Rješavanje problema optimalizacije metodama cjelobrojnog programiranja

Pavlović, Marko Fabijan

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:266726

Rights / Prava: Attribution-NonCommercial-ShareAlike 3.0 Unported/Imenovanje-Nekomercijalno-Dijeli pod istim uvjetima 3.0

Download date / Datum preuzimanja: **2024-03-29**

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

zir.nsk.hr

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

VARAŽDIN

Marko Fabijan Pavlović

RJEŠAVANJE PROBLEMA OPTIMALIZACIJE METODAMA CJELOBROJNOG PROGRAMIRANJA

ZAVRŠNI RAD

Varaždin, 2018.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

VARAŽDIN

Marko Fabijan Pavlović

Matični broj: 42012/13-R

Studij: Poslovni sustavi

RJEŠAVANJE PROBLEMA OPTIMALIZACIJE METODAMA CJELOBROJNOG PROGRAMIRANJA

ZAVRŠNI RAD

Mentor:
doc. dr. sc. Nikolina Žajdela Hrustek

Varaždin, siječanj 2018.

Sadržaj

1. Uvod 1
2. Cjelobrojno programiranje 2
3. Metode za cjelobrojno programiranje 4
3.1. Metoda cjelobrojnih formi 4
3.2. Metoda grananja i ograđivanja 13
4. Problem fiksnih troškova 22
5. Problem ranca 30
6. Problem optimalne investicijske odluke 34
7. Zaključak 39
8. Literatura 40
9. Prilog – aplikacija "Zero-One Knapsack" 41
9.1. Instalacija aplikacije 41
9.2. Korištenje aplikacije 42
9.3. Korištene tehnologije 45

1. Uvod

Moderno poslovanje nemoguće je zamisliti bez korištenja matematičkih modela i metoda. Kako bi rezultati poslovanja bili profitabilni, potrebno je točno i precizno odrediti ulaganja i tijek poslovanja. Većina kritičnih odluka izračunava se linearnim programiranjem, koje uglavnom daje necjelobrojne, odnosno decimalne rezultate, osim ako nije igrom slučaja pogođena kombinacija parametara koja bi dala cjelobrojne iznose.

Tema ovoga rada je cjelobrojno programiranje, podvrsta linearnog programiranja koja sprječava dobivanje decimalnih rezultata. U početku se rješava kao i linearno programiranje, dok se u kasnijim koracima dodaju ograničenja koja izbacuju moguće decimalne rezultate. U radu su teorijski obrađene dvije metode za rješavanje cjelobrojnog programiranja, metoda cjelobrojnih formi i metoda grananja i ograđivanja, te su objašnjene na primjeru. Potom su opisani osnovni problemi za čije je rješavanje cjelobrojno programiranje potrebno, te je za svaki problem prikazan postupak rješavanja jednom od dvije metode. Tri osnovna problema su problem fiksnih troškova, problem ranca i problem optimalne investicijske odluke. Problem fiksnih troškova i problem optimalne investicije javljaju se kod upravljanja financijama, kad je potrebno ulagati novac radi ostvarivanja dodane vrijednosti. Problem ranca predstavlja problem punjenja kapaciteta, odnosno stavljanja stvari u neko spremište. Na taj problem moguće je naići u svakodnevnom životu, stoga je napravljena aplikacija Zero-One Knapsack, koja predstavlja kalkulator za problem ranca.

Ovaj rad predstavlja uvod u cjelobrojno programiranje, njegovu namjenu i svrhu kroz probleme i slučajeve u kojima se primjenjuje. Kako bi se što bolje objasnile metode i problemi, korištena je stručna literatura i akademski članci iz toga područja, kako na hrvatskom tako i na engleskom jeziku. Također je za problem optimalne investicije korištena i ekonomska literatura.

Kako je već spomenuto, na kraju rada nalazi se aplikacija naziva Zero-One Knapsack, koja služi za rješavanje problema ranca. Nakon što se unesu potrebni parametri, aplikacija ispisuje vrijednost elemenata u rancu, te pojedinačno sve elemente koji se nalaze u rancu. Aplikacija će biti detaljno objašnjena u posljednjem poglavlju.

2. Cjelobrojno programiranje

Razvoj cjelobrojnog programiranja započinje 1957. godine. Te su se godine Harry Markowitz i Alan S. Manne počeli znatnije baviti linearnim problemom koji zahtijeva cjelobrojna rješenja, odnosno počeli su razmatrati opći problem cjelobrojnog programiranja, ali nisu imali opću metodu za takav problem. Godinu dana kasnije Ralph Edward Gomory razvio je metodu cjelobrojnih formi. (Lj. Martić, 1979.) Prije objašnjavanja samog cjelobrojnog programiranja potrebno je shvatiti pojmove matematičkog i linearnog programiranja.

Problem matematičkog programiranja definira se kao

$$\min/\max f(x_1, x_2, \dots, x_n)$$

uz ograničenja

$$g_i(x_1, x_2, \dots, x_n) = b_i, \qquad i = 1,2,\dots,m$$
$$x_j \geq 0, \qquad j = 1,2,\dots,n.$$

Ako su funkcija cilja f i ograničenja $g_i,\ i = 1,2,\dots,m$ linearne funkcije, radi se o problemu linearnog programiranja. Linearno programiranje se prema M. Mikić i sur. (2011., str. 452) definira kao:

> *...grana matematike koja se bavi proučavanjem teorijskih, programskih i aplikativnih problema povezanih s optimizacijom unutar zadanih ograničenja izraženih sustavom linearnih jednadžbi i/ili nejednadžbi uz nenegativnost varijabli odlučivanja. Optimizacija se odnosi na odabiranje optimalne kombinacije više varijabli.*

Nadalje, neka je x vektor od n komponenti $x_1, x_2, \dots, x_n$. Za zadanu realnu linearnu funkciju od n strukturnih varijabli traži se ekstrem, koji može biti minimum ili maksimum, uz uvjet da bude zadovoljeno $m + k$ linearnih ograničenja postavljenih na strukturne varijable, a formuliranih u obliku linearnih jednadžbi i nejednadžbi. Varijable $x_j,\ j = 1, \dots, n$ nazivaju se strukturnim varijablama. Problem linearnog programiranja bez ograničenja cjelobrojnosti naziva se relaksirani problem. (D. Kalpić, V. Mornar, 1996.; Z. Lukač, L. Neralić, 2012.)

Ukoliko se na varijable x_j u problemu matematičkog programiranja postavi ograničenje cjelobrojnosti, radi se o problemu čisto cjelobrojnog programiranja. Također, ako su funkcija cilja f i ograničenja $g_i,\ i = 1,2,\dots,m$ linearne funkcije, radi se o problemu čisto cjelobrojnog linearnog programiranja, koji se prema Z. Lukač i L. Neraliću (2012.) definira kao

$$\min/\max z = \sum_{j=1}^{n} c_j x_j$$

uz ograničenja

$$\sum_{j=1}^{n} a_{ij} x_j = b_j;\ i = 1, \dots, m$$

$$x_j \geq 0,\ x \in \mathbb{Z};\ j = 1, \dots, n.$$

Djelomično (ili mješovito) cjelobrojno programiranje oblik je kod kojega se samo za neke varijable zahtijeva da poprime cjelobrojne vrijednosti. Problem djelomično cjelobrojnog linearnog programiranja ima oblik

$$\min/\max z = \sum_{j=1}^{n} c_j x_j + \sum_{k=1}^{p} d_k \bar{x}_k$$

uz ograničenja

$$\sum_{j=1}^{n} a_{ij} x_j + \sum_{k=1}^{p} h_{ik} \bar{x}_k = b_i, \qquad i = 1,2, \dots, m$$

$$x_j \geq 0, \qquad x_j \in \mathbb{Z}, \qquad j = 1,2, \dots, n$$

$$\bar{x}_k \geq 0, \qquad k = 1,2, \dots, p.$$

Postoji jedna posebna klasa problema cjelobrojnog programiranja pod nazivom binarno programiranje, u kojem varijable, uz već spomenuto svojstvo cjelobrojnosti, imaju ograničenje da mogu poprimiti vrijednost samo 0 ili 1.

3. Metode za cjelobrojno programiranje

Dvije su osnovne metode za rješavanje cjelobrojnog programiranja. To su metoda cjelobrojnih formi, odnosno Gomoryjeva metoda, i metoda grananja i ograđivanja. Kod metode cjelobrojnih formi stalno se nadograđuje ista simpleks matrica, generiraju se odsječci koji iz skupa mogućih rješenja izbace dio decimalnih rješenja sve dok se ne postigne ili optimalno cjelobrojno rješenje, ili nemoguće rješenje. Kod metode grananja i ograđivanja rješavanje se grana u više smjerova, ovisno o decimalnim varijablama, te se za svaki krak grananja rješava simpleks matrica neovisna o grananju na drugoj strani. Postoje još i heurističke metode, koje su, prema D. Barkoviću (2001.), vremenski kraće, ali nerijetko vode do rješenja blizu optimalnog, a ne do samog optimalnog rješenja, te zbog toga neće biti obrađene.

3.1. Metoda cjelobrojnih formi

U prethodnom poglavlju spomenuto je kako je R. E. Gomory 1958. godine razvio ovu metodu, prvu metodu koja rješava problem cjelobrojnog programiranja, te se ta metoda po njemu naziva još i Gomoryjeva metoda. Uz potporu sveučilišta Princeton, Gomory je na njihovom E101 računalu proveo prvi uspješan algoritam za računanje cjelobrojnih rješenja, u sklopu Princeton-IBM projekta za matematičko istraživanje. Algoritam je bio primjenjiv za sustave do sedam varijabli. (R. E. Gomory, 1958.)

Gomoryjeva metoda provodi se tako da se najprije zadani linearni problem riješi simpleks metodom, te ako su optimalne vrijednosti varijabli cijeli brojevi, postupak je gotov i postignuto je optimalno rješenje. Ako su varijable necjelobrojne, generira se nova linearna (ne)jednadžba, (Lj. Martić, 1979.), odnosno odsijecajuća ravnina, ili skraćeno odsječak, koja "odsijeca" dio skupa mogućih rješenja tako da odsječeni dio ne sadržava niti jednu cjelobrojnu točku, te se postupak provodi dok se ne dobije optimalno cjelobrojno rješenje. (Z. Lukač, L. Neralić, 2012.)

Algoritam metode cjelobrojnih formi (Z. Lukač, L. Neralić, 2012., str. 136):

1. korak: Riješiti problem cjelobrojnog linearnog programiranja bez ograničenja cjelobrojnosti, odnosno naći rješenje relaksiranog problema.

2. korak: Ako je rješenje relaksiranog problema cjelobrojno, zaustaviti se jer je postignuto optimalno rješenje problema cjelobrojnog programiranja. U protivnom, ići na 3. korak.

3. korak: Izabrati bazičnu varijablu koja nije cjelobrojna i generirati odsijecajuću ravninu. Jednadžba odsijecajuće ravnine dodaje se u optimalnu tablicu, odnosno simpleks matricu rješenja relaksiranog problema te se izvodi korak dualne simpleks metode. Nakon toga se ponovno vraća na 2. korak.

Otvoreno pitanje je kako generirati odsijecajuću ravninu. Za razrješenje problema potrebno je uvesti određene pretpostavke: neka je $[a]$ najveći cijeli broj koji je manji ili jednak od a, i neka je $f_a = a - [a]$, odnosno $a = [a] + f_a$, pri čemu vrijedi $0 \leq f_a < 1$. (Z. Lukač, L. Neralić, 2012.)

Primjeri:

a) $[2.6] = 2,\ f_{2.6} = 2.6 - 2 = 0.6$
b) $[-1.8] = -2,\ f_{-1.8} = -1.8 - (-2) = 0.2$
c) $[5] = 5,\ f_5 = 5 - 5 = 0$
d) $[-3] = -3,\ f_{-3} = -3 - (-3) = 0$
e) $[0] = 0,\ f_0 = 0 - 0 = 0$

Navedeni primjeri pokazuju da f_a za cijeli broj uvijek iznosi 0, tako da računanje f_a ima smisla samo za decimalne brojeve.

Za generiranje odsijecajuće ravnine u stupcu rješenja odabire se redak koji ima necjelobrojno rješenje. Ukoliko je necjelobrojnih rješenja više, odabire se onaj redak koji ima najveći koeficijent f_{i0}. Ukoliko postoje dva ili više redaka s istim koeficijentom f_{io}, svejedno je koji će se redak odabrati. (Z. Lukač, L. Neralić, 2012.)

Potom se za sve varijable x_{ij} u tom retku izračunavaju koeficijenti f_{ij} te vrijedi

$$f_{i0} \leq \sum_{j=1}^{n} f_{ij}\, x_j$$

odnosno ako se sve prebaci na lijevu stranu

$$f_{i0} - \sum_{j=1}^{n} f_{ij}\, x_j \leq 0.$$

Nejednadžba se pretvara u jednadžbu dodavanjem pomoćne varijable s_t gdje je t broj koraka dodavanja nove varijable. Sada jednadžba ima oblik

$$f_{i0} - \sum_{j=1}^{n} f_{ij}\, x_j + s_t = 0.$$

f_{i0} se prebacuje na desnu stranu:

$$-\sum_{j=1}^{n} f_{ij}\, x_j + s_t = -f_{i0}.$$

(D. Barković, 2001.)

Vidljivo je u algoritmu da se koraci 2 i 3 mogu ponavljati više puta. Ako se problem ne riješi nakon dodavanja prvog odsječka, postupak se izvodi iznova. Ne briše se prethodno dodani odsječak, nego se samo još nadodaje novi. J. Petrić, Z. Kojić i L. Šarenac (1978.) govore da se svaki korak dodavanja nove pomoćne varijable s_t bazira na osnovi prethodnog, tako da se već postojećim ograničenjima doda novo linearno ograničenje, odnosno odsječak.

Radi lakšeg razumijevanja metode, slijedi primjer u kojemu treba izračunati funkciju cilja

$$\max z = 12x_1 + 9x_2$$

uz ograničenja

$$4x_1 + 3x_2 \leq 16$$
$$2x_1 + 7x_2 \leq 19$$
$$x_1 - x_2 \geq 3$$
$$x_1, x_2 \geq 0,\ x_1, x_2 \in \mathbb{Z}.$$

U prvom koraku potrebno je pretvoriti nejednadžbe u jednadžbe dodavanjem pomoćnih varijabli i problem zapisati u kanonskom obliku:

$$\max z = 12x_1 + 9x_2 + 0y_1 + 0y_2 + 0y_3$$
$$4x_1 + 3x_2 + y_1 = 16$$
$$2x_1 + 7x_2 + y_2 = 19$$
$$-x_1 + x_2 + y_3 = -3$$
$$x_1, x_2, y_1, y_2, y_3 \geq 0, \qquad x_1, x_2, y_1, y_2, y_3 \in \mathbb{Z}.$$

U drugom koraku izrađuje se simpleks matrica prema kanonskom problemu.

Tablica 3.1. Rješenje relaksiranog problema

X_b	X_1	X_2	Y_1	Y_2	Y_3	b_i
Y_1	4	3	1	0	0	16
Y_2	2	7	0	1	0	19
Y_3	-1	1	0	0	1	-3
$\overline{C_j}$	-12	-9	0	0	0	0
Y_1	0	7	1	0	4	4
Y_2	0	9	0	1	2	13
X_1	1	-1	0	0	-1	3
$\overline{C_j}$	0	-21	0	0	-12	36
X_2	0	1	1/7	0	4/7	4/7
Y_2	0	0	-9/7	1	-22/7	55/7
X_1	1	0	1/7	0	-3/7	25/7
$\overline{C_j}$	0	0	3	0	0	48

Izvor: izrada autora rada

U tablici su pivot elementi označeni narančasto radi lakšeg shvaćanja simpleks postupka. Prvi korak gore spomenutog algoritma, odnosno rješenje relaksiranog problema iznosi 48. Drugi korak algoritma predstavlja provjeru rješenja: funkcija cilja jest cjelobrojna, ali varijable nisu, stoga je potrebno primijeniti treći korak i generirati odsijecajuću ravninu. Budući da vrijedi $f_{10} = f_{30} = \frac{4}{7}$ i $f_{20} = \frac{6}{7}$, za generiranje ravnine odabire se drugi redak.

Koeficijente f_{ij} moguće je računati i na jednostavniji način, prema sljedećem pravilu:

a) **za pozitivne brojeve**:

Neka je $\frac{x}{y}$ pozitivan broj. Tada vrijedi $f\left(\frac{x}{y}\right) = \frac{x \bmod y}{y}$, npr. za razlomak $\frac{27}{5}$ vrijedi $f\left(\frac{27}{5}\right) = \frac{27 \bmod 5}{5} = \frac{2}{5}$. Ako vrijedi $0 \leq \frac{x}{y} < 1$, tada $f\left(\frac{x}{y}\right)$ nije potrebno računati, jer je $f\left(\frac{x}{y}\right) = \frac{x}{y}$.

b) **za negativne brojeve**:

Neka je $-\frac{x}{y}$ negativan broj. Tada vrijedi $f\left(-\frac{x}{y}\right) = \frac{y-(x \bmod y)}{y}$, npr. za primjer $-\frac{20}{7}$ vrijedi $f\left(-\frac{20}{7}\right) = \frac{7-(20 \bmod 7)}{7} = \frac{7-6}{7} = \frac{1}{7}$.

Tako za drugi redak vrijedi $f_{23} = \frac{5}{7}$ i $f_{25} = \frac{6}{7}$ te nova linearna jednadžba glasi:

$$-\frac{5}{7}y_1 - \frac{6}{7}y_3 + s_1 = -\frac{6}{7}$$

Novo formirana jednadžba uvrštava se u simpleks tablicu:

Tablica 3.2. Simpleks postupak nakon dodane prve odsijecajuće ravnine

X_b	X_1	X_2	Y_1	Y_2	Y_3	S_1	b_i
X_2	0	1	1/7	0	4/7	0	4/7
Y_2	0	0	-9/7	1	-22/7	0	55/7
X_1	1	0	1/7	0	-3/7	0	25/7
S_1	0	0	-5/7	0	-6/7	1	-6/7
$\overline{C_j}$	0	0	3	0	0	0	48
X_2	0	1	-1/3	0	0	2/3	0
Y_2	0	0	4/3	1	0	-11/3	11
X_1	1	0	1/2	0	0	-1/2	4
Y_3	0	0	5/6	0	1	-7/6	1
$\overline{C_j}$	0	0	3	0	0	0	48

Izvor: izrada autora rada

Ponovno se provodi provjera rješenja. Sve varijable su cjelobrojne te je rješenje ovoga problema optimalno. Rješenje iznosi 48, bazične varijable iznose $x_1 = 4, x_2 = 0, y_2 = 11$ i $y_3 = 1$, dok su nebazične varijable uvijek 0, odnosno $y_1 = s_1 = 0$. Ispravnost postupka može se provjeriti uvrštavanjem varijabli u početne jednadžbe kanonskog problema:

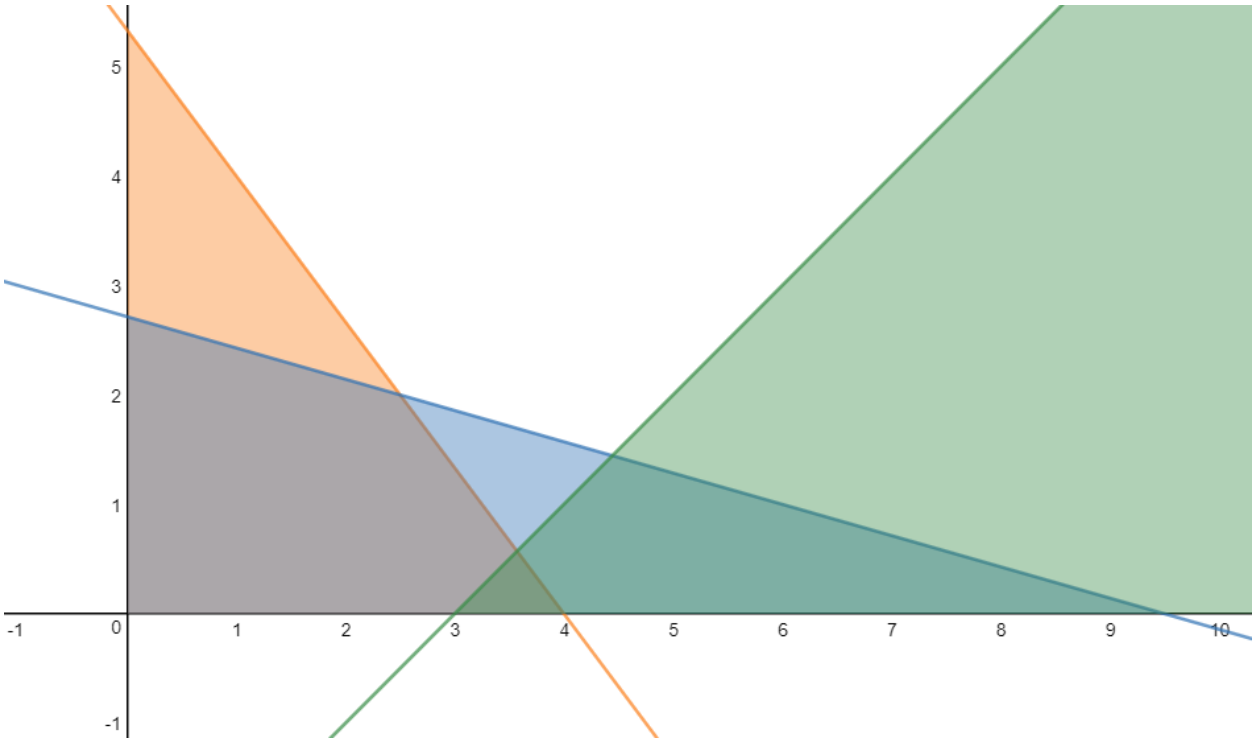
$$\max z = 12x_1 + 9x_2 + 0x_3 + 0x_4 = 12 \cdot 4 + 9 \cdot 0 = 48$$

$$4x_1 + 3x_2 + y_1 = 4 \cdot 4 + 3 \cdot 0 + 0 = 16$$

$$2x_1 + 7x_2 + y_2 = 2 \cdot 4 + 7 \cdot 0 + 11 = 19$$

$$-x_1 + x_2 + y_3 = -4 + 0 + 1 = -3$$

Kako funkcija cilja i svi uvjeti odgovaraju rezultatima simpleks postupka, zaključuje se da je postupak uspješno proveden. Ti rezultati mogu se prikazati i grafički:

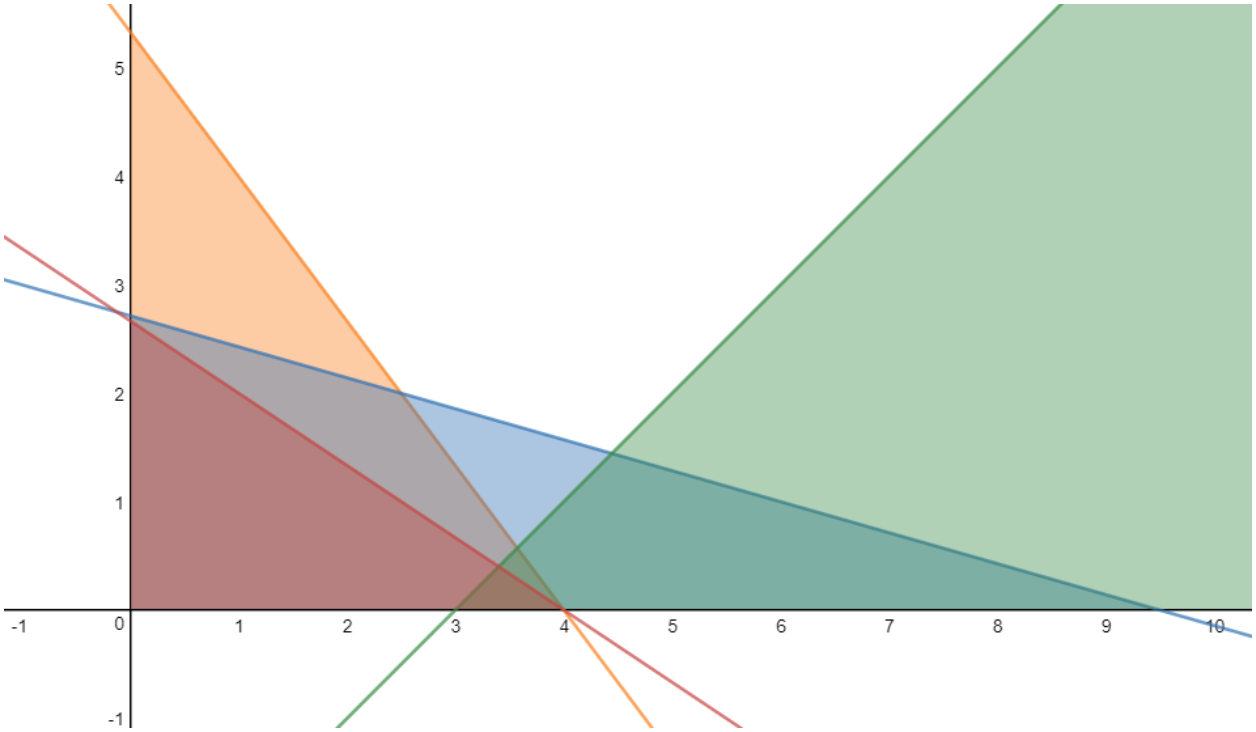


Slika 3.1. Početna ograničenja problema

Izvor: izrada autora rada, graf izrađen na stranici Desmos, www.desmos.com

Graf prikazuje početna linearna ograničenja, bez odsijecajuće ravnine. Ograničenje $4x_1 + 3x_2 \leq 16$ prikazano je narančastom bojom, plava boja prikazuje nejednadžbu $2x_1 + 7x_2 \leq 19$, dok je $x_1 - x_2 \geq 3$ prikazano zelenom bojom. Jedina moguća cjelobrojna rješenja nalaze se na presjeku domena nejednadžbi, a to su $(3,0)$ i $(4,0)$.

Jednadžba odsijecajuće ravnine glasi $-\frac{5}{7}y_1 - \frac{6}{7}y_3 + s_1 = -\frac{6}{7}$, odnosno $s_1 = \frac{5}{7}y_1 + \frac{6}{7}y_3 - \frac{6}{7}$. Iz prve i treće jednadžbe ograničenja kanonskog oblika problema varijable y_1 i y_3 mogu se izraziti pomoću x_1 i x_2, te se dobije da vrijedi $y_1 = -4x_1 - 3x_2 + 16$ i $y_3 = x_1 - x_2 - 3$. Ti izrazi uvrste se u jednadžbu za s_1 i dobije se jednadžba $s_1 = -2x_1 - 3x_2 + 8$. Kako mora biti $s_1 \geq 0$, na graf sa slike 3.1. dodaje se nejednadžba $-2x_1 - 3x_2 + 8 \geq 0$.



Slika 3.2. Dodan odsječak

Izvor: izrada autora rada, graf izrađen na stranici Desmos, www.desmos.com

Odsijecajuća ravnina s_1 prikazana je crvenom bojom. Vidljivo je da su oba moguća cjelobrojna rješenja i dalje unutar skupa rješenja, odnosno nije odsječeno niti jedno cjelobrojno rješenje, upravo kako govore Lukač i Neralić.

Prethodno je prikazana primjena metode na jednostavnom primjeru, u kojem se već nakon dodavanja prvog odsječka dobije rješenje. U nastavku je prikazan primjer u kojem će biti potrebno dodati više od jedne odsijecajuće ravnine:

$$\max z = 25x_1 + 18x_2 + 0y_1 + 0y_2 + 0y_3$$
$$9x_1 + 13x_2 + y_1 = 32$$
$$17x_1 - 7x_2 + y_2 = 45$$
$$-4x_1 + 3x_2 + y_3 = -10$$
$$x_1, x_2, y_1, y_2, y_3 \geq 0,\ x_1, x_2, y_1, y_2, y_3 \in \mathbb{Z}$$

Problem je već zapisan u kanonskom obliku te se izrađuje simpleks matrica. Budući da su koraci rješavanja već objašnjeni, bit će prikazane samo početna i završna tablica u svakom koraku.

Tablica 3.3. Relaksirani problem

X_b	X_1	X_2	Y_1	Y_2	Y_3	b_i
Y_1	9	13	1	0	0	32
Y_2	17	-7	0	1	0	45
Y_3	-4	3	0	0	1	-10
$\overline{C_j}$	-25	-18	0	0	0	0
Y_1	0	0	1	-79/23	-284/23	21/23
X_2	0	1	0	4/23	17/23	10/23
X_1	1	0	0	3/23	7/23	65/23
$\overline{C_j}$	0	0	0	147/23	481/23	1805/23

Izvor: izrada autora rada

Tablica 3.4. Dodan prvi odsječak

X_b	X_1	X_2	Y_1	Y_2	Y_3	S_1	b_i
Y_1	0	0	1	-79/23	-284/23	0	21/23
X_2	0	1	0	4/23	17/23	0	10/23
X_1	1	0	0	3/23	7/23	0	65/23
S_1	0	0	0	-13/23	-15/23	1	-21/23
$\overline{C_j}$	0	0	0	147/23	481/23	0	1805/23
Y_1	0	0	1	0	-109/13	-79/13	84/13
X_2	0	1	0	0	7/13	4/13	2/13
X_1	1	0	0	0	2/13	3/13	34/13
Y_2	0	0	0	1	15/13	-23/13	21/13
$\overline{C_j}$	0	0	0	0	176/13	147/13	886/13

Izvor: izrada autora rada

U tablici 3.3. jednadžba odsijecajuće ravnine generirana je pomoću prvog retka te je dodan odsječak $-\frac{13}{23}y_1 - \frac{15}{23}y_2 + s_1 = -\frac{21}{23}$. Kako prikazuje tablica 3.4., ovdje se cjelobrojno rješenje nije postiglo s prvim odsječkom te je potrebno dodati još jedan. Iz tablice 3.4. odabire se drugi ili četvrti redak, dobija se ista jednadžba koja glasi $-\frac{2}{13}y_3 - \frac{3}{13}s_1 + s_2 = -\frac{8}{13}$.

Tablica 3.5. Dodan drugi odsječak

X_b	X_1	X_2	Y_1	Y_2	Y_3	S_1	S_2	b_i
Y_1	0	0	1	0	-109/13	-79/13	0	84/13
X_2	0	1	0	0	7/13	4/13	0	2/13
X_1	1	0	0	0	2/13	3/13	0	34/13
Y_2	0	0	0	1	15/13	-23/13	0	21/13
S_2	0	0	0	0	-2/13	-3/13	1	-8/13
$\overline{C_j}$	0	0	0	0	176/13	147/13	0	886/13
Y_1	0	0	1	0	-13/3	0	-79/3	68/3
X_2	0	1	0	0	1/3	0	4/3	-2/3
X_1	1	0	0	0	0	0	1	2
Y_2	0	0	0	1	7/3	0	-23/3	19/3
S_1	0	0	0	0	2/3	1	-13/3	8/3
$\overline{C_j}$	0	0	0	0	6	0	49	38

Izvor: izrada autora rada

Bitno je napomenuti da se novi odsječak dodaje uz već prethodni, na posljednje rješenje, odnosno ne briše se prethodni odsječak. Kao što je prikazano u gornjem dijelu tablice 3.5, novi odsječak može se vidjeti desno od stupca s_1. Donji dio tablice 3.5. pokazuje da zadatak nema rješenja: u drugom retku u stupcu rješenja nalazi se negativna vrijednost koja se ne može poništiti budući da su svi koeficijenti varijabli u tom retku pozitivni. Stoga ovaj problem nema mogućeg rješenja.

3.2. Metoda grananja i ograđivanja

Metoda grananja i ograđivanja jedna je od tri metode stabla odlučivanja. Metoda potpunog prebrojavanja i metoda dinamičkog programiranja temelje se na prebrojavanju i razvijene su konkretno za problem trgovačkog putnika, gdje se traženje rješenja većinom svodi na zbrajanje te nije moguće dobiti decimalni iznos. Metoda grananja i ograđivanja također se koristi za rješavanje problema trgovačkog putnika, ali i za rješavanje problema cjelobrojnog programiranja, doduše u izmijenjenom obliku, iako je princip isti. (D. Barković, 2001.)

Počeci ove metode sežu u 1960-ih godina kada su studenti iz Calcutte Katta G. Murty i Caroline Carel započeli istraživanje problema putujućeg trgovca te u suradnji s profesorom na Tehnološkom institutu u Massachusettsu (MIT) Johnom Littleom 1963. objavili algoritam za rješavanje tog problema. Naziv metodi "grananje i ograđivanje" (eng. "*Branch And Bound*") dala je Dura W. Sweeney, Littleova studentica na MIT-u, koja je i sama spomenuta kao koautor. (K. G. Murty, 2012.)

Također 60-ih godina prošlog stoljeća A. H. Land i A. G. Doig razvijali su ovu metodu za rješavanje problema cjelobrojnog i djelomično cjelobrojnog programiranja. R. J. Dakin je metodu poboljšao uvođenjem granica, tako da se optimalno rješenje traži na ograničenom rasponu vrijednosti, što znatno skraćuje cijeli proces. Dakin je svoj članak "*A tree-search algorithm for mixed integer programming problems*" objavio u časopisu *The Computer Journal* u izdanju za siječanj 1965. (R. J. Dakin, 1965.)

Ideja metode je da se, kao i kod Gomoryjeve metode, najprije riješi relaksirani problem, a potom se s varijablama koje su decimalne, a trebale bi biti cjelobrojne, vrši grananje i generiraju dva nova linearna programa koja se smještaju u čvorove binarnog stabla. Necjelobrojnim se varijablama zabranjuje da poprime svoju optimalnu necjelobrojnu vrijednost te se generiraju dva nova linearna programa koji se smještaju u čvorove binarnog stabla, pri čemu lijevi čvor predstavlja linearni program u kojem promatrana varijabla može poprimiti vrijednost manju ili jednaku prvom cijelom broju manjem od dobivenog rješenja, a desni čvor linearni program za vrijednost veću ili jednaku prvom većem cijelom broju. Vrijednost početnog čvora uzima se za gornju ogradu. (D. Kalpić, V. Mornar, 1996.)

Algoritam metode grananja i ograđivanja (Z. Lukač, L. Neralić, 2012., str. 142-143):

1. **Početak algoritma**

Odrediti donju ogradi z_D na maksimalnu vrijednost funkcije cilja problema cjelobrojnog programiranja, tako da se npr. odredi neko moguće rješenje $\bar{x}$ i izračuna vrijednost funkcije cilja $z(\bar{x})$ te se stavi $z_D = z(\bar{x})$. Ako se ne raspolaže s mogućim rješenjem, može se uzeti da je $z_D = -\infty$.

2. **Grananje**

Izvršiti separaciju jednog od preostalih podskupova skupa mogućih rješenja na dva ili više novih podskupova. Na samomo početku treba izvršiti separaciju cijelog skupa mogućih rješenja. Npr., separacija se može provesti tako da jedan dio bude onaj podskup za koji vrijedi $x_k \leq c$, a drugi onaj za koji vrijedi $x_k \geq c + 1$, za neku varijablu k i neku cjelobrojnu konstantu c.

3. **Izračunavanje ograda**

Za svaki novi podskup izračunati gornju ogradu z_G na maksimalnu vrijednost funkcije cilja na tom podskupu.

4. **Isključivanje**

Neki podskup može se isključiti iz daljnjeg razmatranja ako:

I. *vrijedi* $z_G < z_D$
II. *podskup nema mogućeg rješenja*
III. *nađeno je najbolje moguće cjelobrojno rješenje na tom podskupu, s vrijednošću funkcije cilja $z_{D'}$. U ovom slučaju, ako je $z_{D'} > z_D$, staviti $z_D = z_{D'}$, sačuvati to rješenje i primijeniti test I. na sve preostale podskupove.*

5. **Završetak algoritma**

Ako su svi podskupovi isključeni, zaustaviti se. Tada je sačuvano tekuće rješenje optimalno. U protivnom, ići na drugi korak.

U slučaju više decimalnih varijabli, potrebno je odrediti prioritete, odnosno prema kojoj će se varijabli prvo obaviti grananje. D. Kalpić i V. Mornar (1996.) navode tri kriterija. Prvi način je poredati varijable po učešću u funkciji cilja, odnosno prvo se grananje obavlja prema varijablama s najvećim učešćem u funkciji cilja. Kod binarnog programiranja ova metoda nije uvijek primjenjiva jer binarne varijable ne moraju biti izravno prisutne u funkciji cilja. Drugi način je poredati varijable po udaljenosti od cjelobrojnosti, odnosno od najbližeg cijelog broja. Prvo se donosi odluka o varijablama koje su najdalje od cjelobrojnosti. Treći način je kombinacija dva već spomenuta načina, odnosno umnožak koeficijenta u funkciji cilja i udaljenosti od cjelobrojnosti. Najefikasnije je ipak odrediti hijerarhijsko značenje varijabli.

U nastavku se daje prikaz metode na primjeru. Potrebno je pronaći funkciju cilja

$$\max z = 3x_1 + 5x_2$$

uz ograničenja

$$2x_1 + 3x_2 \leq 15$$
$$x_1 - 2x_2 \geq 2 \;\longrightarrow\; -x_1 + 2x_2 \leq -2$$
$$x_1, x_2 \geq 0, \qquad x_1, x_2 \in \mathbb{Z}.$$

U prvom koraku problem se zapisuje u kanonskom obliku:

$$\max z = 3x_1 + 5x_2 + 0y_1 + 0y_2$$
$$2x_1 + 3x_2 + y_1 = 15$$
$$-x_1 + 2x_2 + y_2 = -2$$
$$x_1, x_2, y_1, y_2 \geq 0, \qquad x_1, x_2, y_1, y_2 \in \mathbb{Z},$$

te se prema ograničenjima izrađuje simpleks tablica.

Tablica 3.6. Početna simpleks tablica

X_B	X_1	X_2	Y_1	Y_2	b_i
Y_1	2	3	1	0	15
Y_2	-1	2	0	1	-2
$\overline{C_j}$	-3	-5	0	0	0

Izvor: izrada autora rada

U daljnjem koraku potrebno je odrediti donju ogradu, odnosno neko proizvoljno rješenje koje zadovoljava sve navedene uvjete. Uzmu li se npr. vrijednosti $x_1 = 0, x_2 = 0$, dobije se da je $z_D = z(0,0) = 0$, što ne dozvoljava drugo ograničenje, jer $0 - 2 \cdot 0$ nije veće od 2. Stoga je potrebno odabrati neke druge vrijednosti, npr. $x_1 = 3, x_2 = 0$. Rješenje $z(3,0) = 9$ zadovoljava oba uvjeta te se može uzeti $z_D = 9$. Potom se najprije rješava relaksirani problem. Pivot elementi će biti označeni na isti način kao i kod prethodne metode.

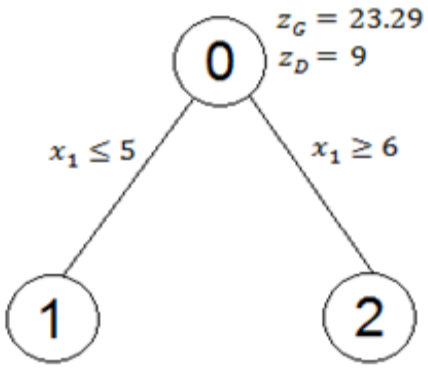
Tablica 3.7. Rješenje relaksiranog problema

X_B	X_1	X_2	Y_1	Y_2	b_i
Y_1	2	3	1	0	15
Y_2	-1	2	0	1	-2
$\overline{C_j}$	-3	-5	0	0	0
Y_1	0	7	1	2	11
X_1	1	-2	0	-1	2
$\overline{C_j}$	0	-11	0	-3	6
X_2	0	1	1/7	2/7	11/7
X_1	1	0	2/7	-3/7	36/7
$\overline{C_j}$	0	0	11/7	1/7	163/7

Izvor: izrada autora rada

Budući da rješenja nisu cjelobrojna, prelazi se na postupak grananja. Najprije se određuje gornja granica mogućih rješenja, ovdje je to $\frac{163}{7} \approx 23.29$. Pretpostavka je da varijabla x_1 ima više hijerarhijsko značenje, pa će se prema njoj napraviti prvo grananje. Varijabla x_1 ovdje iznosi $\frac{36}{7} \approx 5.14$, $5 < 5.14 < 6$, dakle skup se dijeli na dva podskupa, jedan za koji vrijedi $x_1 \leq 5$ i drugi za koji vrijedi $x_1 \geq 6$, čime se varijabli x_1 onemogućava da poprimi svoju optimalnu, ali decimalnu vrijednost.

Zatim se prema obavljenoj separaciji skicira stablo odlučivanja, koje prikazuje tijek separacije. Početni skup grana se na dva podskupa, te se za svaki podskup dodaje nova (ne)jednadžba i provodi postupak kao i za glavni skup S_0. Broj unutar kruga predstavlja indeks podskupa.



Slika 3.2. Početno grananje

Izvor: izrada autora rada

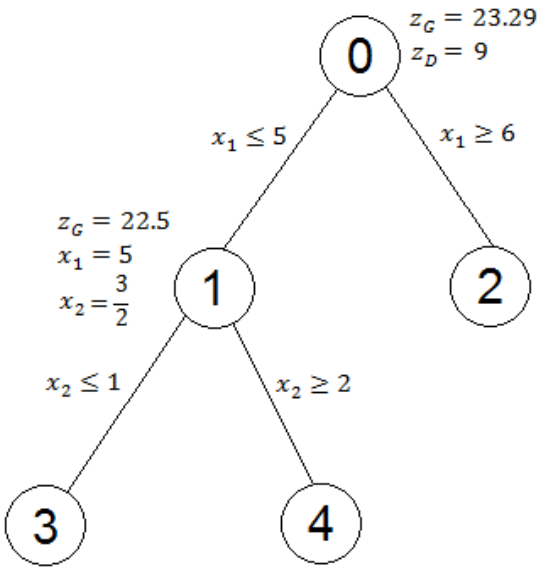
Potom se radi simpleks matrica za skup S_1. Od nejednadžbe $x_1 \leq 5$ dodavanjem nove varijable y_3 radi se jednadžba $x_1 + y_3 = 5$.

Tablica 3.8. Podskup S_1

X_B	X_1	X_2	Y_1	Y_2	Y_3	b_i
X_2	0	1	1/7	2/7	0	11/7
X_1	1	0	2/7	-3/7	0	36/7
Y_3	1	0	0	0	1	5
$\overline{C_j}$	0	0	11/7	1/7	0	163/7
X_2	0	1	1/7	2/7	0	11/7
X_1	1	0	2/7	-3/7	0	36/7
Y_3	0	0	-2/7	3/7	1	-1/7
$\overline{C_j}$	0	0	11/7	1/7	0	163/7
X_2	0	1	0	1/2	1/2	3/2
X_1	1	0	0	0	1	5
Y_1	0	0	1	-3/2	-7/2	1/2
$\overline{C_j}$	0	0	0	5/2	11/2	45/2

Izvor: izrada autora rada

Kako rješenje iznosi $\frac{45}{2} = 22.5$, odnosno unutar z_D i z_G, rješenje je zadovoljavajuće i skup se ne eliminira. Varijabla x_1 je poprimila cjelobrojnu vrijednost, ali x_2 nije i iznosi 1.5, te je sada prema njoj potrebno napraviti daljnje grananje, odnosno $x_2 \leq 1$ i $x_2 \geq 2$. Za gornju granicu na podskupu S_1 uzima se 22.5 te se proširuje stablo grananja:



Slika 3.3. Grananje iz skupa S_1

Izvor: izrada autora rada

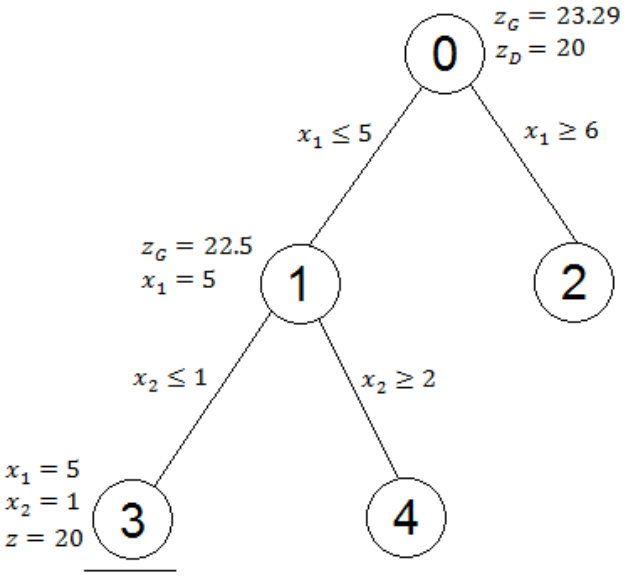
Nastavlja se grananje na skupu S_1, najprije se za skup S_3 dodaje jednadžba $x_2 + y_4 = 1$.

Tablica 3.9. Podskup S_3

X_B	X_1	X_2	Y_1	Y_2	Y_3	Y_4	b_i
X_2	0	1	0	1/2	1/2	0	3/2
X_1	1	0	0	0	1	0	5
Y_1	0	0	1	-3/2	-7/2	0	1/2
Y_4	0	1	0	0	0	1	1
$\overline{C_j}$	0	0	0	5/2	11/2	0	45/2
X_2	0	1	0	1/2	1/2	0	3/2
X_1	1	0	0	0	1	0	5
Y_1	0	0	1	-3/2	-7/2	0	1/2
Y_4	0	0	0	-1/2	-1/2	1	-1/2
$\overline{C_j}$	0	0	0	5/2	11/2	0	45/2
X_2	0	1	0	0	0	1	1
X_1	1	0	0	0	1	0	5
Y_1	0	0	1	0	-2	-3	2
Y_2	0	0	0	1	1	-2	1
$\overline{C_j}$	0	0	0	0	3	5	20

Izvor: izrada autora rada

Ovdje je uvjet cjelobrojnosti zadovoljen i kao $z_{D'}$ se uzima vrijednost 20. Budući da vrijedi $z_{D'} > z_D$, ova vrijednost se prihvaća kao nova donja granica, odnosno sada vrijedi $z_D = 20$ te se S_3 (slučaj III) na stablu odlučivanja eliminira.



Slika 3.4. Eliminiran podskup S_3

Izvor: izrada autora rada

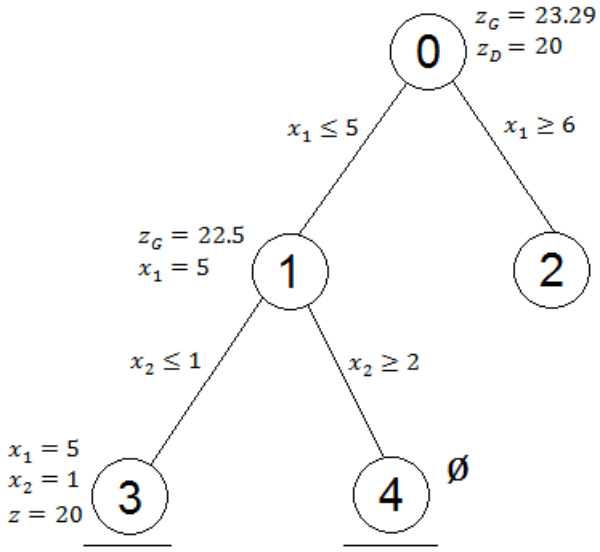
Sada se rješava i podskup S_4. Nejednadžba $x_2 \geq 2$ se pretvara u jednadžbu $x_2 - y_4 = 2$, što nakon množenja s -1 izgleda $-x_2 + y_4 = -2$.

Tablica 3.10. Podskup S_4

X_B	X_1	X_2	Y_1	Y_2	Y_3	Y_4	b_i
X_2	0	1	0	1/2	1/2	0	3/2
X_1	1	0	0	0	1	0	5
Y_1	0	0	1	-3/2	-7/2	0	1/2
Y_4	0	-1	0	0	0	1	-2
$\overline{C_j}$	0	0	0	5/2	11/2	0	45/2
X_2	0	1	0	1/2	1/2	0	3/2
X_1	1	0	0	0	1	0	5
Y_1	0	0	1	-3/2	-7/2	0	1/2
Y_4	0	0	0	1/2	1/2	1	-1/2
$\overline{C_j}$	0	0	0	5/2	11/2	0	45/2

Izvor: izrada autora rada

Ovaj podskup nema rješenja, što se vidi iz četvrtog retka. Također se eliminira i S_4 (slučaj II).



Slika 3.5. Eliminiran podskup S_4

Izvor: izrada autora rada

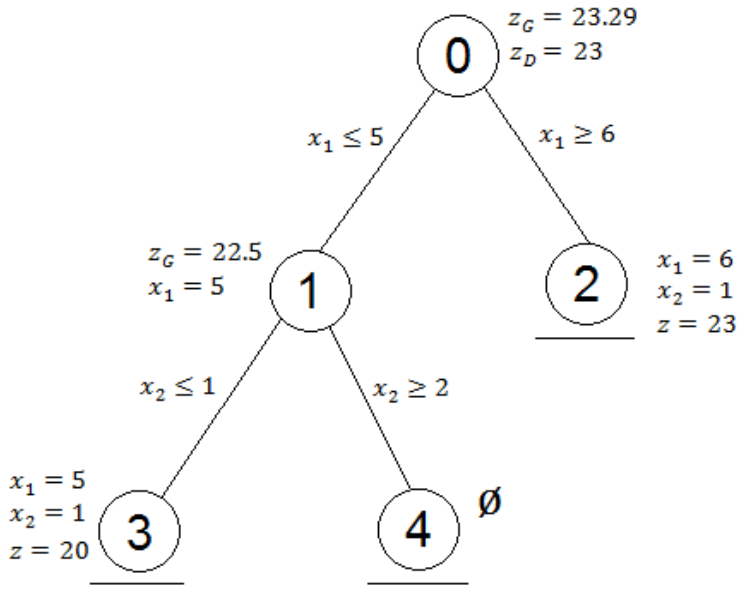
Na lijevoj strani početnog grananja svi su skupovi eliminirani, stoga se prelazi na desnu stranu. Potrebno je vratiti se na tablicu 3.7. i dodati novo ograničenje $x_1 \geq 6$, odnosno jednadžbu $-x_1 + y_3 = -6$.

Tablica 3.11. Podskup S_2

X_B	X_1	X_2	Y_1	Y_2	Y_3	b_i
X_2	0	1	1/7	2/7	0	11/7
X_1	1	0	2/7	-3/7	0	36/7
Y_3	-1	0	0	0	1	-6
$\overline{C_j}$	0	0	11/7	1/7	0	163/7
X_2	0	1	1/7	2/7	0	11/7
X_1	1	0	2/7	-3/7	0	36/7
Y_3	0	0	2/7	-3/7	1	-6/7
$\overline{C_j}$	0	0	11/7	1/7	0	163/7
X_2	0	1	1/3	0	2/3	1
X_1	1	0	0	0	-1	6
Y_1	0	0	-2/3	1	-7/3	2
$\overline{C_j}$	0	0	5/3	0	1/3	23

Izvor: izrada autora rada

Za novu donju granicu uzima se 23, budući da je veća od trenutne z_D. Potom se eliminira i S_2:



Slika 3.6. Završen postupak

Izvor: izrada autora rada

Budući da su svi skupovi eliminirani, postupak je gotov. Optimalno rješenje jednako je iznosu trenutne donje granice, što iznosi 23. Varijable iznose $x_1 = 6$ i $x_2 = 1$, dok pomoćna varijabla y_1 iznosi 2. Nebazične varijable y_2 i y_3 imaju vrijednost 0.

4. Problem fiksnih troškova

Fiksni troškovi nastaju nakon što varijabla poprimi vrijednosti različite od nule. Kada varijabla poprimi vrijednost 0, nestaju i fiksni troškovi. (D. Barković, 2001.)

Neka je j aktivnost koja ima svoje fiksne troškove po jedinici k_j i varijabilne troškove po jedinici c_j, a x_j je razina na kojoj operira aktivnost j. δ_j je varijabla čija vrijednost iznosi 1 ako se proizvod proizvodi, odnosno 0 ako se proizvod ne proizvodi. Funkcija troškova može se izraziti u obliku

$$h_j(x_j) = c_j x_j + k_j \delta_j$$

uz ograničenje

$$x_j(1 - \delta_j) = 0, \qquad \delta_j = 0 \text{ ili } 1.$$

Ako je $x_j > 0$, tada mora biti $\delta_j = 1$, te funkcija ima oblik

$$h_j(x_j) = c_j x_j + k_j.$$

Ako je $x_j = 0$, tada vrijedi $\delta_j = 0$ ili 1. Budući da su troškovi manji za $\delta_j = 0$, a funkcija $h_j(x_j)$ se želi minimizirati, tada vrijedi

$$h_j(x_j) = 0$$

Stoga se ukupan trošak za aktivnost j može izraziti u obliku:

$$h_j(x_j) = \begin{cases} c_j x_j + k_j & \text{za } x_j > 0, \\ 0 & \text{za } x_j = 0. \end{cases}$$

Funkcija cilja predstavlja sumu funkcijâ troškova, može se izraziti kao

$$z = \sum_{j=1}^{n} (c_j x_j + k_j \delta_j) = \sum_{j=1}^{n} h_j x_j$$

uz ograničenja

$$\sum_{j=1}^{n} a_{ij} x_j = a_{i0}, \qquad i = 1,2,\dots,m$$

$$x_j \geq 0, \qquad j = 1,2,\dots,n$$

$$\delta_j = \begin{cases} 0 \text{ ako je } x_j = 0, \\ 1 \text{ ako je } x_j > 0. \end{cases}$$

(Z. Lukač, L. Neralić, 2012.; D. Barković, 2001.)

U nastavku je prikazan primjer problema fiksnih troškova na tvornici čokoladnih proizvoda, riješen metodom grananja i ograđivanja. U tvornici se proizvode bombonijere i čokolade na četiri proizvodna stroja, s kapacitetima pojedinih strojeva od 200, 400, 150 i 250 jedinica proizvoda. Dohodak po jedinici proizvoda za bombonijeru iznosi 700, a za čokoladu 500 kn. Ukupni mjesečni fiksni troškovi proizvodnog pogona iznose 20000 kn, dok su troškovi proizvodnje bombonijere 6000 kn, a proizvodnje čokolade 4500 kn. Potrebno je pronaći program proizvodnje koji ostvaruje maksimalnu dobit. Elementi za konstrukciju modela prikazani su u sljedećoj tablici.

Tablica 4.1. Program proizvodnje: P1=bombonijera, P2=čokolada

	K	P1	P2
S1	200	2	1
S2	400	1	3
S3	150	3	0
S4	250	2	2
Dohodak po jedinici	/	700	500
Mjesečni fiksni troškovi	20000	6000	4500

Izvor: izrada autora rada

Budući da postoje mjesečni fiksni troškovi u iznosu od 20000 kn, vrijednost tih troškova oduzima se od funkcije cilja:

$$\max z = -20000 + (700x_1 - 6000\delta_1) + (500x_2 - 4500\delta_2).$$

Također moraju vrijediti uvjeti

$$2x_1 + x_2 + y_1 = 200,$$
$$x_1 + 3x_2 + y_2 = 400,$$
$$3x_1 + y_3 = 150,$$
$$2x_1 + 2x_2 + y_4 = 250.$$

g_j je varijabla koja je najmanje toliko velika koliko je najveća vrijednost od x_j, no budući da se te najveće vrijednosti često ne znaju, za g_j se odabiru konstante koje su se u ispitivanju pokazale dovoljno velike. Barković (2001.) je odabrao konstante 200 i 100 te će one biti korištene i u ovom primjeru.

Budući da vrijedi $x_j - g_j \delta_j \leq 0$, dobivaju se još dva nova ograničenja:

$$x_1 - 200\delta_1 + y_A = 0$$
$$x_2 - 100\delta_2 + y_B = 0$$

Također se na sve varijable x i y primjenjuju svojstva cjelobrojnosti i nenegativnosti:

$$x_1, x_2, y_1, y_2, y_3, y_4, y_A, y_B \geq 0, \in \mathbb{Z}$$
$$\delta_1, \delta_2 \in \{0,1\}$$

Nakon što su svi uvjeti postavljeni, izrađuje se simpleks matrica, te se rješava relaksirani problem, iz kojeg će se postaviti gornja granica.

Tablica 4.2. Početna tablica

	X_1	X_2	δ_1	δ_2	Y_1	Y_2	Y_3	Y_4	Y_A	Y_B	b_i
Y_1	2	1	0	0	1	0	0	0	0	0	200
Y_2	1	3	0	0	0	1	0	0	0	0	400
Y_3	3	0	0	0	0	0	1	0	0	0	150
Y_4	2	2	0	0	0	0	0	1	0	0	250
Y_A	1	0	-200	0	0	0	0	0	1	0	0
Y_B	0	1	0	-100	0	0	0	0	0	1	0
$\overline{C_j}$	-700	-500	6000	4500	0	0	0	0	0	0	-20000

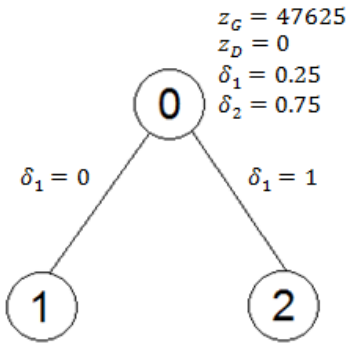
Izvor: izrada autora rada

Tablica 4.3. Rješenje relaksiranog problema

	X_1	X_2	δ_1	δ_2	Y_1	Y_2	Y_3	Y_4	Y_A	Y_B	b_i
Y_1	0	0	0	0	1	0	-1/3	-1/2	0	0	25
Y_2	0	0	0	0	0	1	2/3	-3/2	0	0	125
δ_1	0	0	1	0	0	0	1/600	0	-1/200	0	1/4
δ_2	0	0	0	1	0	0	-1/300	1/200	0	-1/100	3/4
X_1	1	0	0	0	0	0	1/3	0	-1	0	50
X_2	0	1	0	0	0	0	-1/3	1/2	0	0	75
$\overline{C_j}$	0	0	0	0	0	0	215/3	455/2	30	45	47625

Izvor: izrada autora rada

Za donju granicu može se staviti 0, dok je gornja granica 47625. Za δ_1 i δ_2 dobiveni su necjelobrojni iznosi, stoga po njima treba napraviti grananje. Neka se grananje započne s δ_1, čija vrijednost iznosi $\frac{1}{4} = 0.25$. Klasično grananje bi išlo na $\leq$ i $\geq$ za prvi manji i prvi veći cijeli broj, no budući da varijable δ mogu poprimiti vrijednosti samo 0 ili 1, grananje se označava kao $\delta_1 = 0$ na lijevoj strani i $\delta_1 = 1$ na desnoj.



Slika 4.1. Početno grananje

Izvor: izrada autora rada

Tablica 4.4. Podskup S_1

	X_1	X_2	δ_1	δ_2	Y_1	Y_2	Y_3	Y_4	Y_A	Y_B	δ'	b_i
Y_1	0	0	0	0	1	0	-1/3	-1/2	0	0	0	25
Y_2	0	0	0	0	0	1	2/3	-3/2	0	0	0	125
δ_1	0	0	1	0	0	0	1/600	0	-1/200	0	0	1/4
δ_2	0	0	0	1	0	0	-1/300	1/200	0	-1/100	0	3/4
X_1	1	0	0	0	0	0	1/3	0	-1	0	0	50
X_2	0	1	0	0	0	0	-1/3	1/2	0	0	0	75
δ'	0	0	1	0	0	0	0	0	0	0	1	0
$\overline{C_j}$	0	0	0	0	0	0	215/3	455/2	30	45	0	47625
Y_1	0	0	0	0	1	0	0	-1/2	-1	0	-200	75
Y_2	0	0	0	0	0	1	0	-3/2	2	0	400	25
δ_1	0	0	1	0	0	0	0	0	0	0	1	0
δ_2	0	0	0	1	0	0	0	1/200	-1/100	-1/100	-2	5/4
X_1	1	0	0	0	0	0	0	0	0	0	200	0
X_2	0	1	0	0	0	0	0	1/2	-1	0	-200	125
Y_3	0	0	0	0	0	0	1	0	-3	0	-600	150
$\overline{C_j}$	0	0	0	0	0	0	0	455/2	245	45	43000	36875

Izvor: izrada autora rada

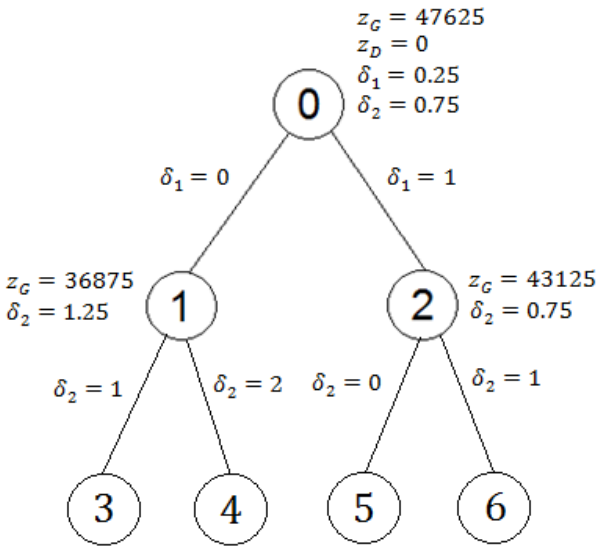
Iako su δ_1 i δ_2 konstante, pomoćne varijable su potrebne kako bi se mogle raditi transformacije nad matricom. Konkretno za podskup S_1 dodana je jednadžba $\delta_1 + \delta' = 0$, dok će za podskup S_2 jednadžba glasiti $-\delta_1 + \delta' = -1$.

Tablica 4.5. Podskup S_2

	X_1	X_2	δ_1	δ_2	Y_1	Y_2	Y_3	Y_4	Y_A	Y_B	δ'	b_i
Y_1	0	0	0	0	1	0	-1/3	-1/2	0	0	0	25
Y_2	0	0	0	0	0	1	2/3	-3/2	0	0	0	125
δ_1	0	0	1	0	0	0	1/600	0	-1/200	0	0	1/4
δ_2	0	0	0	1	0	0	-1/300	1/200	0	-1/100	0	3/4
X_1	1	0	0	0	0	0	1/3	0	-1	0	0	50
X_2	0	1	0	0	0	0	-1/3	1/2	0	0	0	75
δ'	0	0	-1	0	0	0	0	0	0	0	1	-1
$\overline{C_j}$	0	0	0	0	0	0	215/3	455/2	30	45	0	47625
Y_1	0	0	0	0	1	0	-1/3	-1/2	0	0	0	25
Y_2	0	0	0	0	0	1	2/3	-3/2	0	0	0	125
δ_1	0	0	1	0	0	0	0	0	0	0	-1	1
δ_2	0	0	0	1	0	0	-1/300	1/200	0	-1/100	0	3/4
X_1	1	0	0	0	0	0	0	0	0	0	-200	200
X_2	0	1	0	0	0	0	-1/3	1/2	0	0	0	75
Y_A	0	0	0	0	0	0	-1/3	0	1	0	-200	150
$\overline{C_j}$	0	0	0	0	0	0	245/3	455/2	0	45	6000	43125

Izvor: izrada autora rada

Nakon što su izračunata oba podskupa, rezultati se prikazuju na stablu odlučivanja:



Slika 4.2. Grananja druge razine

Izvor: izrada autora rada

Daljnje grananje nastavlja se po varijabli δ_2, skup S_1 dijeli se na podskupove prema $\delta_2 = 1$ i $\delta_2 = 2$, a skup S_2 na podskupove prema $\delta_2 = 0$ i $\delta_2 = 1$.

Tablica 4.6. Podskup S_3

	X_1	X_2	δ_1	δ_2	Y_1	Y_2	Y_3	Y_4	Y_A	Y_B	δ'	δ'_2	b_i
Y_1	0	0	0	0	1	0	0	-1/2	-1	0	-200	0	75
Y_2	0	0	0	0	0	1	0	-3/2	2	0	400	0	25
δ_1	0	0	1	0	0	0	0	0	0	0	1	0	0
δ_2	0	0	0	1	0	0	0	1/200	-1/100	-1/100	-2	0	5/4
X_1	1	0	0	0	0	0	0	0	0	0	200	0	0
X_2	0	1	0	0	0	0	0	1/2	-1	0	-200	0	125
Y_3	0	0	0	0	0	0	1	0	-3	0	-600	0	150
δ'_2	0	0	0	1	0	0	0	0	0	0	0	1	1
$\overline{C_j}$	0	0	0	0	0	0	0	455/2	245	45	43000	0	36875
Y_1	0	0	0	0	1	0	0	0	-2	-1	-400	-100	100
Y_2	0	0	0	0	0	1	0	0	-1	-3	-200	-300	100
δ_1	0	0	1	0	0	0	0	0	0	0	1	0	0
δ_2	0	0	0	1	0	0	0	0	0	0	0	1	1
X_1	1	0	0	0	0	0	0	0	0	0	200	0	0
X_2	0	1	0	0	0	0	0	0	0	1	0	100	150
Y_3	0	0	0	0	0	0	1	0	-3	0	-600	0	150
Y_4	0	0	0	0	0	0	0	1	-2	-2	-400	-200	50
$\overline{C_j}$	0	0	0	0	0	0	0	0	700	500	134000	45500	25500

Izvor: izrada autora rada

Kako varijabla δ_2 može poprimiti samo vrijednosti 0 i 1, skup S_4 je nemoguć te se desna strana grananja zanemaruje i prelazi se na grananje podskupova skupa S_2.

Tablica 4.7. Podskup S_5

	X_1	X_2	δ_1	δ_2	Y_1	Y_2	Y_3	Y_4	Y_A	Y_B	δ'	δ'_2	b_i
Y_1	0	0	0	0	1	0	-1/3	-1/2	0	0	0	0	25
Y_2	0	0	0	0	0	1	2/3	-3/2	0	0	0	0	125
δ_1	0	0	1	0	0	0	0	0	0	0	-1	0	1
δ_2	0	0	0	1	0	0	-1/300	1/200	0	-1/100	0	0	3/4
X_1	1	0	0	0	0	0	0	0	0	0	-200	0	200
X_2	0	1	0	0	0	0	-1/3	1/2	0	0	0	0	75
Y_A	0	0	0	0	0	0	-1/3	0	1	0	-200	0	150
δ'_2	0	0	0	1	0	0	0	0	0	0	0	1	0
$\overline{C_j}$	0	0	0	0	0	0	245/3	455/2	0	45	6000	0	43125
Y_1	0	0	0	0	1	0	-2/3	0	0	-1	0	-100	100
Y_2	0	0	0	0	0	1	-1/3	0	0	-3	0	-300	300
δ_1	0	0	1	0	0	0	0	0	0	0	-1	0	1
δ_2	0	0	0	1	0	0	0	0	0	0	0	1	0
X_1	1	0	0	0	0	0	0	0	0	0	-200	0	200
X_2	0	1	0	0	0	0	0	0	0	1	0	100	0
Y_A	0	0	0	0	0	0	-1/3	0	1	0	-200	0	150
Y_4	0	0	0	0	0	0	-2/3	1	0	-2	0	-200	150
$\overline{C_j}$	0	0	0	0	0	0	700/3	0	0	500	6000	45500	9000

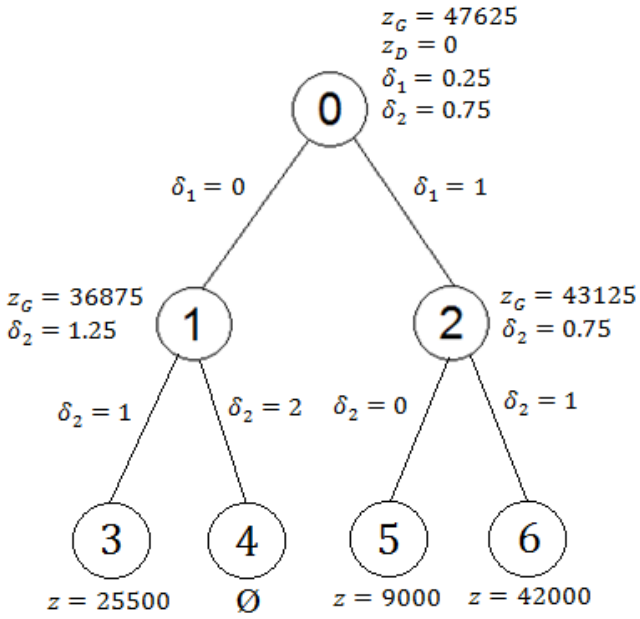
Izvor: izrada autora rada

Tablica 4.8. Podskup S_6

	X_1	X_2	δ_1	δ_2	Y_1	Y_2	Y_3	Y_4	Y_A	Y_B	δ'	δ'_2	b_i
Y_1	0	0	0	0	1	0	-1/3	-1/2	0	0	0	0	25
Y_2	0	0	0	0	0	1	2/3	-3/2	0	0	0	0	125
δ_1	0	0	1	0	0	0	0	0	0	0	-1	0	1
δ_2	0	0	0	1	0	0	-1/300	1/200	0	-1/100	0	0	3/4
X_1	1	0	0	0	0	0	0	0	0	0	-200	0	200
X_2	0	1	0	0	0	0	-1/3	1/2	0	0	0	0	75
Y_A	0	0	0	0	0	0	-1/3	0	1	0	-200	0	150
δ'_2	0	0	0	-1	0	0	0	0	0	0	0	1	-1
$\overline{C_j}$	0	0	0	0	0	0	245/3	455/2	0	45	6000	0	43125
Y_1	0	0	0	0	1	0	-1/3	-1/2	0	0	0	0	25
Y_2	0	0	0	0	0	1	2/3	-3/2	0	0	0	0	125
δ_1	0	0	1	0	0	0	0	0	0	0	-1	0	1
δ_2	0	0	0	1	0	0	0	0	0	0	0	-1	1
X_1	1	0	0	0	0	0	0	0	0	0	-200	0	200
X_2	0	1	0	0	0	0	-1/3	1/2	0	0	0	0	75
Y_A	0	0	0	0	0	0	-1/3	0	1	0	-200	0	150
Y_B	0	0	0	0	0	0	1/3	-1/2	0	1	0	-100	25
$\overline{C_j}$	0	0	0	0	0	0	200/3	250	0	0	6000	4500	42000

Izvor: izrada autora rada

Sada kad su svi podskupovi riješeni, dobiva se konačan izgled stabla grananja:



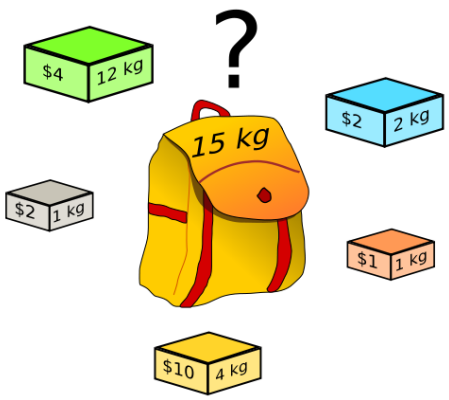
Slika 4.3. Završno stablo

Izvor: izrada autora rada

Slijedi interpretacija problema. Varijabla δ_1 označava da se proizvodi prvi proizvod, odnosno bombonijera, dok varijabla δ_2 označava da se proizvodi drugi proizvod, tj. čokolada. U slučaju da se proizvodi samo čokolada (skup S_3), ostvaruje se profit od 25500 kn. Ako se proizvodi samo bombonijera (skup S_5), ostvaruje se profit od 9000 kn. Ako se pak proizvode oba proizvoda (skup S_6), postiže se profit od 42000 kn, stoga slijedi zaključak da je najbolji proizvodni program onaj u kojem se proizvode i bombonijera i čokolada. Iz stupca b_i u tablici 4.8. iščitava se da se bombonijera proizvodi u 200 komada, a čokolada u 75 komada.

5. Problem ranca

Uz pretpostavku da postoji neko spremište i poveći broj predmeta, odnosno objekata, koje je moguće staviti u spremište, ali ne stanu svi, potrebno je staviti onoliko da se popuni kapacitet spremišta, ali i da vrijednost tih predmeta bude što je moguće veća. To spremište može biti bilo što, npr. kamionska prikolica, transportni kontejner, ili planinarski ruksak, odnosno ranac, odakle i dolazi naziv samog problema (eng. *Knapsack Problem*).



Slika 5.1. Ranac

Izvor: en.wikipedia.org

S. Martello i P. Toth (1990.) ovaj problem matematički definiraju na sljedeći način: najprije se objekti numeriraju od 1 do n te se uvodi vektor binarnih varijabli x_j, $j = 1,2,\dots,n$, za koje vrijedi sljedeće:

$$x_j = \begin{cases} 1, & \text{ako je predmet odabran,} \\ 0, & \text{ako predmet nije odabran.} \end{cases}$$

Nadalje, neka b predstavlja kapacitet ranca, c_j je sadašnja neto vrijednost objekta j, a a_j težinska vrijednost, odnosno veličina objekta j. Tada se problem formulira kao

$$\max z = \sum_{j=1}^{n} c_j x_j$$

uz sljedeće ograničenje

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

Ukratko, jednostavna definicija ovog problema prema (Z. Lukač, L. Neralić, 2012.): "Problem ranca sastoji se u izboru predmeta kojima će vrijednost biti maksimalna, s time da njihova ukupna težina ne premaši b jedinica."

Postupak rješavanja problema ranca prikazan je na primjeru utovara tereta i riješen je metodom grananja i ograđivanja. Radnici na utovaru stavljaju teret na kamionsku prikolicu ukupne nosivosti 45 tona. Materijali su nejednako zapakirani u blokove različite mase, i, ovisno o materijalu, svaki blok ima drugačiju vrijednost. Neto vrijednost i masa pojedinog bloka prikazani su u sljedećoj tablici:

Tablica 5.1. Opis problema

	blok 1	blok 2	blok 3	blok 4	blok 5	blok 6	blok 7
c_j	50	20	40	60	60	40	70
a_j	12	9	16	13	8	10	18

Izvor: izrada autora rada

Ako se blokovi zamijene varijablama x_j, funkcija cilja poprima sljedeći oblik:

$$\max z = 50x_1 + 20x_2 + 40x_3 + 60x_4 + 60x_5 + 40x_6 + 70x_7,$$

uz ograničenje

$$12x_1 + 9x_2 + 16x_3 + 13x_4 + 8x_5 + 10x_6 + 18x_7 \leq 45$$

$$x_n \in \{0,1\}, \qquad n = 1,2,\dots,7.$$

U tablici 4.1. redak c_j predstavlja sadašnju neto vrijednost bloka u tisućama kuna, dok a_j predstavlja masu bloka u tonama. Masa svih blokova ukupno iznosi 86 tona, što uvelike prelazi dozvoljenu nosivost prikolice, iz čega je vidljivo da ne mogu svi blokovi stati na prikolicu. Zbog toga je potrebno odabrati blokove koji će se utovariti. Najprije je potrebno izračunati koeficijente važnosti c_j/a_j za svaki blok.

Tablica 5.2. Težinski koeficijenti

	blok 1	blok 2	blok 3	blok 4	blok 5	blok 6	blok 7
c_j/a_j	4.17	2.22	2.5	4.62	7.5	4	3.89

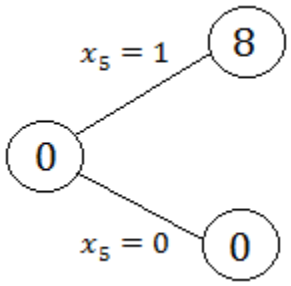
Izvor: izrada autora rada

Potom se problem rješava po metodi grananja i ograđivanja, odnosno izrađuje se stablo odlučivanja. Odabire se varijabla (u ovom slučaju blok) koja ima najveći koeficijent važnosti, što je u ovom slučaju blok br. 5, te se po varijabli x_5 radi početno grananje.

Tablica 5.3. Varijable poredane po važnosti

	blok 5	blok 4	blok 1	blok 6	blok 7	blok 3	blok 2
var.	X_5	X_4	X_1	X_6	X_7	X_3	X_2
c_j/a_j	7.5	4.62	4.17	4	3.89	2.5	2.22

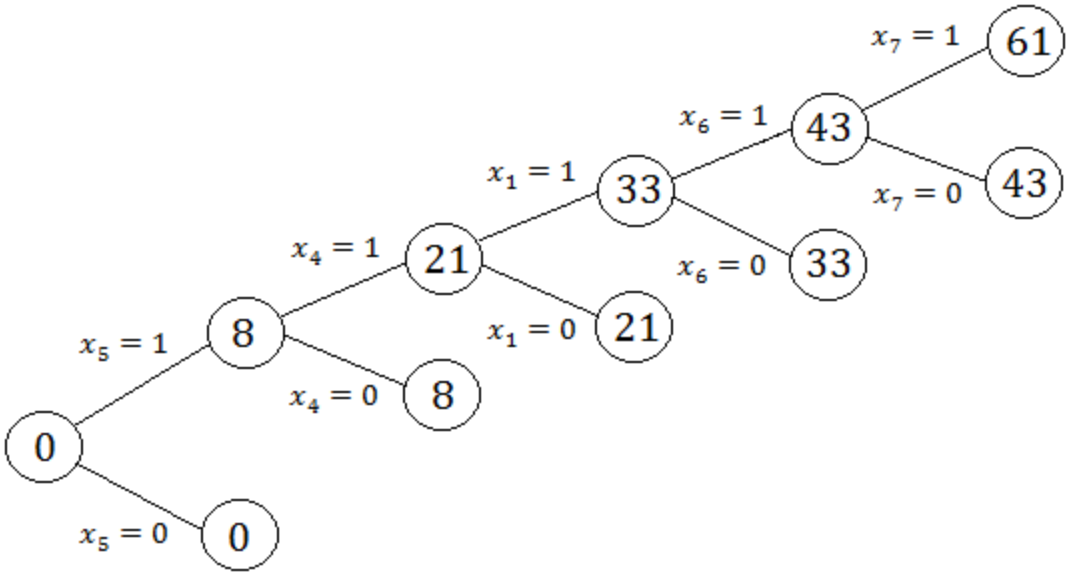
Izvor: izrada autora rada



Slika 5.2. Početno grananje

Izvor: izrada autora rada

Standardno grananje odvija se s operatorima uspoređivanja $\leq$ i $\geq$. Budući da ovdje varijable x_j mogu poprimiti samo vrijednosti 0 ili 1, koristi se znak jednakosti. Kako se ne bi isprobavale sve moguće kombinacije, kojih ima 2^n, odnosno u ovom slučaju $2^7 = 128$, pravilo nalaže da se prema redoslijedu važnosti u ranac stavljaju objekti dok se ne pređe ograničenje, a potom se za prvu varijablu x_k koja ne stane u ranac uzme razlika ograničenja i trenutne popunjenosti ranca proporcionalna težinskoj vrijednosti varijable x_k. Tako se za funkciju cilja dobije rješenje koje nije konačno i može biti necjelobrojno. Uvjet za konačno rješenje je da je iznos rješenja djeljiv s najvećom zajedničkom mjerom svih neto vrijednosti (D. Barković, 2001.), što u ovom slučaju iznosi 10. Rješenje je najveći cijeli broj manji od prvog dobivenog rješenja koji zadovoljava taj uvjet.



Slika 5.3. Stablo odlučivanja s pređenom granicom

Izvor: izrada autora rada

Kao što je vidljivo, nakon dodavanja varijable x_7 premašuje se granica b, stoga x_7 ne može u cijelosti ući u ranac. Varijable koje ulaze u potpunosti su x_5, x_4, x_1 i x_6, njihove težinske vrijednosti zbrojeno iznose 43, što je za 2 manje od dozvoljene granice. Prema gore navedenom pravilu, u ranac se može staviti $\frac{2}{18} = \frac{1}{9}$ varijable x_7, te funkcija cilja iznosi

$$\max z = 60 + 60 + 50 + 40 + \frac{1}{9} \cdot 70 = 217.77.$$

Kako rješenje mora biti djeljivo s 10, najveći cijeli broj koji to zadovoljava je 210. Budući da više nema varijable koja se u cijelosti može staviti u ranac, odnosno koja ima težinsku vrijednost manju ili jednaku 2, provjerava se funkcija cilja za već ubačene varijable, koja iznosi 210. Stoga je konačno rješenje $x_1 = x_4 = x_5 = x_6 = 1$ i $x_2 = x_3 = x_7 = 0$, što se u ovom primjeru može interpretirati da će na prikolicu biti utovareni blokovi br. 1, 4, 5 i 6, dok za blokove br. 2, 3 i 7 na prikolici nema mjesta. Na prikolicu će biti utovaren teret od 43 tone, a vrijednost tereta iznosi 210 000 kn.

6. Problem optimalne investicijske odluke

Poslovanje određenih poduzeća svodi se na investiranje u projekte te ulaganje u provedbu tih projekata. Nakon završetka prethodno prihvaćenog projekta potrebno je investirati u nove projekte, što nerijetko bude i više projekata istovremeno. Kako ponuda za projekte ima više nego što pojedina tvrtka može prihvatiti u tom trenutku, potrebno je odabrati one projekte koji su profitabilniji.

Slika 6.1. Razmišljanje o ulaganju
Izvor: ITLever™, www.itlever.com

Problemi financijskog investiranja rješavaju se budžetiranjem kapitala, što je stalna praktična djelatnost posredničkih financijskih tvrtki. S. Orsag (2002.) budžetiranje kapitala definira kao postupak donošenja odluka o dugoročnim investicijama u realnu poslovnu imovinu poduzeća. Orsag cjelokupan postupak budžetiranja kapitala dijeli na šest koraka:

1. pronalaženje investicijskih oportuniteta,
2. prikupljanje podataka,
3. određivanje novčanih tokova investicijskih oportuniteta,
4. određivanje budžeta kapitala,
5. vrednovanje i donošenje odluke,
6. analiza izvođenja i prilagođavanja.

Problem optimalne investicijske odluke u pogledu linearnog programiranja odnosi se na korake 3-5, dok su ostali koraci ekonomske prirode.

Uz pretpostavku da su projekti nezavisni, odnosno svaki projekt je zaseban te nema utjecaja na neki drugi projekt ili mu nije preduvjet, problem optimalne investicijske odluke odnosi se na izbor projekata za koje će sadašnja neto vrijednost biti maksimalna, uz ograničena raspoloživa sredstva kroz godine izvođenja projekta. (Z. Lukač, L. Neralić, 2012.)

Potrebno je uvesti sljedeće oznake:

c_j – sadašnja neto vrijednost projekta $j,\ j = 1,2,\dots,n$

a_{ij} – sadašnja vrijednost troškova projekta j u godini/razdoblju $i,\ i = 1,2,\dots,m,\ j = 1,2,\dots,n$

b_i – sadašnja vrijednost gornje ograde na budžet za godinu/razdoblje $i,\ i = 1,2,\dots,m$

n – broj projekata

x_j – cjelobrojna varijabla $j = 1,2,\dots,n$ za koju vrijedi

$$x_j = \begin{cases} 1, & \text{ako je projekt } j \text{ prihvaćen} \\ 0, & \text{ako projekt } j \text{ nije prihvaćen} \end{cases}$$

Problem se može forumlirati kao

$$\max z = \sum_{j=1}^{n} c_j x_j$$

uz ograničenja

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \qquad i = 1,2,\dots,m$$

$$0 \leq x_j \leq 1, \qquad x_j \in \mathbb{Z}, \qquad j = 1,2,\dots,n.$$

U nastavku je prikazan primjer problema izbora optimalne investicije riješen metodom odsijecajućih ravnina. Građevinska tvrtka dobila je ponude za izvođenje četiri različita projekta, od kojih bi izvedba svakoga trajala tri godine. Tvrtka nije u mogućnosti prihvatiti sva četiri projekta, jer ima ograničen godišnji budžet koji smije potrošiti na ulaganje u izgradnju.

Slika 6.2. Građevinski projekti

Izvor: Buildforia, www.buildforia.com.au

Godišnji troškovi projekata kroz nisu uvijek isti, kako zbog tečajne razlike, tako i zbog različite prirode ulaganja: u prvoj godini potrebno je obaviti otkup tržišta i pribaviti sve potrebne građevinske dozvole, u drugoj godini najviše se orijentira na samu izgradnju, dok je u trećoj godini potrebno dovršiti gradnju te urediti eksterijer i interijer. Projekti se nalaze u sljedećoj tablici, svi iznosi su u milijunima kuna.

Tablica 6.1. Početni troškovi projekta

	Stambena zgrada	Sportska dvorana	Autobusni kolodvor	Benzinska crpka	Raspoloživi iznos
Godina 1	10	8	9	8	28
Godina 2	9	10	7	8	26
Godina 3	7	12	8	11	30
Vrijednost projekta	42	32	39	44	

Izvor: izrada autora rada

Označe li se projekti varijablama x_j gdje j predstavlja broj projekta, dobije se funkcija cilja

$$\max z = 42x_1 + 32x_2 + 39x_3 + 44x_4$$

uz ograničenja

$$10x_1 + 8x_2 + 9x_3 + 8x_4 \leq 28$$
$$9x_1 + 10x_2 + 7x_3 + 8x_4 \leq 26$$
$$7x_1 + 12x_2 + 8x_3 + 11x_4 \leq 30$$
$$0 \leq x_j \leq 1, \qquad x_j \in \mathbb{Z}, \qquad j = 1, 2, 3, 4.$$

Tablica 6.2. Početni izgled simpleks matrice

X_B	X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	b_i
Y_1	10	8	9	8	1	0	0	0	0	0	0	28
Y_2	9	10	7	8	0	1	0	0	0	0	0	26
Y_3	7	12	8	11	0	0	1	0	0	0	0	30
Y_4	1	0	0	0	0	0	0	1	0	0	0	1
Y_5	0	1	0	0	0	0	0	0	1	0	0	1
Y_6	0	0	1	0	0	0	0	0	0	1	0	1
Y_7	0	0	0	1	0	0	0	0	0	0	1	1
$\overline{C_j}$	-42	-32	-39	-44	0	0	0	0	0	0	0	0

Izvor: izrada autora rada

Tablica 6.3. Prvo rješenje

X_B	X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	b_i
X_2	0	1	0	0	1/8	0	0	-5/4	0	-9/8	-1	1/8
Y_2	0	0	0	0	-5/4	1	0	7/2	0	17/4	2	3/4
Y_3	0	0	0	0	-3/2	0	1	8	0	11/2	1	5/2
X_1	1	0	0	0	0	0	0	1	0	0	0	1
Y_5	0	0	0	0	-1/8	0	0	5/4	1	9/8	1	7/8
X_3	0	0	1	0	0	0	0	0	0	1	0	1
X_4	0	0	0	1	0	0	0	0	0	0	1	1
$\overline{C_j}$	0	0	0	0	4	0	0	2	0	3	12	129

Izvor: izrada autora rada

Iz tablice je vidljivo da je rješenje cjelobrojno i iznosi 129, no neke varijable su decimalne. Interpretacijom rješenja dolazi se do zaključka da se projekti 1, 3 i 4 izvode u potpunosti, dok će se od projekta 2 izvesti samo osmina, odnosno izvest će se 12,5% projekta 2. Kako je takvo rješenje neprihvatljivo, potrebno je pronaći drugo rješenje, u kojem će sve varijable biti cjelobrojne.

Kako je objašnjeno u poglavlju metode cjelobrojnih formi, traži se redak pomoću kojeg će se generirati odsijecajuća ravnina. Kako vrijedi $\max(f_{10}, f_{20}, f_{30}, f_{50}) = f_{50} = \frac{1}{8}$, formira se nova odsijecajuća ravnina čija jednadžba glasi

$$-\frac{7}{8}y_1 - \frac{1}{4}y_2 - \frac{1}{8}y_6 + s_1 = -\frac{7}{8}$$

te se uvrštava u simpleks matricu i dalje rješava.

Tablica 6.4. Novi početni izgled simpleks matrice

X_B	X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	S_1	b_i
X_2	0	1	0	0	1/8	0	0	-5/4	0	-9/8	-1	0	1/8
Y_2	0	0	0	0	-5/4	1	0	7/2	0	17/4	2	0	3/4
Y_3	0	0	0	0	-3/2	0	1	8	0	11/2	1	0	5/2
X_1	1	0	0	0	0	0	0	1	0	0	0	0	1
Y_5	0	0	0	0	-1/8	0	0	5/4	1	9/8	1	0	7/8
X_3	0	0	1	0	0	0	0	0	0	1	0	0	1
X_4	0	0	0	1	0	0	0	0	0	0	1	0	1
S_1	0	0	0	0	-7/8	0	0	-1/4	0	-1/8	0	1	-7/8
$\overline{C_j}$	0	0	0	0	4	0	0	2	0	3	12	0	129

Izvor: izrada autora rada

Tablica 6.5. Novo rješenje

X_B	X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	S_1	b_i
X_2	0	1	0	0	0	0	0	-9/7	0	-8/7	-1	1/7	0
Y_2	0	0	0	0	0	1	0	27/7	0	31/7	2	-10/7	2
Y_3	0	0	0	0	0	0	1	59/7	0	40/7	1	-12/7	4
X_1	1	0	0	0	0	0	0	1	0	0	0	0	1
Y_5	0	0	0	0	0	0	0	9/7	1	8/7	1	-1/7	1
X_3	0	0	1	0	0	0	0	0	0	1	0	0	1
X_4	0	0	0	1	0	0	0	0	0	0	1	0	1
Y_1	0	0	0	0	1	0	0	2/7	0	1/7	0	-8/7	1
$\overline{C_j}$	0	0	0	0	0	0	0	6/7	0	17/7	12	32/7	125

Izvor: izrada autora rada

Ovoga puta sve varijable su cjelobrojne te je postignuto optimalno rješenje. Varijable koje označavaju projekte iznose $x_1 = x_3 = x_4 = 1$ i $x_2 = 0$, što znači da tvrtka treba prihvatiti poslove izgradnje stambene zgrade, autobusnog kolodvora i benzinske crpke, dok se projekt izgradnje sportske dvorane neće prihvatiti. Kako je iznos rješenja u milijunima kuna, tvrtka će od tri prihvaćena projekta uprihoditi 125,000,000.00 kn.

7. Zaključak

Cjelobrojno programiranje poseban je oblik matematičkog programiranja u kojem se za varijable postavlja zahtjev da moraju poprimiti cjelobrojne vrijednosti, odnosno zahtjev cjelobrojnosti. Taj zahtjev ne mora biti postavljen na sve varijable, i u tom slučaju riječ je o mješovito cjelobrojnom programiranju. Binarno programiranje jedan je poseban oblik cjelobrojnog programiranja, u kojem varijable mogu poprimiti samo vrijednosti 0 ili 1.

Prvu metodu za rješavanje problema cjelobrojnog programiranja razvio je R. E. Gomory 1958. godine. Ta metoda temeljena je na generiranju odsijecajućih ravnina, pri čemu svaka ravnina smanjuje mogući raspon rješenja tako što iz skupa mogućih rješenja izbaci neka necjelobrojna. Druga metoda je metoda grananja i ograđivanja, razvijena par godina kasnije i temeljena na izradi stabla odlučivanja. Prema varijablama za koje su dobivene decimalne vrijednosti vrši se grananje tako da se varijabla zaokruži na prvi manji i prvi veći cijeli broj. Nema određenog pravila koja metoda je brža i jednostavnija, jer se kod obje metode rješenje može postići vrlo brzo, ali se može i rastegnuti u nekoliko koraka.

Problem fiksnih troškova govori o troškovima koji nastaju kad neka varijabla poprimi iznos veći od nule, taj problem bitan je kod proizvodnog procesa, kako bi se odredio optimalni program proizvodnje. Problem optimalne investicijske odluke je problem binarnog programiranja, gdje se donose da-ne odluke, odnosno varijabla može imati vrijednost 1 ili 0. Problem ranca također spada u probleme binarnog programiranja, određujemo hoće li se neka stvar naći u rancu ili neće. Kod tih problema najbitnije je znati u početku odrediti funkciju cilja i ograničenja, kako bi se mogla napraviti početna simpleks matrica. Problem fiksnih troškova i problem optimalne investicijske odluke gotovo su identični u rješavanju, oba problema mogu se riješiti na obadvije metode. Problem ranca ipak je jednostavnije riješiti metodom grananja i ograđivanja, u protivnom bi se za svaku varijablu x_j moralo dodavati ograničenje $x_j + y_j = 1$, kao što se radi kod problema optimalne investicije. Budući da problem ranca ima samo jedno ograničenje, metodom stabla odlučivanja može se riješiti i bez provođenja simpleks postupka.

8. Literatura

[1] Barković D. (2001.) *Operacijska istraživanja*. Sveučilište Josipa Jurja Strossmayera – Ekonomski fakultet u Osijeku

[2] Dakin R. J. (1965.) *A tree-search algorithm for mixed integer programming problems*. The Computer Journal, Issue 3, Volume 8. Preuzeto s https://academic.oup.com/comjnl/article/8/3/250/336099 na dan 8.11.2017.

[3] Kalpić D., Mornar V. (1996.) *Operacijska istraživanja*. Zagreb: Društvo za razvoj informacijske pismenosti

[4] Lukač Z., Neralić L. (2012.) *Operacijska istraživanja* (udžbenik Sveučilišta u Zagrebu). Zagreb: Element d.o.o.

[5] Gomory R. E. (1958.) *Outline of an algorithm for integer solutions to linear programs*. Princeton university. Preuzeto s https://projecteuclid.org/euclid.bams/1183522679 na dan 26.10.2017.

[6] Martello S., Toth P. (1990.) *Knapsack problems*. DEIS, University of Bologna

[7] Martić Lj. (1979.) *Matematičke medote za ekonomske analize, II. svezak* (udžbenik Sveučilišta u Zagrebu). Zagreb: Narodne novine

[8] Mikić M., Orsag S., Pološki Vokić N., Švaljek S. (2011.) *Ekonomski leksikon*. Zagreb: Leksikografski zavod "Miroslav Krleža" – Masmedia

[9] Murty, K. G. (2012.) *The Branch and Bound Approach: A personal account*, IOE Dept., U. of Michigan. Preuzeto s http://www-personal.umich.edu/~murty/B&BHistory.pdf na dan 7.11.2017.

[10] Orsag S. (2001.) *Budžetiranje kapitala*. Zagreb: Masmedia

[11] Petrić J., Kojić Z., Šarenac L. (1978.): *Operaciona istraživanja – zbirka rešenih zadataka*. Beograd: Univerzitet u Beogradu

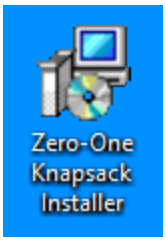
9. Prilog – aplikacija “Zero-One Knapsack”

Zero-One Knapsack aplikacija je koja se bavi rješavanjem problema ranca, objašnjenog u petom poglavlju. Korisnik na početku unosi broj ukupnih varijabli i ograničenje ranca, potom za svaku varijablu pojedinačno unosi neto i težinsku vrijednost. Aplikacija radi na principu težinskih koeficijenata, kako je prikazano u tablici 5.3. (str. 30), odnosno varijable se poredaju po težinskim koeficijentima, te se ranac popunjava dok se ne premaši njegov kapacitet. Vrijednost ranca predstavlja zbroj neto vrijednosti varijabli u rancu.

Slika 9.1. Logo aplikacije

Izvor: izrada autora rada na stranici IconsFlow

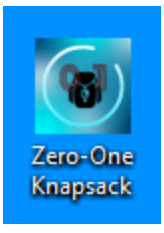
9.1. Instalacija aplikacije



Za početak instalacije aplikacije potrebno je pokrenuti instalacijsku datoteku kao na slici 9.2. Postupak instalacije provodi se identično kao za svaku drugu aplikaciju.

Slika 9.2. Instalacijska datoteka

Izvor: izrada autora rada



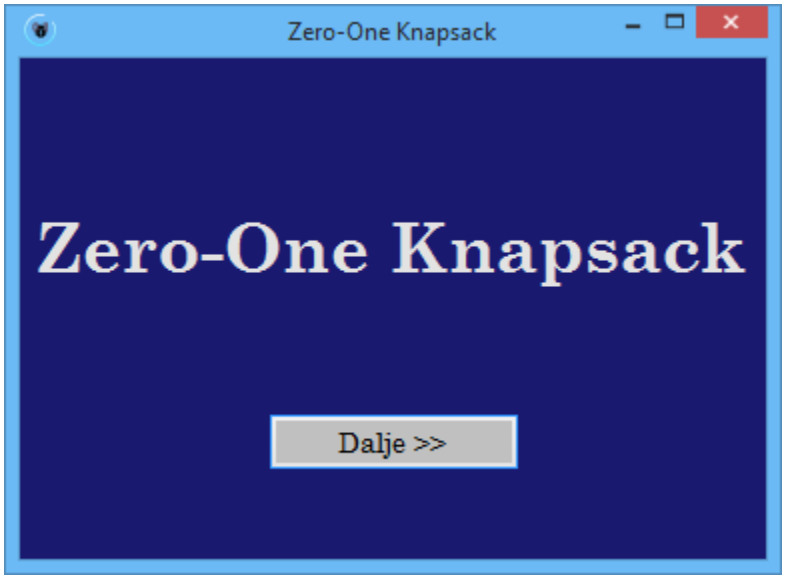
Nakon završetka instalacije na radnoj površini pojavit će se ikona kao na slici 9.3 i preko te ikone se pokreće aplikacija.

Slika 9.3. Ikona za pokretanje aplikacije

Izvor: izrada autora rada

9.2. Korištenje aplikacije

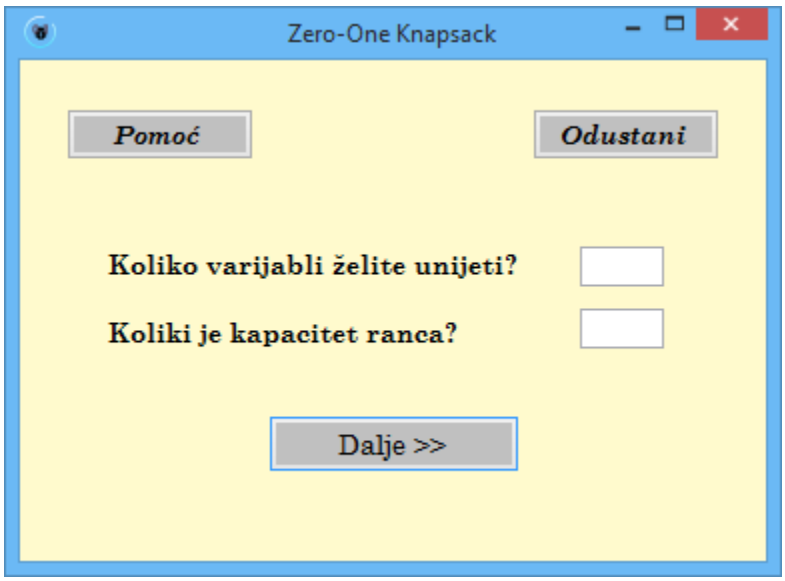
Nakon pokretanja aplikacije otvori se forma kao na sljedećoj slici:



Slika 9.4. Početna forma

Izvor: izrada autora rada

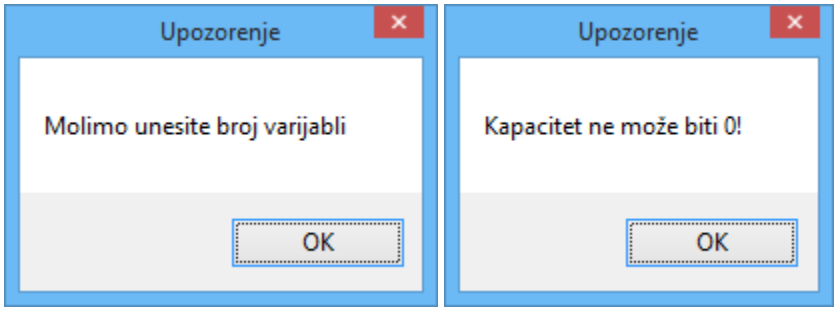
Klikom na gumb *Dalje* forma se mijenja u oblik prilagođen za unos podataka:



Slika 9.5. Forma za unos podataka

Izvor: izrada autora rada

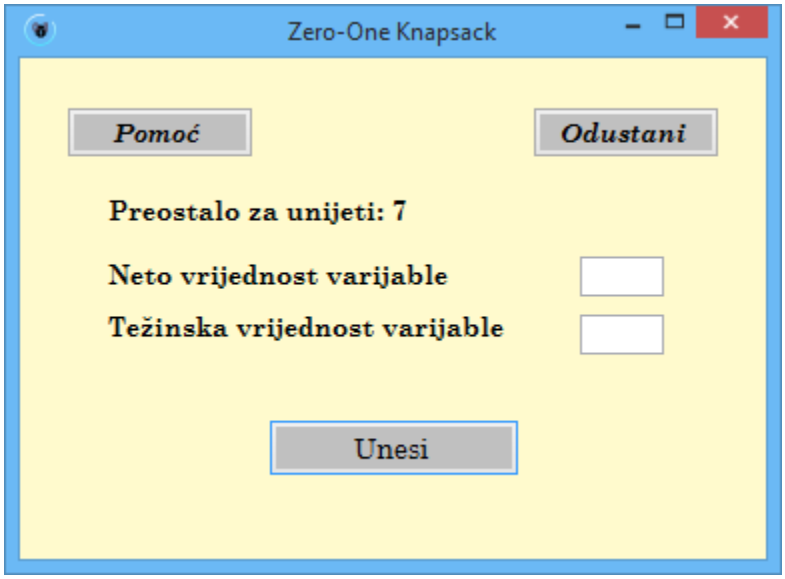
U prethodno prikazanoj formi unosi se broj varijabli koliko će se staviti u ranac, te kapacitet ranca, nakon čega se ponovno odabere gumb *Dalje*. Potrebno je unijeti cijele brojeve veće od 0, u suprotnom izbacuje poruke upozorenja: ako se vrijednost ne unese, ispiše se poruka "Molimo unesite ... !", ako se pak unese vrijednost 0, poruka glasi "... ne može biti 0!"



Slika 9.6. Upozorenja: a) ako se polje ostavi prazno, b) ako se unese vrijednost 0

Izvor: izrada autora rada

Nakon što se unesu ispravne vrijednosti i odabere gumb *Dalje*, prelazi se na unos varijabli. Iznad polja za unos vrijednosti uvijek piše koliko je varijabli ostalo za unijeti. Također je potrebno unijeti pozitivne cijele brojeve, u protivnom aplikacija izbacuje poruke upozorenja identične onima kod prethodnog unosa.



Slika 9.7. Unos varijabli

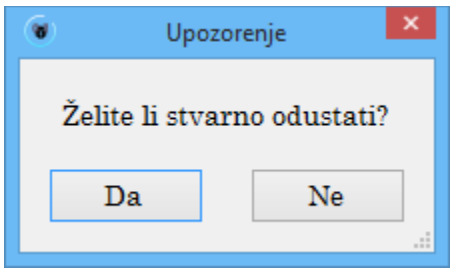
Izvor: izrada autora rada

Nakon što se unese i posljednja varijabla, aplikacija prikazuje rezultat i varijable koje se nalaze u rancu, poredane po težinskom koeficijentu. Klikom na gumb *Vrati se na početak* ponovno se dolazi na početnu formu sa slike 9.4.

Slika 9.8. Prikaz rješenja

Izvor: izrada autora rada

Kako se može primijetiti na slikama 9.5. i 9.7., u gornjem dijelu forme nalaze se opcije *Pomoć* i *Odustani*, koje su dostupne tijekom cijelog procesa unošenja početnih vrijednosti i varijabli. Klikom na gumb *Odustani* moguće je prekinuti proces unošenja podataka, te krenuti ispočetka.



Slika 9.9. Prekid izračuna

Izvor: izrada autora rada

Ukoliko korisnik odabere "Da", aplikacija ga vraća na početnu stranicu sa slike 9.4. Pritiskom na "Ne" vraća se natrag u trenutni proces izračuna.

Klikom na gumb Pomoć prikazuje se stranica s uputama za korištenje aplikacije.

Slika 9.10. Stranica za pomoć

Izvor: izrada autora rada

9.3. Korištene tehnologije

- pisanje koda – programski jezik C#, radno okruženje Visual Studio 2017
- izrada ikone – IconsFlow – Free Icon Maker, https://iconsflow.com/
- izrada instalacije – Inno Setup