{ji}$ súlyt rendeljük. A probléma megoldása egy olyan Hamilton-vonal[4] keresése, amelyhez tartozó súlyok összege minimális. Little, Murty, Sweeney és Karel kidolgoztak egy másik, ugyancsak a szétválasztás és korlátozás elvén alapuló módszert, amellyel a körutazási probléma viszonylag könnyen megoldható, s gépi realizálása sem jelent akadályt.

Az utazó ügynök probléma megoldása további kutatások tárgyát képezi és még ma is publikálnak új algoritmusokat. Gyakorlati alkalmazhatóságát illusztrálja a következő példa: egy gépen n különböző munkadarabot kell megmunkálni. Világos, hogy a megmunkálás össz ideje nem függ a műveletek végzésének sorrendjétől, a gépátállítási idő azonban lényegében attól függ. Amennyiben feleletet adunk arra a kérdésre, hogy milyen sorrendben kell az egyes munkadarabokat megmunkálni, hogy az összes gépállítási idő minimális legyen, akkor tulajdonképpen egy utazó ügynök problémát oldunk meg.

Az utazó ügynök probléma elméleti jelentőségét kiemeli, hogy teljes egészében reprezentálja az általános egészértékű programozási feladatokat. Bebizonyítható, hogy minden egészértékű lineáris típusú programozási feladat alkalmas transzformációkkal visszavezethető az utazó ügynök problémára.[5]

Összefoglalás

Örvendetes, hogy ma már ismertek az egészértékű programozás széles körű alkalmazási lehetőségei, habár a gyakorlati és elméleti nehézségek erősen csökkentik értékét. Kutatások folynak a matematikai programozás terén ezen nehézségek kiküszöbölésére. Figyelmen kívül hagyva a számítási lehetőségeket, a múltban csodálatos egészértékű programozási modellek születtek, amelyek azonban reális ráfordítással nem voltak megoldhatók, s így nem érettek a gyakorlati alkalmazásra. Figyelembe véve az egyes problémák sajátos szerkezetét, speciális módszereket dolgoztak ki, amelyek az általánosoknál hatékonyabban oldják meg a kitűzött feladatot. Habár a hátizsák és utazó ügynök problémák speciálisak, az egészértékű lineáris típusú problémák széles osztálya visszavezethető rájuk alkalmas transzformációkkal. Így a rendelkezésre álló speciális algoritmusok gyakran alkalmazást nyerhetnek magas fokú hatékonyság mellett, s ezzel elértük célunkat: bővítettük az egészértékű modellek alkalmazhatóságát a gyakorlati élet különféle területein.

Jegyzetek

[1] A számítógép központi egysége által elhasznált idő másodpercekben kifejezve.

[2] A bizonyítást [2] tartalmazza.

[3] Mindkét programot az Informatikai és Szervezési Kutatóintézet UNIVAC 1100/20 számítógépére írta a szerző ASCII FORTRAN programozási nyelven.

[4] Hamilton-vonal egy összefüggő gráfban az a kör, amely tartalmazza a gráf összes szögpontját.

[5] Ez a bizonyítás Karp: Reducibility among combinatonal problems című munkájában található meg.

Irodalom

1. Kreko Béla: Optimumszámítás. Közgazdasági és Jogi Könyvkiadó, Budapest, 1972.
2. Forgó Ferenc: Nemkonvex és diszkrét programozás. Közgazdasági és Jogi Könyvkiadó, Budapest, 1978.
3. H. P. Williams: Model Brilding in Mathematical Programming, 1978, John Willey & Sons.
4. Mészáros Katalin: A filter eljárásra épülő algoritmus számítástechnikai elemzése. Magiszteri dolgozat, Szabadka, 1982.
5. Közgazdasági operációkutatási alkalmazások. Közgazdasági és Jogi Könyvkiadó, Budapest, 1976.

Rezime

Nekoliko specifičnih struktura i postupaka rešenja modela celobrojnog programiranja

Danas su već poznate dalekosežne mogućnosti primene celobrojnog programiranja, mada postoje teorijske i praktične poteškoće. U matematičkom programiranju napori se koncentrišu na eliminisanje ovih poteškoća. Neuzimajući u obzir mogućnost rešavanja problema celobrojnog programiranja na računaru u prošlosti su stvoreni divni modeli celobrojnog programiranja, koji su međutim bili nerešivi, a zbog toga nisu podesni za praktičnu primenu. Uzimajući u obzir specifičnu strukturu pojedinih problema razrađeni su specijalni metodi, koji od opštih metoda efikasnije rešavaju te probleme.

U radu je prikazan problem ranca i problem trgovačkog putnika, kao i modeli i metodi za rešavanje. Samostalno ti problemi nisu značajni, međutim široko područje celobrojnog programiranja se može svesti na jedan od tih problema odgovarajućim transformacijama. Na taj način postojeći specijalni algoritmi su primenjeni sa velikom efikasnošću. S tim je naš cilj — proširenje primene modela celobrojnog programiranja — postignut.

Summary

Several Specific Structures and Procedures for Solving Integer Programming

Today we know of the wide-range possibility of applying integer programming, though there are some theoretical and practical difficulties. In mathematical programming efforts are concentrated on eliminating these difficulties. Not taking into account the possibility of solving the problem of integer programming of computers great models nave been made of integer programming in the past which couldn't be solved and for that reason are not suitable for practical usage. Taking into consideration the specific structure of each prob-

lem separately special methods have been worked out, which can solve problems more efficiently than the general methods.

In this paper the problem of a traveling salesman has been presented with the models and methods of solving the problem. Separately these problems are not so significant, but the wide area of integer programming can be brought down to one problem by corresponding transformations. In that way, the existing special algorhythms are applicable with great efficiency. By widening the application of the model of integer programming our goal is attained.