2.6.3. Celobrojno programiranje

Problemi celobrojnog programiranja u opštem slučaju svode se na rešavanje zadataka linearnog i nelinearnog programiranja gde se kao poseban uslov postavlja da brojne vrednosti promenljivih moraju biti celi nenegativni brojevi. Kao posebna klasa zadataka celobrojnog programiranja izdvajaju se zadaci kombinatornog karaktera, gde promenljive u matematičkom modelu mogu uzeti samo vrednost "1" ili "0".

Problem celobrojnosti promenljivih nameće se vrlo često kod određene klase realnih problema. U sledećim primerima biće data opšta formulacija nekih tipičnih zadataka celobrojnog programiranja.

53. Zadatak

Preduzeće proizvodi n različitih tipova mašina. Za realizaciju proizvodnje potrebno je m različitih vrsta resursa sa kojima se raspolaže u ograničenim količinama $a_1, a_2, \ldots, a_i, \ldots\ldots, a_m$. Poznato je:

a_{ij} – normativ utroška i-tog resursa za proizvodnju jedne j-te mašine,

c_j – dobit ostvarena isporukom jedne mašine j-tog tipa ($j = 1, 2, \ldots, n$).

Pretpostavlja se da na kraju planskog perioda nije poželjno imati nedovršenih mašina.

Formirati matematički model problema, pod uslovom da preduzeće želi ostvariti maksimalnu dobit od proizvodnje.

Rešenje. Matematički model sačinjava funkcija cilja koja definiše ukupnu dobit koja će se ostvariti u planskom periodu,

$$F(X) = \sum_{j=1}^{n} c_j x_j = c_1 x_1 + c_2 x_2 + \ldots + c_j x_j + \ldots + c_n x_n$$

koja zavisi od raspoloživih resursa, tj. uslovi u kojima se ostvaruje proizvodnja definisani su sistemom jednačina

$$\sum_{j=1}^{n} a_{ij} x_j = a_i, \qquad i = 1, 2, \ldots, m.$$

$$x_j \geqslant 0, \qquad j = 1, 2, \ldots, n.$$

Brojne vrednosti promenljivih mogu biti samo celi brojevi.

54. Zadatak

Proizvodni proces karakteriše se sa n različitih vrsta poslova ($j = 1, 2, \ldots, n$), odnosno radnih operacija. U planskom periodu kvantitativne mere za pojedine operacije su $b_1, b_2, \ldots, b_j, \ldots, b_n$. Takođe, zadata je matrica $\|r_{ij}\|$ gde je r_{ij} – produktivnost i-tog tipa mašine na j-tom poslu, zatim, matrica $\|c_{ij}\|$, gde su c_{ij} – troškovi obavljanja j-tog posla na mašini i-tog tipa, kao i cene c_i jedne mašine i-tog tipa.

Formirati matematički model zadatka iznalaženja optimalnog mašinskog parka, (tj. brojni iznos mašina za svaki tip), kao i određivanja optimalne raspodele mašina na određene poslove pod uslovom da se ostvare minimalni ukupni troškovi realizacije proizvodnog procesa.

Rešenje. Matematički model problema formira se tako što se uvode promenljive:

y_i – ukupan broj mašina i-tog tipa, $i = 1, 2, \ldots, m$.

x_{ij} – broj mašina i-tog tipa koje se koriste za izvršenje j-te vrste posla.

Očigledno je da promenljive y_i moraju biti celi brojevi, dok promenljive x_{ij} ne moraju ako se produktivnost mašina r_{ij} nije deljiva bez ostatka sa odgovarajućim vrednostima b_j.

Polazeći od navedenih oznaka ukupne troškove u planskom periodu možemo definisati sledećim izrazom

$$(\min) F(X, Y) = \sum_{i=1}^{m} c_i y_i + \sum_{i=1}^{m} \sum_{j=1}^{n} r_{ij} x_{ij}.$$

Potrebne količine mašina definisane su sledećim sistemom jednačina

$$\sum_{i=1}^{m} r_{ij} x_{ij} = b_j, \qquad (j = 1, 2, \ldots, n)$$

$$\sum_{j=1}^{n} x_{ij} = y_i, \qquad (i = 1, 2, \ldots, m),$$

gde su promenljive $x_{ij} \geq 0$, dok su $y_i \geq 0$ i celobrojne vrednosti.

Zadatak se sastoji u tome da se odrede takve celobrojne vrednosti promenljivih y_i i vrednosti promenljivih x_{ij} za koje će funkcija $F(X, Y)$ imati minimalnu vrednost, a ograničenja neće biti narušena.

55. Zadatak

Planskom organu pomorskog transporta stoje na raspolaganju m različitih tipova brodova u količinama od $q_1, q_2, \ldots, q_i, \ldots, q_m$. Na svakom od brodova postoji n različitih prostora za utovar robe ($j = 1, 2, \ldots, n$), čiji su kapaciteti u odnosu na vrstu robe d_{ijk} (podrazumeva se da za određene vrednosti indeksa i – j veličina d_{ijk} može biti jednaka nuli). Potrebno je prevesti r različitih vrsta robe u količinama $b_1, b_2, \ldots, b_k, \ldots\ldots, b_r$.

Formirati matematički model zadataka određivanja optimalnog sastava brodova, ako su troškovi eksploatacije jednog broda i-tog tipa c_i.

Rešenje. Optimalni sastav brodova formira se na bazi minimizacije ukupnih troškova eksploatacije, tj. treba odrediti brojne vrednosti skupa promenljivih y_1, $y_2, \ldots, y_i, \ldots, y_m$, za koje funkcija

$$F(Y) = \sum_{i=1}^{m} c_i y_i$$

ima minimalnu vrednost.

Ako sa x_{jk} označimo broj jedinica robe k-te vrste utovarene u j-ti prostor, tada se ograničenja mogu definisati na sledeći način

$$\sum_{i=1}^{m} d_{ijk} y_i - x_{jk} \geqslant 0,$$
$$\sum_{j=1}^{n} x_{jk} = b_k$$

$$0 \leqslant y_i \leqslant q_i \quad \text{i} \quad x_{jk} \geqslant 0$$

gde indeksi uzimaju vrednosti: i = 1, 2, . . . , m; j = 1, 2, . . . , n; i k=1,2,. . . ,r.

Promenljive u modelu y_i i x_{jk} moraju biti celi brojevi

56. **Zadatak**

Problem planiranja upotrebe transportnih sredstava može se definisati na sledeći način. Postoji m različitih vrsta transportnih sredstava, koja se mogu koristiti u ograničenom broju časova, $a_1, a_2, \ldots, a_i, \ldots, a_m$. Ova sredstva treba upotrebiti na n različitih marš-ruta, pri čemu se na svakoj marš-ruti mora ostvariti određeni broj putovanja, što se definiše vrednostima $b_1, b_2, \ldots, b_j, \ldots, b_n$. Za izvršenje jednog putovanja i-tom mašinom na j-toj marš-ruti potrebno je t_{ij} časova, uz troškove c_{ij}.

Formirati matematički model celobrojnog programiranja, ako se želi optimalno raspodeliti transportna sredstva tako da troškovi transporta budu minimalni.

Rešenje. Ako se sa x_{ij} označi broj izvršenih putovanja od strane i-tog transportnog sredstva upotrebljenog na j-toj marš-ruti, tada će ukupni troškovi transporta biti definisani izrazom,

$$F(X) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}.$$

Pri upotrebi transportnih sredstava moraju biti zadovoljena ograničenja

$$\sum_{j=1}^{n} t_{ij} x_{ij} < a_i, \qquad (i = 1, 2, \ldots, m).$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \qquad (j = 1, 2, \ldots, n)$$

i opšti uslov da sve promenljive u modelu moraju biti veće od nule i celobrojne.

2.6.4. Metode rešavanja zadataka celobrojnog programiranja

Za rešavanje zadataka celobrojnog programiranja postoje različite metode koje baziraju na svojstvima pojedinih klasa zadataka celobrojnog programiranja.

Metoda odsečaka – bazira na sukcesivnom rešavanju konačnog broja specijalno formiranih zadataka linearnog programiranja. Svaki od zadataka formuliše se na osnovi prethodnog, dodajući već postojećim ograničenjima novo linearno ograničenje – "odsečak". Ovaj metod biće ilustrovan na konkretnim primerima.

57. **Zadatak**

Zadatak celobrojnog programiranja svodi se na matematički model sa funkcijom cilja

$$(\max) F(X) = x_1 + 4x_2$$

i uslovima definisanim linearnim ograničenjima,

$$-x_1 + 2x_2 + x_3 = 2$$

$$3x_1 + 2x_2 + x_4 = 6$$

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

Potrebno je naći celobrojne vrednosti promenljivih x_1, x_2, x_3 i x_4 koje obezbeđuju maksimum funkcije F (X), a istovremeno zadovoljavaju i ograničenja, definisana sistemom jednačina.

Rešenje. Algoritam za rešavanje postavljenog zadatka definisan je sa tri nezavisna koraka.

1. korak. U ovom koraku zanemaruje se uslov celobrojnosti i rešava se zadatak linearnog programiranja L_0.

Primenom simpleks metode dobija se traženo rešenje, koje je prikazano tabelama 1, 2 i 3.

Tabela 1.

C_0	B	X_0	1	4	0	0
			x_1	x_2	x_3	x_4
0	x_3	2	−1	$\boxed{2}$	1	0
0	x_4	6	3	2	0	1
$F_j - c_j$		0	−1	−4	0	0

Tabela 2.

C_0	B	X_0	1	4	0	0
			x_1	x_2	x_3	x_4
4	x_2	1	−1/2	1	1/2	0
0	x_4	4	$\boxed{4}$	0	−1	1
$F_j - c_j$		4	−3	0	2	0

Tabela 3.

C_0	X	X_0	1	4	0	0
			x_1	x_2	x_3	x_4
4	x_2	3/2	0	1	3/8	1/8
1	x_1	1	1	0	−1/4	1/4
$F_j - c_j$		7	0	0	5/4	3/4

Dobijeno rešenje zadatka L_0 je

$$X^0 = (1; \frac{3}{2}; 0; 0) \quad \text{i} \quad F(X) = 7.$$

Kako ovo rešenje nije celobrojno, prelazi se na drugi korak.

2. korak. Na osnovu poslednje simpleks tabele u 1. koraku formira se "odsečak" – novo linearno ograničenje sa kojim se proširuje simpleks tabela i formira novi zadatak linearnog programiranja L_1.

U opštem slučaju novo ograničenje se formira na sledeći način. Ako se uvedu oznake:

$\{a\}$ – decimalni deo broja a,

$[a]$ – najveći ceo broj manji ili jednak datom broju a,

k – indeksi promenljivih koje u poslednjoj simpleks tabeli ne pripadaju bazi,

s – broj reda u poslednjoj simpleks tabeli sa najvećom vrednošću a_{so},

tada se novo ograničenje (Gomory-jev odsečak) može pisati u obliku

$$\{a_{so}\} - \sum_k \{a_{sk}\} \cdot x_k \leqslant 0.$$

U gornjem primeru je:

$$\{a_{so}\} = \{\frac{3}{2}\} = \frac{1}{2}, \qquad k = 3 \text{ i } 4,$$

$$\{a_{s3}\} = \{\frac{3}{8}\} = \frac{3}{8} \text{ i } \{a_{s4}\} = \{\frac{1}{8}\} = \frac{1}{8}.$$

Prema tome, ograničenje je oblika

$$\frac{1}{2} - \frac{3}{8}x_3 - \frac{1}{8}x_4 \leqslant 0,$$

a uvođenjem nove izravnavajuće promenljive dobija se

$$\frac{3}{8}x_3 + \frac{1}{8}x_4 - u_1 = \frac{1}{2},$$

ili ako jednačinu pomnožimo sa -1 dobija se izraz

$$-\frac{3}{8}x_3 - \frac{1}{8}x_4 + u_1 = -\frac{1}{2}.$$

3. korak. Dodajući novo dobijeno ograničenje u poslednju simpleks tabelu dobija se početna simpleks tabela zadatka L_1.

Tabela 4.

C	B	X_0	1	4	0	0	0
			x_1	x_2	x_3	x_4	u_1
4	x_2	3/2	0	1	3/8	1/8	0
1	x_1	1	1	0	−1/4	1/4	0
0	u_1	−1/2	0	0	−3/8	−1/8	1
$F_j - c_j$		7	0	0	5/4	3/4	0

Tabela 4 ne sadrži moguće rešenje problema (jer je $u_1 = -\frac{1}{2}$). Cilj dalje transformacije je dobijanje početnog mogućeg rešenja. Zato se u redu u_1 bira kolona r sa najvećim negativnim brojem i proverava se da li se dobija

$$\min_i \left\{ \frac{b_i}{a_{ir}} \right\}$$

Za ono i koje odgovara negativnoj promenljivoj u_1. Ako je uslov ispunjen tada u novu bazu ulazi promenljiva x_r. U slučaju da ovo nije ispunjeno ide se na novu kolonu sa negativnim brojem.

U ovom primeru promenljiva x_3 ulazi u sledeće bazno rešenje, koje je istovremeno i moguće rešenje (tabela 5).

Tabela 5.

C_0	B	X_0	1	4	0	0	0
			x_1	x_2	x_3	x_4	u_1
4	x_2	1	0	1	0	0	1
1	x_1	4/3	1	0	0	1/3	−2/3
0	x_3	4/3	0	0	1	1/3	−8/3
$F_j - c_j$		16/3	0	0	0	1/3	10/3

U tabeli 5 dobijeno je optimalno rešenje problema koje nije celobrojno. Prema tome, vraćamo se na korak 2.

2. korak. Formira se novo ograničenje i novi zadatak L_2. Ograničenje se formira na bazi reda x_1, otuda je

$$\{\frac{4}{3}\} - \{\frac{1}{3}\} \cdot x_4 - \{-\frac{2}{3}\} \cdot u_1 \leq 0.$$

Kako je $\{\frac{4}{3}\} = \frac{1}{3}$, $\{\frac{1}{3}\} = \frac{1}{3}$ i

$$\{-\frac{2}{3}\} = -\frac{2}{3} - \{-\frac{2}{3}\} = -\frac{2}{3} -(-1) = \frac{1}{3}$$

to će ograničenje biti sledećeg oblika

$$\frac{1}{3} - \frac{1}{3}x_4 - \frac{1}{3}u_1 + u_2 = 0,$$

gde je u_2 nova izravnavajuća promenljiva. Prema tome, konačna forma ograničenja je

$$-\frac{1}{3}x_4 - \frac{1}{3}u_1 + u_2 = -\frac{1}{3}.$$

Napomena. Pri određivanju decimalnog dela negativnog mešovitog broja treba imati u vidu opšti izraz

$$\{-a\} = -a - [-a].$$

3. korak. Dodajući napred formirano ograničenje iz prethodnog koraka u poslednju simpleks tabelu (tabela 5) dobija se početna simpleks tabela za ovaj korak.

Tabela 6.

C_0	B	X_0	1	4	0	0	0	0
			x_1	x_2	x_3	x_4	u_1	u_2
4	x_2	1	0	1	0	0	0	0
1	x_1	4/3	1	0	0	1/3	−2/3	0
0	x_3	4/3	0	0	1	1/3	−8/3	0
0	u_2	−1/3	0	0	0	$\boxed{-1/3}$	−1/3	1
$F_j - c_j$		16/3	0	0	0	1/3	10/3	0

Bazno rešenje u tabeli 6 nije moguće, otuda se za kolone x_4 određuje

$$(\min)\left\{\begin{array}{l} \frac{4}{3} : \frac{1}{3} = 4 \\ \frac{4}{3} : \frac{1}{3} = 4 \\ -\frac{1}{3} : -\frac{1}{3} = 1 \end{array}\right\} = 1.$$

Kako minimum odgovara redu u_2 to se u sledećoj iteraciji može dobiti moguće rešenje zadatka, koje je dato u tabeli 7.

Tabela 7.

C_0	B	X_0	1	4	0	0	0	0
			x_1	x_2	x_3	x_4	u_1	u_2
4	x_2	1	0	1	0	0	1	0
1	x_1	1	1	0	0	0	–1	1
0	x_3	1	0	0	1	0	–3	1
0	x_4	1	0	0	0	1	1	–3
$F_j - c_j$		5	0	0	0	0	3	1

Novo dobijeno moguće rešenje je optimalno u smislu kriterijuma simpleks metoda. Takođe, dobijeno rešenje je optimalno za napred postavljeni zadatak celobrojnog programiranja, jer su vrednosti bazni promenljivih celobrojne, tj.

$$x_1 = 1, \quad x_2 = 1, \quad x_3 = 1 \text{ i } x_4 = 1.$$

Vrednost funkcije F(X) je

$$\max F(X) = 5.$$

58. Zadatak

Naći rešenje zadataka celobrojnog programiranja:

a) $(\max) F(X) = 10 + 2x_1 + 2x_2,$

$2x_1 + x_2 + x_3 = 5$

$2x_1 + 3x_2 + x_4 = 9.$

b) $(\max) F(X) = 3x_1 + 4x_2$

$3x_1 + 2x_2 \leqslant 8$

$x_1 + 4x_2 \leqslant 10.$

c) $(\max) F(X) = 21x_1 + 11x_2$

$7x_1 + 4x_2 + x_3 = 13,$

gde sve promenljive u modelima moraju biti veće ili jednake nuli i celi brojevi.

Rešenje. Primenom metoda odsečaka dobijaju se rešenja:
a) Zadatak ima višestruko optimalno celobrojno rešenje:

1. $x_1 = 0, \quad x_2 = 3, \quad x_3 = 2, \quad x_4 = 0.$

2. $x_1 = 1, \quad x_2 = 2, \quad x_3 = 1, \quad x_4 = 1.$

3. $x_1 = 2, \quad x_2 = 1, \quad x_3 = 0, \quad x_4 = 2.$

$(\max) F(X) = 16.$

b) Optimalno celobrojno rešenje je:

$x_1 = 1, \quad x_2 = 2$ i $(\max) F(X) = 11.$

c) Optimalno celobrojno rešenje je:

$x_1 = 0, \quad x_2 = 3, \quad x_3 = 1,$

$(\max) F(X) = 33.$

59. Zadatak

Rešiti zadatak celobrojnog programiranja, ako se traži

$$(\max) F(X) = 6x_1 + x_2$$

pod uslovom da je

$$-2,9x_1 + 6x_2 \leq 17,4$$

$$3x_1 - x_2 \leq 1,$$

a promenljive $x_1 \geq 0$ i $x_2 \geq 0$ i celi brojevi.

Rešenje. Ako se uvedu izravnavajuće promenljive matematički model se svodi na oblik

$$(\max) F(X) = 6x_1 + x_2$$

$$-2,9x_1 + 6x_2 + x_3 = 17,4$$

$$3x_1 - x_2 + x_4 = 1$$

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

Iz dobijenog modela vidi se da promenljiva x_3 ne mora biti celobrojna da bi promenljive x_1 i x_2 bile celobrojne. U ovom slučaju postupak rešavanja zadatka ima neke specifičnosti u odnosu na prethodni postupak rešavanja zadataka celobrojnog programiranja.

1. korak. U ovom koraku se rešava zadatak zanemarujući uslov celobrojnosti. Rešenje problema dato je u sledećim simpleks tabelama.

Tabela 1.

C_0	B	X_0	6	1	0	0
			x_1	x_2	x_3	x_4
0	x_3	17,4	−2,9	6	1	0
0	x_4	1	3	−1	0	1
$F_j - c_j$		0	−6	−1	0	0

Tabela 2.

C_o	X	X_o	6	1	0	0
			x_1	x_2	x_3	x_4
0	x_3	$\frac{55,1}{3}$	0	$\frac{15,1}{3}$	1	$\frac{2,9}{3}$
6	x_1	$\frac{1}{3}$	1	$-\frac{1}{3}$	0	$\frac{1}{3}$
$F_j - c_j$		2	0	−3	0	2

Tabela 3.

C_o	B	X_o	6	1	0	0
			x_1	x_2	x_3	x_4
1	x_2	3,649	0	1	0,198	0,192
6	x_1	1,55	1	0	0,066	0,397
$F_j - c_j$		12,947	0	0	0,594	2,576

U poslednjoj tabeli (3), dobijeno je optimalno rešenje. Međutim, promenljive nisu celobrojne.

2. korak U ovom koraku definiše se drugi Gomory-jev presek, kao novo ograničenje u modelu

$$\{a_{so}\} - \sum_{k=1}^{m} \alpha_{sk}\, x_k \leqslant 0,$$

gde su α_{sk} – koeficijenti koji se izračunavaju na osnovu sledećih izraza:

1) Za promenljive koje ne moraju biti celobrojne

$$\alpha_{sk} = \begin{cases} a_{sk}, \text{ ako je } a_{sk} \geqslant 0, \\ \dfrac{\{a_{so}\}}{1 - \{a_{so}\}} \cdot |a_{sk}|, \text{ za } a_{sk} < 0; \end{cases}$$

2) Za promenljive koje moraju biti celobrojne

$$\alpha_{sk} = \begin{cases} \{a_{sk}\}, & \text{ako je } \{a_{sk}\} < \{a_{so}\}, \\ \dfrac{\{a_{so}\}}{1-\{a_{so}\}}(1-\{a_{sk}\}), & \text{ako je } \{a_{sk}\} > \{a_{so}\}. \end{cases}$$

Kako je

$$\{a_{so}\} = \{3{,}649\} = 0{,}649$$

$$\alpha_{13} = 0{,}198 \quad \text{i} \quad \alpha_{14} = 0{,}192,$$

to je ograničenje oblika

$$0{,}649 - 0{,}198\,x_3 - 0{,}192\,x_4 + u_1 = 0,$$

odnosno, možemo pisati da je

$$-0{,}198\,x_3 - 0{,}192\,x_4 + u_1 = -0{,}649.$$

3. korak. Dodajući novo dobijeno ograničenje u poslednju simpleks tabelu dobija se nova početna simpleks tabela zadataka L_1 .

Tabela 4.

C_0	B	X_0	6	1	0	0	0
			x_1	x_2	x_3	x_4	u_1
1	x_2	3,649	0	1	0,198	0.192	0
6	x_1	1,55	1	0	0,066	0,397	0
0	u_1	–0,649	0	0	–0,198	–0,192	1
$F_j - c_j$		12,947	0	0	0,594	2,576	0

Kako je

$$(\min)\begin{bmatrix} \dfrac{3{,}649}{0{,}198} = 18{,}43 \\[2ex] \dfrac{1{,}55}{0{,}066} = 23{,}48 \\[2ex] \dfrac{-0{,}649}{-0{,}198} = 3{,}28 \end{bmatrix} = 3{,}28,$$

$$\min\begin{bmatrix} \dfrac{3{,}649}{0{,}192} = 19{,}005 \\[2ex] \dfrac{1{,}55}{0{,}397} = 3{,}9 \\[2ex] \dfrac{-0{,}649}{-0{,}192} = 3{,}38 \end{bmatrix} = 3{,}38\,,$$

to se uvođenjem promenljive x_3 ili x_4 u bazu može dobiti moguće rešenje. Novo moguće rešenje je prikazano u sledećoj simpleks tabeli (tabela 5).

Tabela 5.

C_o	B	X_o	6	1	0	0	0
			x_1	x_2	x_3	x_4	u_1
1	x_2	3	0	1	0	0	1
6	x_1	1,33	1	0	0	0,333	0,333
0	x_3	3,28	0	0	1	0,97	–5,05
$F_j - c_j$		10,99	0	0	0	2	3

Dobijeno rešenje je optimalno. Međutim, kako promenljiva x_1 nije celobrojna, vraćamo se na korak 2.

2. korak. Formira se novo ograničenje i novi zadatak L_2. Ograničenje se formira na osnovu reda x_1 u prethodnoj tabeli 5. Otuda se dobija

$$0{,}33 - 0{,}333\,x_4 - 0{,}333\,u_1 + u_2 = 0,$$

odnosno, ako se jednačina sredi

$$-0{,}333x_4 - 0{,}333u_1 + u_2 = -0{,}33.$$

3. korak. Proširena simpleks tabela je

Tabela 6.

C_0	B	X_0	6	1	0	0	0	0
			x_1	x_2	x_3	x_4	u_1	u_2
1	x_2	3	0	1	0	0	1	0
6	x_1	1,33	1	0	0	0,333	0,333	0
0	x_3	3,28	0	0	1	0,97	−5,05	0
0	u_2	−0,33	0	0	0	−0,333	−0,333	1
$F_j - c_j$		10,99	0	0	0	2	2	0

Kako je

$$\min \begin{bmatrix} \dfrac{1{,}33}{0{,}333} = 3{,}99 \\ \dfrac{3{,}28}{0{,}97} = 3{,}38 \\ \dfrac{-0{,}33}{-0{,}333} = 1 \end{bmatrix} = 1$$

to se moguće rešenje može dobiti uvođenjem u bazu promenljive x_4, kao što je to pokazano u tabeli 7.

Tabela 7.

C_0	B	X_0	6	1	0	0	0	0
			x_1	x_2	x_3	x_4	u_1	u_2
1	x_2	3	0	1	0	0	1	0
6	x_1	1	1	0	0	0	0	1
0	x_3	2,31	0	0	1	0	−6,02	−15,15
0	x_4	1	0	0	0	1	1	−3
$F_j - c_j$		9	0	0	0	0	1	6

U tabeli 7 dobijeno je optimalno rešenje zadataka celobrojnog programiranja,

$$x_1 = 1, \quad x_2 = 3, \quad x_4 = 1 \quad \text{i} \quad (\max) F(X) = 9.$$

Za promenljivu x_3 nije se tražilo da ima celobrojnu vrednost.

60. Zadatak

Rešiti zadatke celobrojnog programiranja definisane matematičkim modelima:

a) $(\max) F(X) = x_1 + 8x_2$

$3x_1 + x_2 \leqslant 9$

$0{,}16x_1 + x_2 \leqslant 1{,}9$

$x_1 \geqslant 0, \quad x_2 \geqslant 0$ i celi brojevi.

b) $(\max) F(X) = 0{,}25x_1 + x_2$

$0{,}5x_1 + x_2 \leqslant 1{,}75$

$x_1 + 0{,}3x_2 \leqslant 1{,}5$

$x_1 \geqslant 0, \quad x_2 \geqslant 0$ i celi brojevi.

c) $(\max) F(X) = 8x_1 + 6x_2$

$3x_1 + 5x_2 + x_3 = 11$

$4x_1 + x_2 + x_4 = 8$

$x_1 \geqslant 0$ i celobrojno, $x_2 \geqslant 0$.

Rešenje. Primenom napred izloženog metoda celobrojnog programiranja (zadatak 58.) dobijaju se rešenja:

a) $x_1 = 2$, $x_2 = 1$ i $(\max) F(X) = 10$. Izravnavajuće promenljive su: $x_3 = 2$ i $x_4 = 0{,}58$;

b) $x_1 = 1$, $x_2 = 1$ i $(\max) F(X) = 1{,}25$. Izravnavajuće promenljive su: $x_3 = 0{,}25$ i $x_4 = 0{,}2$;

c) $x_1 = 1$, $x_2 = 1{,}6$ i $(\max) F(X) = 17{,}62$.

General Cutting Planes

Consider the following integer program:

$$\begin{array}{lrcrcl} \text{Maximize} & 7x_1 & + & 9x_2 & & \\ \text{subject to} & -x_1 & + & 3x_2 & \leq & 6 \\ & 7x_1 & + & x_2 & \leq & 35 \\ & x_1, & & x_2 & \geq & 0 \text{ integer.} \end{array}$$

If we ignore integrality, we get the following optimal tableau (with the updated columns and reduced costs shown for nonbasic variables):

Variable	x_1	x_2	s_1	s_2	$-z$	RHS
x_2	0	1	7/22	1/22	0	7/2
x_1	1	0	-1/22	3/22	0	9/2
$-z$	0	0	28/11	15/11	1	63

Let's look at the first constraint:

$$x_2 + 7/22 s_1 + 1/22 s_2 = 7/2$$

We can manipulate this to put all of the integer parts on the left side, and all the fractional parts on the right to get:

$$x_2 - 3 = 1/2 - 7/22 s_1 - 1/22 s_2$$

Now, note that the left hand side consists only of integers, so the right hand side must add up to an integer. Which integer can it be? Well, it consists of some positive fraction minus a series of positive values. Therefore, the right hand side can only be $0, -1, -2, \ldots$; it cannot be a positive value. Therefore, we have derived the following constraint:

$$1/2 - 7/22 s_1 - 1/22 s_2 \leq 0.$$

This constraint is satisfied by every feasible integer solution to our original problem. But, in our current solution, s_1 and s_2 both equal 0, which is infeasible to the above constraint. This means the above constraint is a cut, called the *Gomory cut* after its discoverer. We can now add this constraint to the linear program and be guaranteed to find a different solution, one that might be integer.

We can also generate a cut from the other constraint. Here we have to be careful to get the signs right:

$$x_1 - 1/22 s_1 + 3/22 s_2 = 9/2$$

$$x_1 + (-1 + 21/22) s_1 + 3/22 s_2 = 4 + 1/2$$

$$x_1 - s_1 - 4 = 1/2 - 21/22 s_1 - 3/22 s_2$$

gives the constraint

$$1/2 - 21/22 s_1 - 3/22 s_2 \leq 0.$$

In general, let $\lfloor a \rfloor$ be defined as the largest integer less than or equal to *a*. For example, $\lfloor 3.9 \rfloor = 3$, $\lfloor 5 \rfloor = 5$, and $\lfloor -1.3 \rfloor = -2$.

If we have a constraint

$$x_k + \sum a_i x_i = b$$

with *b* not an integer, we can write each $a_i = \lfloor a_i \rfloor + a_i'$, for some $0 \leq a_i' < 1$, and $b = \lfloor b \rfloor + b'$ for some 0 ;*SPMlt*; *b*' ;*SPMlt*; 1. Using the same steps we get:

$$x_k + \sum \lfloor a_i \rfloor x_i - \lfloor b \rfloor = b' - \sum a_i' x_i$$

to get the cut

$$b' - \sum a_i' x_i \leq 0.$$

This cut can then be added to the linear program and the problem resolved. The problem is guaranteed not to get the same solution.

This method can be shown to guarantee finding the optimal integer solution. There are a couple of disadvantages:

1. Round-off error can cause great difficulties: Is that 3.000000001 really a 3, or should I generate a cut? If I make the wrong decision I could either cut off a feasible solution (if it is really a 3 but I generate a cut) or I could end up with an infeasible solution (if it is not a 3 but I treat it as one).
2. The number of constraints that are generated can be enormous. Just like branch and bound can generate a huge number of subproblems, this technique can generate a huge number of constraints.

The combination of these makes this cutting plane technique impractical by itself. Recently however, more powerful techniques have been discovered for special problem structure. This is the subject of the next section.