Univerzitet u Nišu
Prirodno-matematički fakultet

MODIFIKACIJE METODA MATEMATIČKOG PROGRAMIRANJA I PRIMENE

Mentor:
Dr. Predrag S. Stanimirović,
redovni profesor

Kandidat:
Marko D. Petković,
DexterOfNis@gmail.com

1 Uvod

- Matematičko programiranje je deo široke naučne oblasti poznate pod nazivom Operaciona istraživanja
- Prvi rad, kako iz matematičkog programiranja objavio je ruski matematičar Kantorovič

L. V. Kantorovič (1912-1986)

- Još 1940. godine, pukovnik Vlastimir Ivanović objavio je rad "Pravila za proračun potrebnog broja transportnih sredstava"
- Pukovnik Ivanović je bio prvi kod nas i medju prvima u svetu koji se bavio matematičkim programiranjem.

Problemi matematičkog programiranja javljaju se u različitim disciplinama.

- Menadžer na berzi mora da odabere ulaganja koja će generisati najveći mogući profit a da pri tome rizik od velikih gubitaka bude na unapred zadatom nivou.
- Menadžer proizvodnje organizuje proizvodnju u fabrici tako da količina proizvoda i kvalitet budu maksimalni a utrošak materijala i vremena i škart minimalni. Pritom ima na raspolaganju ograničene resurse (broj rednika, kapacitet mašina, radno vreme).
- Naučnik pravi matematički model fizičkog procesa koji najbolje opisuje odredjenu fizičku pojavu, a na raspolaganju ima konačni broj mernih rezultata.

Znači, da bi zadali problem matematičkog programiranja moramo:

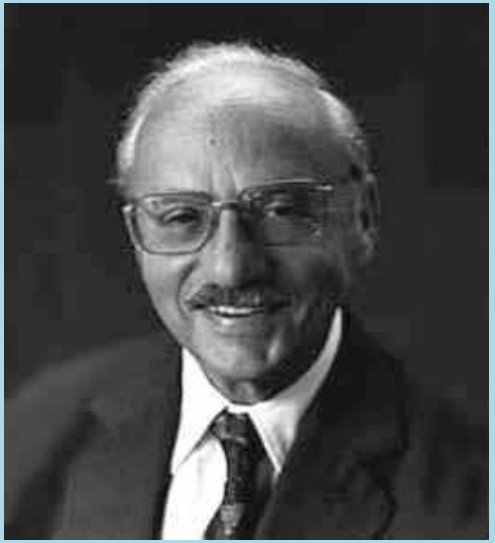
1. Odabrati jednu ili više optimizacionih promenljivih,
2. Odabrati funkciju cilja i
3. Formirati skup ograničenja.

U ovom radu detaljno ćemo proučiti dve klase problema matematičkog programiranja: linearno programiranje i višekriterijumsku optimizaciju.

- Zadatak linearnog programiranja je da odredi maksimum (minimum) linearne funkcije koja zavisi od više promenljivih pod uslovom da su neke od ovih promenljivih nenegativne i da zadovoljavaju linearna ograničenja u obliku jednačina i/ili nejednačina.

- Do prvog opšteg metoda za rešavanje problema linearnog programiranja došao je američki matematičar Dancig 1947. godine.

- Suština Dancingovog Simpleks metoda je da se izmedju $\binom{n}{m}$ bazičnih rešenja pronadje ono koje maksimizira ili minimizira linearnu ciljnu funkciju.



G. B. Dantzig (1914-2005)

1947. Dancig je otkrio prvi opšti metod (simpleks metod).

1972. Minti i Kli su dokazali da je složenost simpleks metoda eksponencijalna

1979. Hačijan je konstruisao prvi polinomijalni metod (metod elipse).

1984. Karmakar je otkrio prvi polinomijalni metod koji je praktično primenljiv.

Danas su primal-dual metodi unutrašnje tačke dominantni za rešavanje problema linearnog programiranja.

Medjutim, postoje primeri na kojima je simpleks metod bolji od primal-dual metoda (npr kod loše uslovljenih primera).

- Iako je linearno programiranje veoma primenljivo u praksi, mnoge probleme iz prakse je nemoguće adekvatno linearizovati a da se pritom drastično ne izgubi na tačnosti.
- Osim nelinearnosti, u mnogim problemima je potrebno naći optimum više od jedne funkcije cilja.
- U tom slučaju, moramo rešavati problem višekriterijumske optimizacije
- Postoji više metoda za rešavanje problema višekriterijumske optimizacije
- Zajedničko svim tim metodima je da se polazni problem na odgovarajući način svodi na problem linearnog i nelinearnog programiranja.

2 Problem linearnog programiranja

- Definisaćemo problem linearnog programiranja i navesti osnovne osobine skupa dopustivih rešenja.
- Formulisaćemo geometrijski i simpleks metod.
- Detaljno ćemo proučiti dve faze simpleks metoda.
- Izračunaćemo složenost simpleks metoda.
- Formulisaćemo revidirani simpleks metod.

2.1 Definicija i osnovna svojstva

Problem linearnog programiranja se zadaje na sledeći način:

$$\max(\min)\ f(x) = c_1 x_1 + \cdots + c_n x_n$$

$$N_i^{(1)}: \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1, \ldots, p$$

$$N_i^{(2)}: \quad \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \quad i = p+1, \ldots, q$$

$$J_i: \quad \sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = q+1, \ldots, m$$

$$x_j \geq 0, \quad j \in \mathcal{J} = \{1, \ldots, s\}, \quad s \leq n$$

Promenljive $x_{s+1}, \ldots x_n$ za koje nije ispunjen uslov nenegativnosti nazivamo slobodne promenljive.

- Rešenja $x = (x_1, \ldots, x_n)$ sistema nazvaćemo dopustivim rešenjima a njihov skup označićemo sa Ω_P.
- U geometrijskoj interpretaciji, svako rešenje $x \in \Omega_P$ predstavlja tačku n-dimenzionalnog prostora $\mathbb{R}^n$.

Ukoliko su sva ograničenja jednačine, dobijamo standardni oblik:

$$\begin{aligned} &\min c^T x, \\ &Ax = b, \\ &x \geq 0, \end{aligned}$$

Ukoliko su sva ograničenja jednačine, dobijamo simetrični oblik:

$$\begin{aligned} &\min c^T x, \\ &Ax \geq b, \\ &x \geq 0. \end{aligned}$$

Primer 1 Fabrika proizvodi dve vrste artikala A_1 i A_2, i to na mašinama M_1 i M_2. Za artikal A_1 mašina M_1 radi 2^h, a mašina M_2 radi 4^h, i za vrstu A_2 mašina M_1 radi 4^h, a mašina M_2 radi 2^h. Fabrika dobija 3500 dinara po jedinici proizvoda A_1, a 4800 dinara po jedinici proizvoda A_2. Koliko treba proizvoditi artikala A_1 i A_2 i kako iskoristiti rad mašina M_1 i M_2 da dnevna dobit fabrike bude maksimalna?

Rešenje: Neka je x_1 broj proizvedenih artikala A_1, a x_2 broj proizvedenih artikala A_2 u toku dana. Tada je dnevna dobit fabrike:

$$f(x) = 3500x_1 + 4800x_2$$

uz uslove:

$$\begin{aligned} 2x_1 + 4x_2 &\leq 24 \\ 4x_1 + 2x_2 &\leq 24 \\ x_1 \geq 0, x_2 &\geq 0. \end{aligned}$$

Ovim smo dobili simetrični oblik.

Ako sada uvedemo slack promenljive x_3 i x_4 dobijamo ekvivalentan problem u standardnom obliku:

$$\begin{aligned} \max f(x) &= 3500x_1 + 4800x_2 \\ 2x_1 + 4x_2 - 24 &= -x_3 \\ 4x_1 + 2x_2 - 24 &= -x_4 \\ x_1 \geq 0, x_2 &\geq 0. \end{aligned}$$

odnosno u matričnoj formi:

$$A = \begin{bmatrix} 2 & 4 & 1 & 0 \\ 4 & 2 & 0 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 24 \\ 24 \end{bmatrix} \quad c = \begin{bmatrix} 3500 \\ 4800 \end{bmatrix}$$

2.2 Osobine skupa Ω_P

Neka je problem linearnog programiranja zadat u standardnom obliku. Sada je $\Omega_P = \{x \mid Ax = b,\ x \geq 0\}$.

Teorema 1 Skup $\Omega_P = \{x \mid Ax = b,\ x \geq 0\}$ je konveksan.

Označimo sa K_i i-tu kolonu matrice A ($A_{\bullet i}$).

Definicija 1 Rešenje x, sistema $Ax = b$ je osnovno (bazično) ako su u jednačini

$$b = x_1 K_1 + \cdots + x_n K_n$$

vektori K_i za koje je $x_i \neq 0$ linearno nezavisni.

Definicija 2 Matrica $A_B = [K_{i_1} \cdots K_{i_m}]$ je osnovna (bazična) ako je regularna. Preostale kolone matrice A formiraju nebazičnu matricu A_N. Promenljive $x_{i_1}, \ldots x_{i_m}$ nazivamo bazičnim dok preostale promenljive nazivamo nebazičnim. Dve bazične matrice su susedne ako se razlikuju u jednoj koloni.

$$A_B x_B + A_N x_N = b \Longrightarrow x_B = A_B^{-1} b - A_B^{-1} A_N x_N$$

Definicija 3 Tačka x je ekstremna tačka konveksnog skupa Ω_P ako za nju važi:

$$x = \lambda x^{(1)} + (1-\lambda)x^{(2)} \wedge \lambda \in [0,1] \Leftrightarrow x = x^{(1)} = x^{(2)}$$

Teorema 2 Ako je $\Omega_P \neq \emptyset$, tada on sadrži bar jedno bazično rešenje.

Teorema 3 Ako je $x \in \Omega_P$ bazično rešenje sistema $Ax = b, x \geq 0$, tada je x ekstremna tačka skupa Ω_P. Obratno, ako je x ekstremna tačka skupa Ω_P, tada je x bazično rešenje.

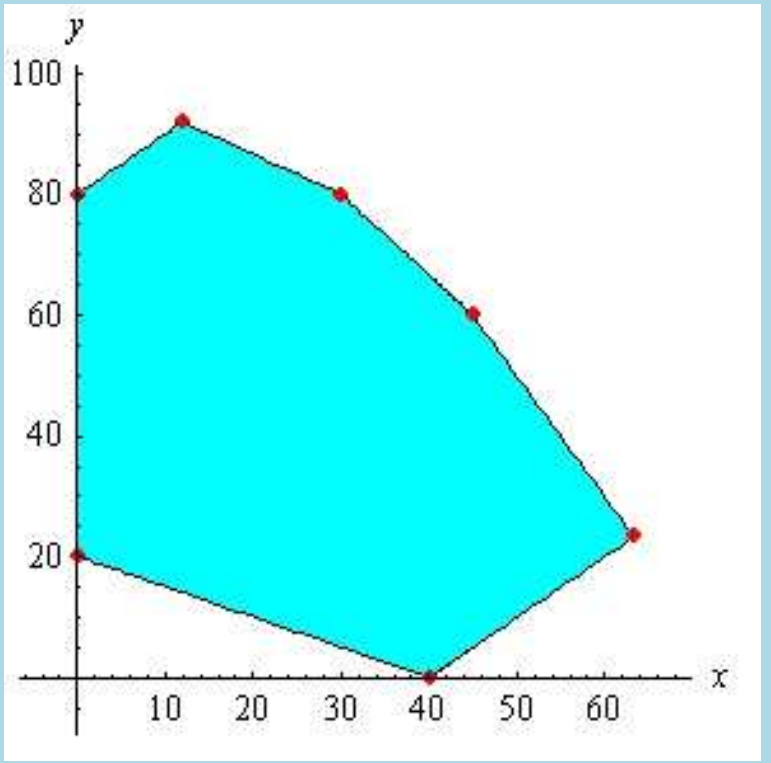
Teorema 4 Neka je skup Ω_P ograničen. Tada postoji $\inf_{x \in \Omega_P} f(x)$, i on se dostiže u ekstremnoj tački x^* skupa Ω_P. Skup $\Omega_P^* = \{x | \, x \in \Omega_P, f(x) = f(x^*)\}$ je konveksan.

Zaključujemo da optimalno rešenje treba tražiti medju najviše $\binom{n}{m}$ bazično dopustivih rešenja. Ova činjenica je fundamentalna i za geometrijski i za simpleks metod.

2.3 Geometrijski metod

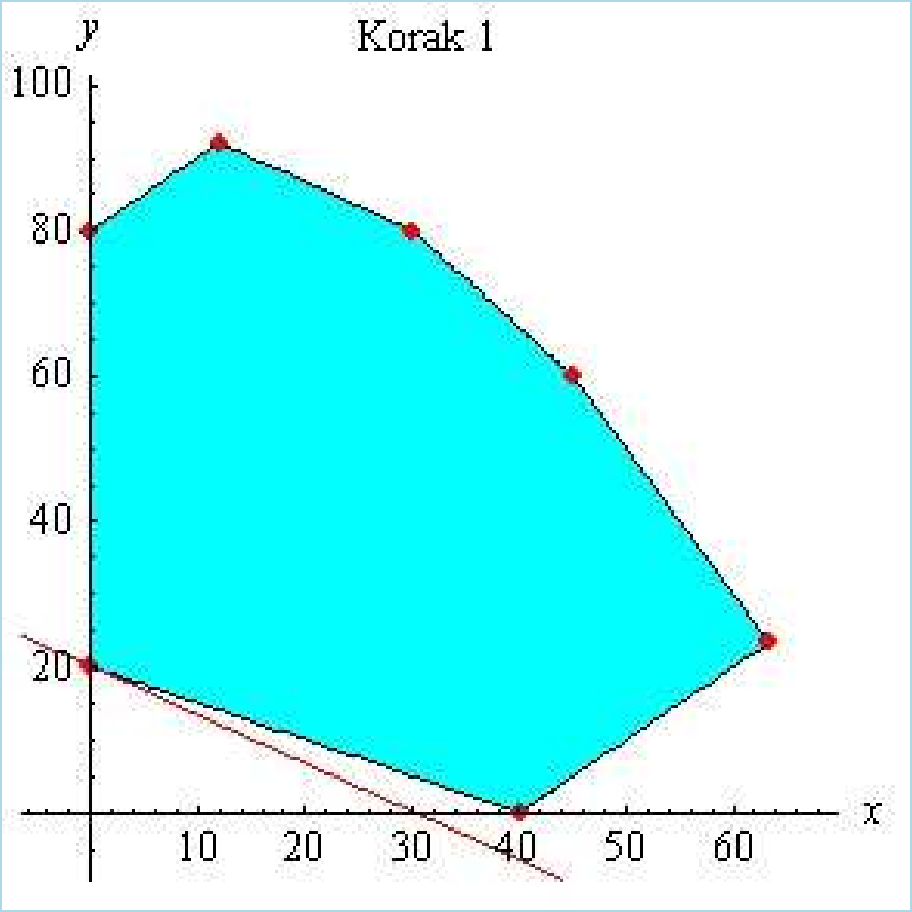
Primer 2

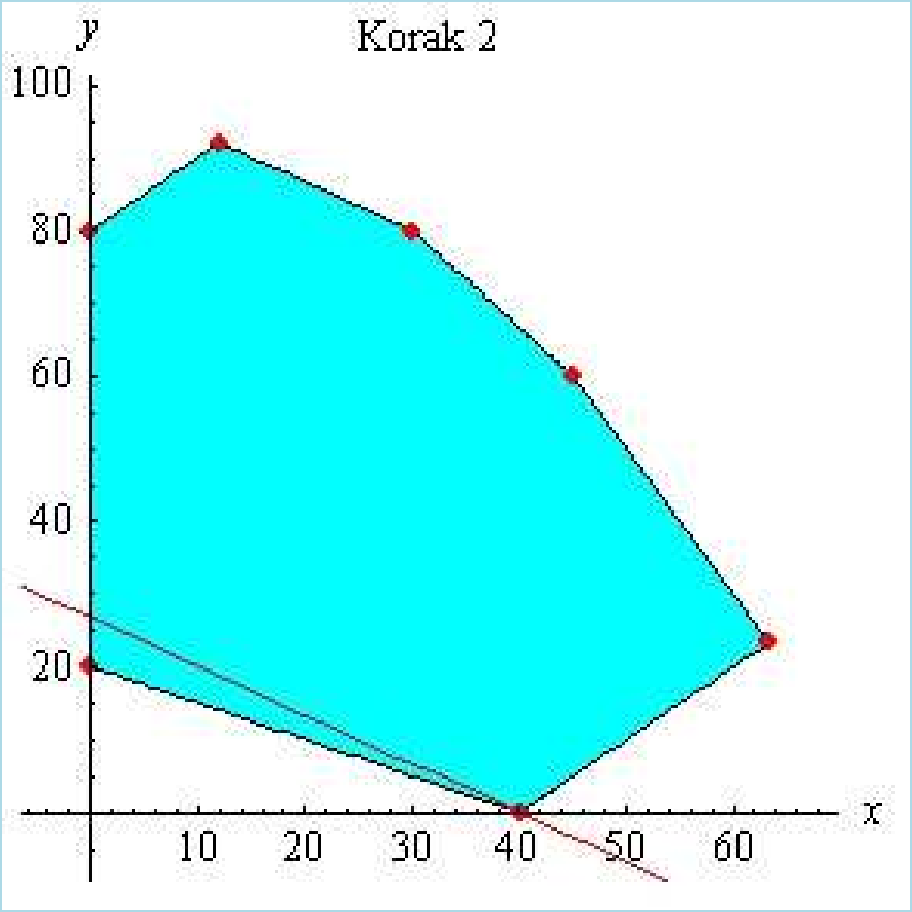
$$
\begin{aligned}
\max \quad & f(x,y) = 8x + 12y \\
\text{p.o.} \quad & 8x + 4y \leq 600 \\
& 2x + 3y \leq 300 \\
& 4x + 3y \leq 360 \\
& 5x + 10y \geq 600 \\
& x - y \geq -80 \\
& x - y \leq 40 \\
& x, y \geq 0
\end{aligned}
$$

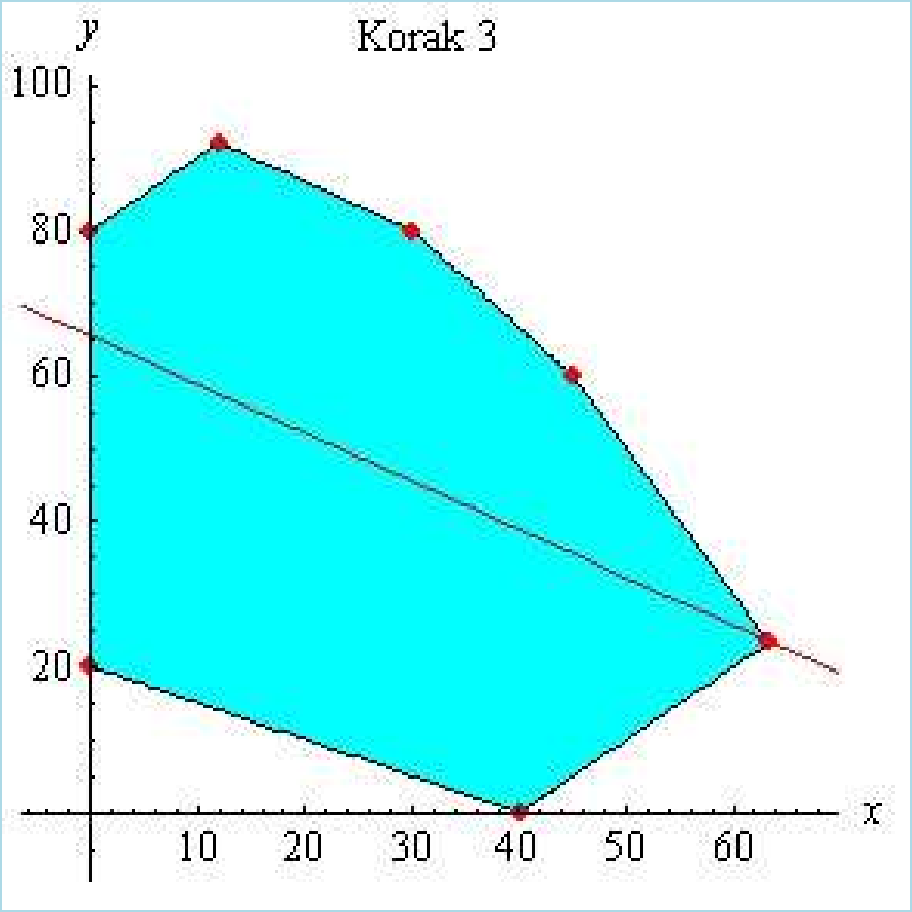


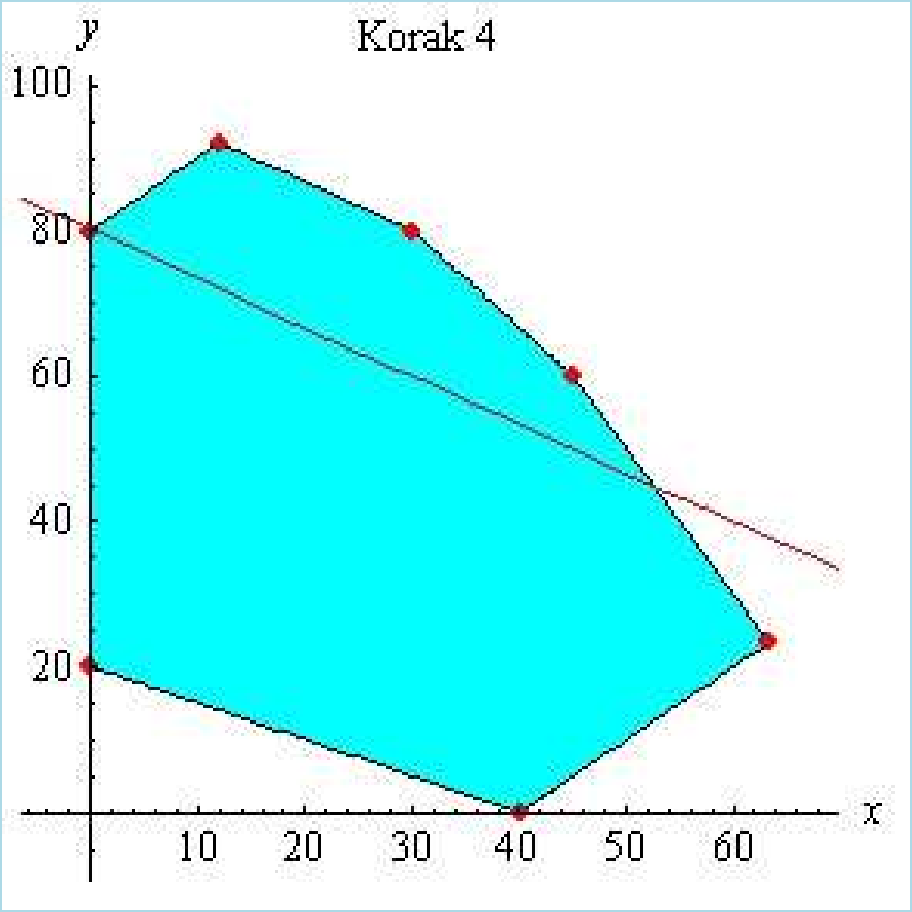
Program GEOM.

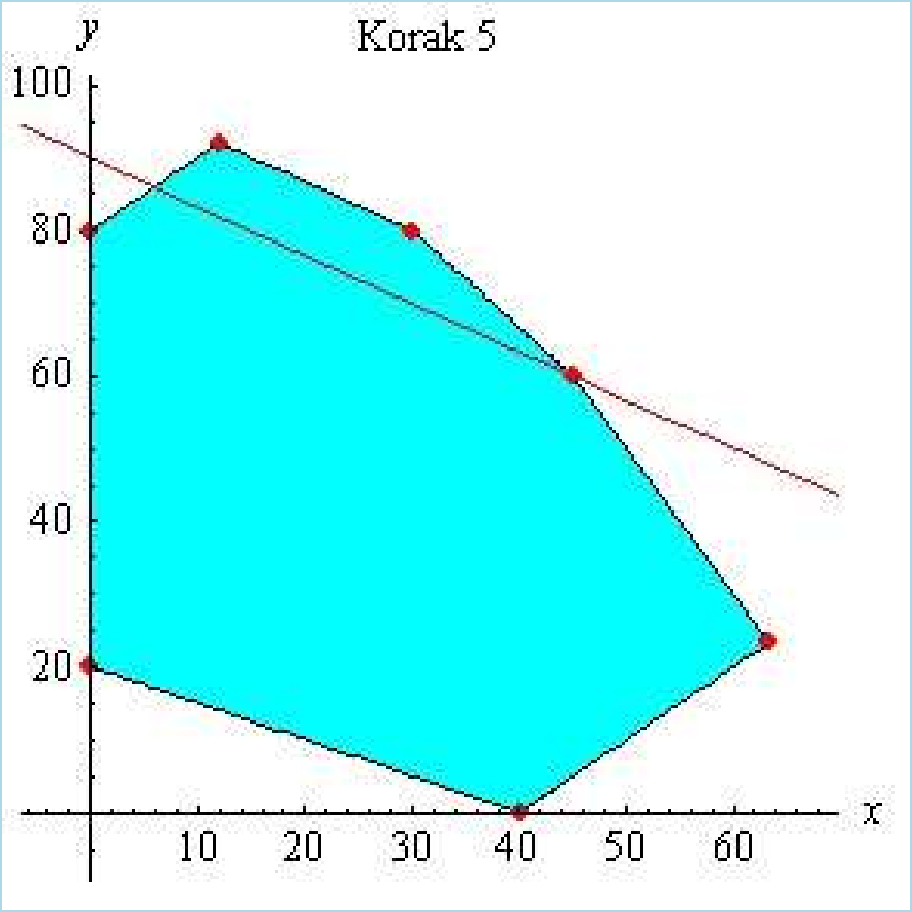
```
Geom[8x+12y, {8x+4y<=600, 2x+3y<=300, 4x+3y<=360, 5x+10y>=600,
x-y>=80, x-y<=40, x>=0, y >= 0}]
```

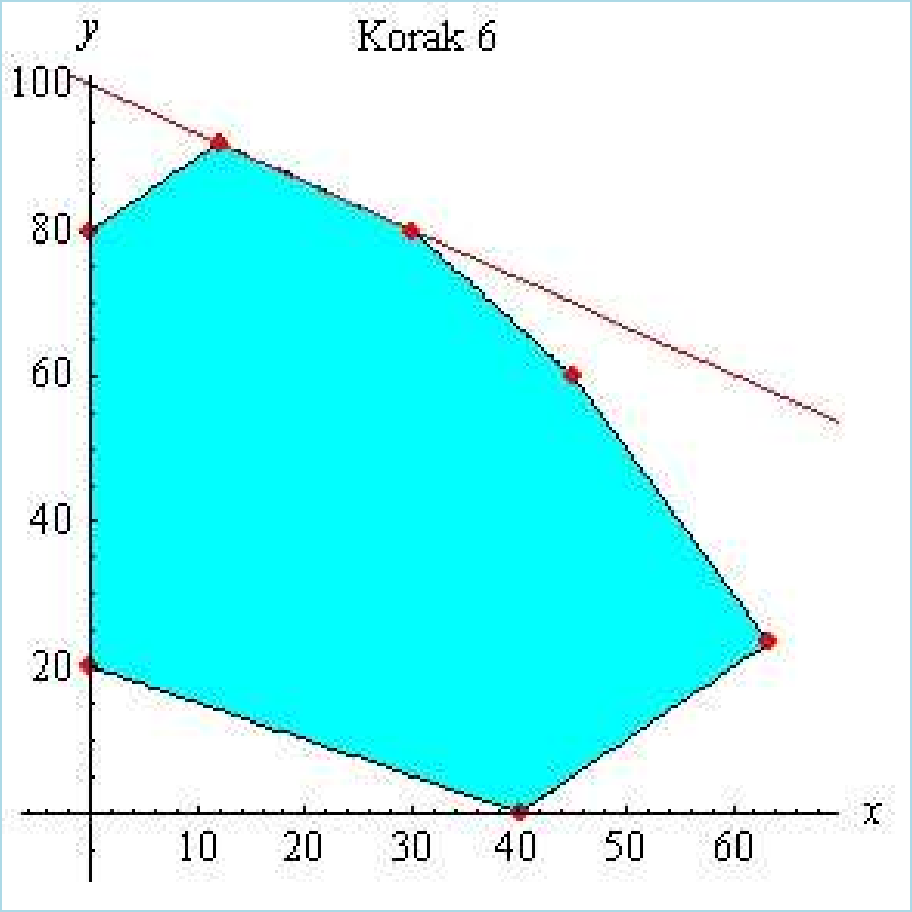

Korak 1
y
100
80
60
40
20
x
10
20
30
40
50
60


Korak 2
y
100
80
60
40
20
x
10
20
30
40
50
60


Korak 3
y
100
80
60
40
20
x
10
20
30
40
50
60


Korak 4
y
100
80
60
40
20
x
10
20
30
40
50
60


Korak 5
y
100
80
60
40
20
x
10
20
30
40
50
60


Korak 6
y
100
80
60
40
20
x
10
20
30
40
50
60

2.4 Simpleks Metod

$$\min \quad f(x) = \sum_{i=1}^{n} c_i x_i + d$$

$$\text{p.o} \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \qquad x_j \geq 0, \quad j = 1, \ldots, n.$$

Simetrični oblik

$$\Rightarrow$$

$$\min \quad f(x) = \sum_{i=1}^{n} c_i x_i + d$$

$$\text{p.o} \quad \sum_{j=1}^{n} a_{ij} x_j - b_i = -x_{n+i}, \qquad x_j \geq 0, \quad j = 1, \ldots, n+m.$$

Kanonski oblik

Svakom kanonskom obliku pridružujemo jedno bazično rešenje:

$$x^* = (0, \ldots, 0, b_1, \ldots, b_m)$$

Ovo rešenje nazivamo bazičnim rešenjem odgovarajućeg kanonskog oblika.

Problem u kanonskom obliku možemo tabelarno prikazati na sledeći način:

$$\begin{array}{ccccccc} x_{N,1} & x_{N,2} & \cdots & x_{N,n} & -1 & & \\ a_{11} & a_{12} & \cdots & a_{1n} & b_1 & = & -x_{B,1} \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 & = & -x_{B,2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m & = & -x_{B,m} \\ c_1 & c_2 & \cdots & c_n & d & = & f \end{array}$$

Ovu tabelu nazivamo Takerovom tabelom.

Lema 1 Neka je $b_i \geq 0$ za svako $i = 1, \ldots, m$ i neka važi $c_j \leq 0$ za svako $j = 1, \ldots, n$. Tada je bazično rešenje x^* optimalno.

Lema 2 Neka je $b_i \geq 0$ i za neko $k \in \{1, \ldots, n\}$ je ispunjeno $c_k > 0$ i $a_{ik} \leq 0$, $i = 1, \ldots, m$. Tada je ciljna funkcija na skupu Ω_P neograničena odozgo.

Pretpostavimo sada da je $c_j > 0$ za neko $j \in \{1, \ldots, n\}$ i $a_{ij} > 0$ za neko $i \in \{1, \ldots, m\}$.

$$-x_{B,i} = a_{i1}x_{N,1} + \ldots + a_{ij}x_{N,j} + \ldots + a_{in}x_{N,n} - b_i$$

Promenljivu $x_{N,j}$ možemo uvećavati do vrednosti:

$$\frac{b_i}{a_{ij}}, \quad b_i > 0, \quad a_{ij} > 0,$$

Daljim uvećavanjem, promenljiva $x_{B,i}$ postaje negativna. Najveća moguća vrednost za $x_{N,j}$ tako da sve bazične promenljive ostanu nenegativne je:

$$\frac{b_p}{a_{pj}} = \min\left\{ \frac{b_i}{a_{ij}} \mid a_{ij} > 0,\ 1 \leq i \leq n \right\}.$$

Za $x_{N,j} = \frac{b_p}{a_{pj}}$ imamo da je $x_{B,p} = 0$. Prema tome $x_{N,j}$ je postala bazična a $x_{B,p}$ nebazična. Time smo dobili novo bazično rešenje, a samim tim i novi kanonski oblik (Takerovu tabelu).

$$
\begin{array}{ccccccccc}
x_{N,1} & x_{N,2} & \cdots & x_{N,j} & \cdots & x_{N,n} & -1 & & \\
a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1n} & b_1 & = & -x_{B,1} \\
a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2n} & b_2 & = & -x_{B,2} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & & \vdots \\
a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{in} & b_i & = & -x_{B,i} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & & \vdots \\
a_{m1} & a_{m2} & \cdots & a_{mj} & \cdots & a_{mn} & b_m & = & -x_{B,m} \\
c_1 & c_2 & \cdots & c_j & \cdots & c_n & d & = & f
\end{array}
$$

Algoritam 1 Replace

$$a^1_{pj} = \frac{1}{a_{pj}};$$

$$a^1_{pl} = \frac{a_{pl}}{a_{pj}}, \quad l \neq j;$$

$$a^1_{qj} = -\frac{a_{qj}}{a_{pj}}, \quad q \neq p;$$

$$a^1_{ql} = a_{ql} - \frac{a_{pl}a_{qj}}{a_{pj}}, \quad q \neq p,\ l \neq j;$$

Algoritam 2 BasicMax (Simpleks metod)

- **Korak 1.** Ako je $c_1, \ldots, c_n \leq 0$, STOP. Bazično dopustivo rešenje koje odgovara Takerovoj tabeli je optimalno.
- **Korak 2.** Izabrati proizvoljno $c_j > 0$.
- **Korak 3.** Ispitati da li je $a_{ij} \leq 0$ za svako $i = 1, \ldots, m$. Ako jeste, STOP. Ciljna funkcija je na dopustivom skupu neograničena odozgo, tj. maksimum je $+\infty$.
- **Korak 4.** Izračunati

$$\min_{1 \leq i \leq m} \left\{ \frac{b_i}{a_{ij}}, \quad a_{ij} > 0 \right\} = \frac{b_p}{a_{pj}}$$

i zameniti promenljive $x_{N,j}$ i $x_{B,p}$ i preći na Korak 1.

Lema 3 Posle svake iteracije algoritma BasicMax vrednost funkcije cilja d^1 u odgovarajućem bazičnom rešenju veća ili jednaka odgovarajućoj vrednosti d za prethodnu tabelu. Takodje je $c_j^1 < 0$.

$$d^1 = d - c_j \frac{b_p}{a_{pj}} \geq d, \qquad c_j^1 = -\frac{c_j}{a_{pj}} < 0$$

Primer 3 Neka je zadat sledeći problem:

$$\begin{aligned} \max \quad & 5x_1 + 4x_2, \\ \text{p.o.} \quad & x_1 + x_2 \leq 80, \\ & 3x_1 + x_2 \leq 180, \\ & x_1 + 3x_2 \leq 180, \\ & x_1 \geq 0,\ x_2 \geq 0 \end{aligned}$$

Posle uvodjenja slack promenljivih dobijamo sledeću Takerovu tabelu:

$$T_0 = \begin{array}{cccc} x_1 & x_2 & -1 & \\ 1 & 1 & 80 & = -x_3 \\ 3 & 1 & 180 & = -x_4 \\ 1 & 3 & 180 & = -x_5 \\ 5 & 4 & 0 & = f \end{array}$$

$$T_0 = \begin{array}{ccccc} x_1 & x_2 & -1 & & \\ 1 & 1 & 80 & = & -x_3 \\ 3 & 1 & 180 & = & -x_4 \\ 1 & 3 & 180 & = & -x_5 \\ 5 & 4 & 0 & = & f \end{array}$$

Korak 2. Uočavamo $c_1 = 5 > 0$. Nastavljamo dalje.

$$
T_0 = \begin{array}{cccc}
x_1 & x_2 & -1 & \\
1 & 1 & 80 & = -x_3 \\
3 & 1 & 180 & = -x_4 \\
1 & 3 & 180 & = -x_5 \\
5 & 4 & 0 & = f
\end{array}
$$

Korak 2. Uočavamo $c_1 = 5 > 0$. Nastavljamo dalje.

Korak 3. Uočavamo $a_{11} = 1 > 0$. Nastavljamo dalje.

$$
\begin{array}{rrrrrr}
 & x_1 & x_2 & -1 & & \\
 & 1 & 1 & 80 & = & -x_3 \\
T_0 = & 3 & 1 & 180 & = & -x_4 \\
 & 1 & 3 & 180 & = & -x_5 \\
 & 5 & 4 & 0 & = & f
\end{array}
$$

Korak 2. Uočavamo $c_1 = 5 > 0$. Nastavljamo dalje.

Korak 3. Uočavamo $a_{11} = 1 > 0$. Nastavljamo dalje.

Korak 4. Sada imamo da je:

$$\frac{b_1}{a_{11}} = 80$$

$$
\begin{array}{rrrrrr}
 & x_1 & x_2 & -1 & & \\
 & 1 & 1 & 80 & = & -x_3 \\
T_0 = & 3 & 1 & 180 & = & -x_4 \\
 & 1 & 3 & 180 & = & -x_5 \\
 & 5 & 4 & 0 & = & f
\end{array}
$$

Korak 2. Uočavamo $c_1 = 5 > 0$. Nastavljamo dalje.

Korak 3. Uočavamo $a_{11} = 1 > 0$. Nastavljamo dalje.

Korak 4. Sada imamo da je:

$$\frac{b_1}{a_{11}} = 80, \qquad \frac{b_2}{a_{22}} = 60$$

$$
\begin{array}{rrrrrr}
 & x_1 & x_2 & -1 & & \\
 & 1 & 1 & 80 & = & -x_3 \\
T_0 = & 3 & 1 & 180 & = & -x_4 \\
 & 1 & 3 & 180 & = & -x_5 \\
 & 5 & 4 & 0 & = & f
\end{array}
$$

Korak 2. Uočavamo $c_1 = 5 > 0$. Nastavljamo dalje.

Korak 3. Uočavamo $a_{11} = 1 > 0$. Nastavljamo dalje.

Korak 4. Sada imamo da je:

$$\frac{b_1}{a_{11}} = 80, \qquad \frac{b_2}{a_{22}} = 60, \qquad \frac{b_3}{a_{33}} = 180$$

$$T_0 = \begin{array}{cccc} x_1 & x_2 & -1 & \\ 1 & 1 & 80 & = -x_3 \\ \boxed{3} & 1 & 180 & = -x_4 \\ 1 & 3 & 180 & = -x_5 \\ 5 & 4 & 0 & = f \end{array}$$

Korak 2. Uočavamo $c_1 = 5 > 0$. Nastavljamo dalje.

Korak 3. Uočavamo $a_{11} = 1 > 0$. Nastavljamo dalje.

Korak 4. Sada imamo da je:

$$\frac{b_1}{a_{11}} = 80, \qquad \frac{b_2}{a_{22}} = 60, \qquad \frac{b_3}{a_{33}} = 180$$

odnosno:

$$\frac{b_2}{a_{21}} = \min\left\{ \frac{b_i}{a_{i1}} \mid a_{i1} > 0 \right\}$$

Biramo $a_{21} = 3$ za pivot element i primenjujemo algoritam Replace.

$$T_0 = \begin{array}{cccccc} x_1 & x_2 & -1 & & \\ 1 & 1 & 80 & = & -x_3 \\ 3 & 1 & 180 & = & -x_4 \\ 1 & 3 & 180 & = & -x_5 \\ 5 & 4 & 0 & = & f \end{array}$$

$$T_1 = \begin{array}{cccccc} x_4 & x_2 & -1 & & \\ -1/3 & 2/3 & 20 & = & -x_3 \\ 1/3 & 1/3 & 60 & = & -x_1 \\ -1/3 & 8/3 & 120 & = & -x_5 \\ -5/3 & 7/3 & -300 & = & f \end{array}$$

$$a_{21}^1 = \frac{1}{a_{21}} = 1,$$

$$a_{22}^1 = \frac{a_{22}}{a_{21}} = \frac{1}{3},$$

$$a_{11}^1 = -\frac{a_{11}}{a_{21}} = -\frac{1}{3},$$

$$a_{31}^1 = -\frac{a_{31}}{a_{21}} = -\frac{1}{3},$$

$$a_{32}^1 = a_{32} - \frac{a_{22}a_{31}}{a_{21}} = \frac{8}{3}.$$

$$b_1^1 = b_1 - \frac{b_2 a_{11}}{a_{21}} = 20,$$

$$b_2^1 = \frac{b_2}{a_{21}} = 60,\ c_1^1 = -\frac{c_1}{a_{21}} = -\frac{5}{3},$$

$$b_3^1 = b_3 - \frac{b_2 a_{31}}{a_{21}} = 120.$$

$$c_2^1 = c_2 - \frac{c_1 a_{22}}{a_{21}} = \frac{7}{3}$$

$$d^1 = d - \frac{b_2 c_1}{a_{21}} = -300.$$

$$
\begin{array}{rrrrrl}
 & x_4 & x_2 & -1 & & \\
 & -1/3 & \frac{2}{3} & 20 & = & -x_3 \\
T_1 = & 1/3 & 1/3 & 60 & = & -x_1 \Rightarrow \\
 & -1/3 & 8/3 & 120 & = & -x_5 \\
 & -5/3 & 7/3 & -300 & = & f
\end{array}
\qquad
\begin{array}{rrrrrl}
 & x_4 & x_3 & -1 & & \\
 & -1/2 & 3/2 & 30 & = & -x \\
T_2 = & 1/2 & -1/2 & 50 & = & -x \\
 & 1 & -4 & 40 & = & -x \\
 & -1/2 & -\frac{7}{2} & -370 & = & f
\end{array}
$$

Bazično dopustivo rešenje koje odgovara tabeli T_2 je $x^2 = (50, 30, 0, 0, 40)$, a ciljna funkcija uzima vrednost $f(x^2) = 370$.

Kako je uslov u Koraku 1 zadovoljen, to je x^2 optimalno rešenje, a vrednost ciljne funkcije je $f(x^2) = 370$.

2.5 Simpleks metod za bazično nedopustive tabele

$$
\begin{array}{ccccccc}
x_{N,1} & x_{N,2} & \cdots & x_{N,n} & -1 & & \\
a_{11} & a_{12} & \cdots & a_{1n} & b_1 & = & -x_{B,1} \\
\vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\
a_{i1} & a_{i2} & \cdots & a_{in} & b_i & = & -x_{B,i} \\
\vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\
a_{m1} & a_{m2} & \cdots & a_{mn} & b_m & = & -x_{B,m} \\
c_1 & c_2 & \cdots & c_n & d & = & f
\end{array}
$$

Lema 4 Ako je $a_{i1}, \ldots, a_{in} \geq 0$ i $b_i < 0$ tada je problem nedopustiv.

Ukoliko je $b_i < 0$ i $a_{ij} < 0$. Ako je u sledećem izrazu $p = i$:

$$\frac{b_p}{a_{pj}} = \min_{l>i} \left(\left\{ \frac{b_i}{a_{ij}} \right\} \bigcup \left\{ \frac{b_l}{a_{lj}} \mid a_{lj} > 0 \right\} \right)$$

posle zamene promenljivih $x_{B,p}$ i $x_{N,j}$, b_i postaje pozitivno.

Algoritam 3 NoBasicMax (Algoritam simpleks metoda za probleme koji nisu bazično dopustivi)

- **Korak 1.** Ako su $b_1, b_2, \ldots, b_m \geq 0$, preći na **Korak 5**.
- **Korak 2.** Izabrati $b_i < 0$, tako da je i maksimalno.
- **Korak 3.** Ako važi $a_{i1}, a_{i2}, \ldots, a_{in} \geq 0$, STOP. Problem linearnog programiranja je nedopustiv. U suprotnom izabrati $a_{ij} < 0$.
- **Korak 4.** Ako je $i = m$, izabrati za ključni element a_{mj}, izvršiti transformaciju i preći na **Korak 1**. U suprotnom, ako je $i < m$, izabrati $a_{ij} < 0$ i izračunati

$$\min_{l>i}\left(\left\{\frac{b_i}{a_{ij}}\right\}\bigcup\left\{\frac{b_l}{a_{lj}}; a_{lj} > 0\right\}\right) = \frac{b_p}{a_{pj}}$$

  Izabrati za ključni element a_{pj}, izvršiti transformaciju, i preći na **Korak 1**.
- **Korak 5.** Primeniti simpleks algoritam za bazično dopustive probleme, algoritam **BasicMax**.

2.6 Revidirani Simpleks metod

- Simpleks metod radi tako što u svakom koraku vrši zamenu promenljivih sa ciljem povećanja (smanjenja) funkcije cilja ili dobijanja dopustivog rešenja.
- U algoritmu Replace vrši se niz deljenja, što dovodi do nagomilavanja numeričke greške.
- Zbog ovoga simpleks metod greši u primerima u kojima je potreban veliki broj iteracija.
- Da bi se to izbeglo, potrebno je da se elementi Takerove tabele računaju pomoću polazne matrice problema. To se postiže revidiranim simpleks metodom.

$$\begin{array}{lll} \max & (c^*)^T x_N - d & A_B x_B + A_N x_N = b \\ p.o. & T x_N - b^* = -x_B & A_B^{-1} A_N x_N - A_B^{-1} b = -x_B \\ & x = (x_B, x_N) \geq 0 & T = A_B^{-1} A_N \quad b^* = A_B^{-1} b \end{array}$$

$$A_B T_{\bullet j} = K_{v_{N,j}} \qquad T_{i\bullet} = (A_B^{-1})_{i\bullet} A_N, \; A_B^T (A_B^{-1})_{i\bullet}^T = (I_m)_{i\bullet}^T$$

Prednosti:

1. Elementi Takerove tabele T računaju se direkno na osnovu matrice A čime se izbegava nagomilavanje greške računskih operacija.
2. Pošto nam u algoritmima simpleks metode nije potrebna cela matrica T već samo pojedini redovi i kolone, revidiranom simpleks metodom mi rekonstruišemo samo te redove i kolone a ne celu matricu T.

Nedostaci:

1. Pošto u svakom koraku moramo ili da tražimo inverznu matricu ili da rešavamo nekoliko (maksimalno 3) sistema linearnih jednačina, iteracija revidiranog simpleksa je znatno sporija od iteracije običnog.
2. Kod običnog simpleksa smo koristili jednostavan algoritam za zamenu promenljivih. Ovde je potrebno implementirati kompleksne algoritme za rešavanje sistema linearnih jednačina ili inverziju matrica.

2.7 Pojam cikliranja i anticiklična pravila

- U lemi pokazali smo da posle svake iteracije algoritma BasicMax vrednost funkcije cilja se ili povećava ili ostaje ista.
- Pri tome, nismo razmatrali mogućnost da se vrednost funkcije cilja ne menja tokom rada algoritma BasicMax.

x_1	x_2	x_3	x_4	-1		
$\frac{1}{4}$	-8	-1	9	0	$=$	$-x_5$
$\frac{1}{2}$	-12	$-\frac{1}{2}$	3	0	$=$	$-x_6$
0	0	1	0	1	$=$	$-x_7$
$\frac{3}{4}$	-20	$\frac{1}{2}$	-6	0	$=$	f

Ukoliko bi nastavili sa primenom algoritma BasicMax, dobijenih 6 tabela bi se stalno ponavljale i nikad ne bi došli do optimalnog rešenja. Primetimo da u koracima 2,3 i 4 izbor elementa ne mora biti jedinstven.

Ova pojava se naziva cikliranje i najčešće se tretira kao retka pojava.

Medjutim još 1977. godine su Kotiah i Steinberg otkrili čitavu klasu "neveštačkih" problema koji cikliraju.

Postoje dva metoda za izbegavanje cikliranja (anticikličnih pravila):

- Leksikografska metoda,
- Blandova pravila.

Po Blandu, treba primenjivati sledeće dve modifikacije koraka 2 i 4:

- Korak 2'. Izabrati $c_j > 0$ tako da je indeks odgovarajuće nebazične promenljive $v_{N,j}$ najmanji.
- Korak 4'. Izračunati

$$\min_{1 \le i \le m} \left\{ \frac{b_i}{a_{ij}}, \quad a_{ij} > 0 \right\} = \frac{b_p}{a_{pj}}$$

U slučaju jednakih vrednosti izabrati ono p takvo da je indeks odgovarajuće bazične promenljive $v_{B,p}$ najmanji. Zameniti nebazičnu promenljivu $x_{N,j}$ i bazičnu promenljivu $x_{B,p}$ i preći na Korak 1.

$$x_{N,j} = x_{v_{N,j}} \qquad x_{B,i} = x_{v_{B,i}}$$

2.8 Složenost simpleks metoda i Minti-Kli poliedri

- U uvodu smo napomenuli da i pored dobrih osobina koje je pokazao u praksi, simpleks algoritam nije polinomijalan.
- To tvrdjenje su prvi dokazali Minti i Kli u radu, još 1970. godine uz pretpostavku da se za pivot kolonu uzima prva kolona kod koje je $c_j < 0$.

Definicija 4 Posmatrajmo sledeći problem linearnog programiranja zadat preko Takerove tabele:

$$\begin{array}{ccccccc} x_1 & x_2 & \cdots & x_n & -1 & & \\ 1 & 0 & \cdots & 0 & t & = & -x_{n+1} \\ 2\epsilon & 1 & \cdots & 0 & t^2 & = & -x_{n+2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ 2\epsilon^{n-1} & 2\epsilon^{n-2} & \cdots & 1 & t^m & = & -x_{n+m} \\ \epsilon^n & \epsilon^{n-1} & \cdots & 1 & 0 & = & f \end{array}$$

Označimo ovaj problem sa $\mathcal{P}_n(\epsilon, t)$ i nazovimo ga uopštenim problemom Minti-Kli-a dimenzije n.

How Good Is the Simplex Algorithm?

VICTOR KLEE*

Department of Mathematics, University of Washington, Seattle, Washington

AND

GEORGE J. MINTY†

Department of Mathematics, Indiana University, Bloomington, Indiana

1. INTRODUCTION

By constructing long "increasing" paths on appropriate convex polytopes, we show that the simplex algorithm for linear programs (at least with its most commonly used pivot rule, Dantzig [1]) is not a "good algorithm" in the sense of Jack Edmonds. That is, the number of pivots or iterations that may be required is not majorized by any polynomial function of the two parameters that specify the size of the program. In particular, $2^d - 1$ iterations may be required in solving a linear program whose feasible region, defined by d linear inequality constraints in d nonnegative variables or by d linear equality constraints in $2d$ nonnegative variables, is projectively equivalent to a d-dimensional cube. Further, for each d there are positive constants α_d and β_d such that

$$\alpha_d n^{[d/2]} < \Xi(d, n) < \beta_d n^{[d/2]} \quad \text{for all} \quad n > d, \tag{1}$$

where $\Xi(d, n)$ is the maximum number of iterations required in solving nondegenerate linear programs whose feasible regions are d-dimensional

x_1	x_2	x_3	-1		
1	0	0	5	$=$	$-x_4$
4	1	0	25	$=$	$-x_5$
8	4	1	125	$=$	$-x_6$
4	2	1	0	$=$	f

x_4	x_2	x_3	-1		
1	0	0	5	$=$	$-x_1$
-4	1	0	5	$=$	$-x_5$
-8	4	1	85	$=$	$-x_6$
-4	2	1	-20	$=$	f

x_4	x_5	x_3	-1		
1	0	0	5	$=$	$-x_1$
-4	1	0	5	$=$	$-x_2$
8	-4	1	65	$=$	$-x_6$
4	-2	1	-30	$=$	f

x_1	x_5	x_3	-1		
1	0	0	5	$=$	$-x_4$
4	1	0	25	$=$	$-x_2$
-8	-4	1	25	$=$	$-x_6$
-4	-2	1	-50	$=$	f

x_1	x_5	x_6	-1		
1	0	0	5	$=$	$-x_4$
4	1	0	25	$=$	$-x_2$
-8	-4	1	25	$=$	$-x_3$
4	2	-1	-75	$=$	f

x_4	x_5	x_6	-1		
1	0	0	5	$=$	$-x_1$
-4	1	0	5	$=$	$-x_2$
8	-4	1	65	$=$	$-x_2$
-4	2	-1	-95	$=$	f

x_4	x_2	x_6	-1		
1	0	0	5	$=$	$-x_1$
-4	1	0	5	$=$	$-x_5$
-8	-4	1	85	$=$	$-x_3$
4	-2	-1	-105	$=$	f

x_1	x_5	x_6	-1		
1	0	0	5	$=$	$-x_4$
4	1	0	25	$=$	$-x_5$
8	4	1	125	$=$	$-x_3$
-4	-2	-1	-125	$=$	f

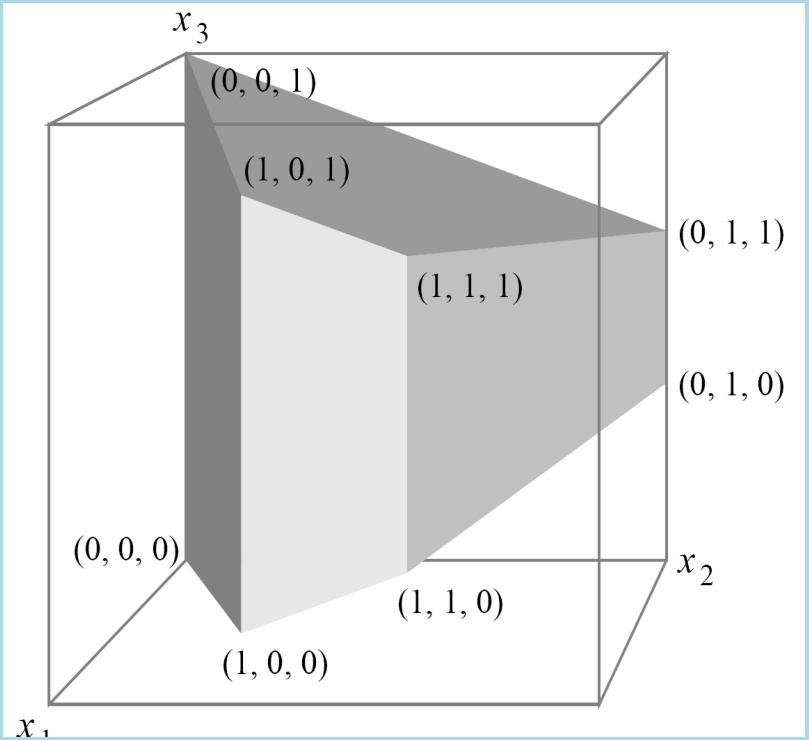
Lema 5 Neka važe uslovi teoreme 5. Ako primenimo algoritam za zamenu promenljivih k puta, pri čemu su pivot elementi $a^i_{p_i p_i} = 1$ gde su brojevi $p_i \in \{1, \ldots, n\}$, $i = 0, \ldots, k-1$ a sa a^l_{ij} je označen element (i, j) Takerove tabele posle l transformacija, dobijamo Takerovu tabelu T_k čiji su elementi jednaki $t^k_{ij} = (-1)^{c^k_{ij}} t^0_{ij}$, za $j < n+1$. Sa c^l_{ij} smo označili broj pivot elemenata p_s za koje važi $j \leq p_s < i$ i $1 \leq s \leq l$.

Lema 6 U svakoj iteraciji algoritma BasicMax, primenjenog na problem $\mathcal{P}_n(\epsilon, t)$ važiće $p = j$, tj. pivot element biće na glavnoj dijagonali Takerove tabele.

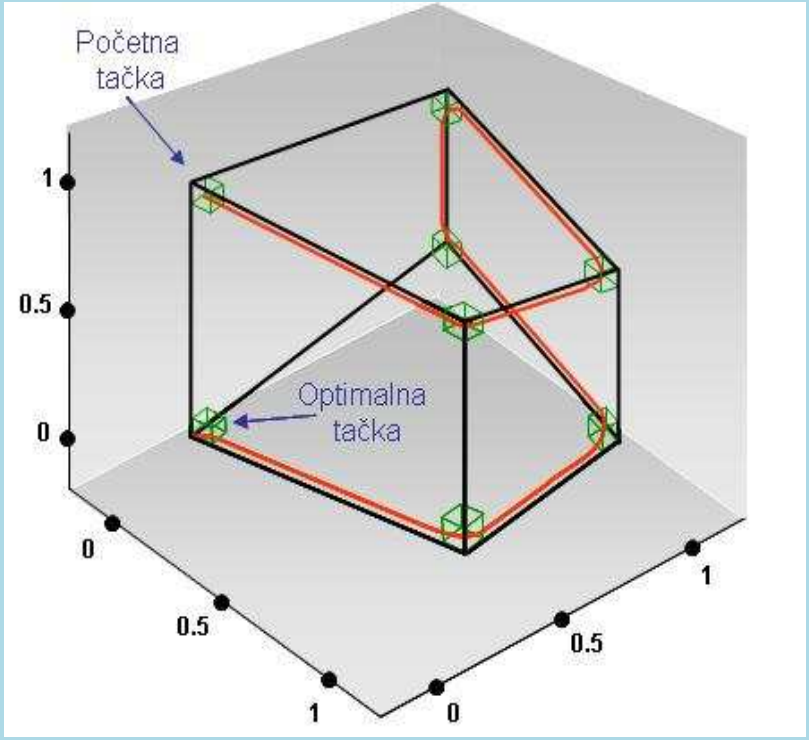
Teorema 5 Neka je $\epsilon, t > 0$ i $\frac{\epsilon}{t} > \frac{1}{2}$. Algoritam $\mathrm{BasicMax}$, primenjen na problem $\mathcal{P}(\epsilon, t)$, prolazi $2^n - 1$ iteracija do optimalnog rešenja $x^* = (0, \ldots, 0, t^n)$.

Zaključak: Složenost simpleks metoda je $\Omega(2^n)$.

Pitanje: Da li postoji pravilo izbora pivot kolone takvo da je složenost simpleks metoda u svim slučajevima polinomijalna?



Minti-Kli poliedar



Putanja simpleks metoda

Teorema 6 (Kli, Kleinschmidt, 1987) Za "skoro svako" determinističko pravilo izbora pivot kolone postoji klasa primera problema linearnog programiranja tako da broj iteracija simpleks metoda zavisi eksponencijalno od dimenzije problema.

Hipoteza 1 (Hirch, 1957) Dijametar (kombinatorni) svakog konveksnog poliedra u D dimenzija koji ima N strana je najviše $N - D$.

- Mini-Kli poliedri i njihove osobine proučavani od strane velikog broja autora.
- Uz višestruko ponavljanje odgovarajućih nejednakosti, pokazano je da odredjena klasa interior-point metoda takodje ima eksponencijalnu složenost u najgorem slučaju.
- Ukoliko se i pivot kolona bira slučajno, očekivani broj iteracija za primer $\mathcal{P}_n(\epsilon, t)$ iznosi:

$$F_n(\bar{x}) = n + 2\sum_{k=1}^{n} \frac{(-1)^{k+1}}{k+2}\binom{n-k}{2} \approx \left(\frac{\pi}{4} - \frac{1}{2}\right) n^2$$

Na osnovu ovoga imamo da očekivana složenost simpleks metoda na primeru $\mathcal{P}_n(\epsilon, t)$ iznosi $\Theta(n^4)$!

3 Modifikacije simpleks metoda i implementacija

- **Razmotrićemo nekoliko originalnih modifikacija simpleks i revidiranog simpleks metoda**
- **Simpleks metod i modifikacije implementirane su u programskim jezicima** Visual Basic 6.0 **i** MATHEMATICA**.**
- **Tako su nastala dva naša originalna programa** MarPlex **i** RevMarPlex**.**

3.1 Poboljšanje algoritma Replace

- U praksi, Takerove tabele mnogih problema linearnog programiranja imaju dosta nula (oko 80%) u sebi.
- Broj operacija koje izvrši algoritam je $(m+1)(n+1)$ i ne zavisi od strukture same tabele.
- Na osnovu algoritma Replace je:

$$a_{ql}^1 = a_{ql} - \frac{a_{pl}a_{qj}}{a_{pj}}, \quad q \neq p,\ l \neq j$$

Vrednost će se menjati vrednost akko su projekcije na ključnu vrstu a_{pl} i a_{qj} različite od 0.

Konstruišimo sada skupove V i K na sledeći način:

$$V = \{l \mid a_{pl} \neq 0,\ l = 1, \ldots, n+1\},$$
$$K = \{q \mid a_{qj} \neq 0,\ q = 1, \ldots, m+1\}.$$

Znači, bilo koji element iz Takerove tabele se menja akko su mu koordinate redom u skupovima V i K.

3.2 Modifikacija algoritma NoBasicMax

- **Korak 4.** Ako je $i = m$, izabrati za ključni element a_{mj}, izvršiti transformaciju i preći na **Korak 1.** U suprotnom, ako je $i < m$, izabrati $a_{ij} < 0$ i izračunati

$$\min_{l>i}\left(\left\{\frac{b_i}{a_{ij}}\right\}\bigcup\left\{\frac{b_l}{a_{lj}}; a_{lj} > 0\right\}\right) = \frac{b_p}{a_{pj}}$$

Izabrati za ključni element a_{pj}, izvršiti transformaciju, i preći na **Korak 1.**

Možemo uočiti dva nedostatka algoritma NoBasicMax:

1. Ako je $p = i$ i ako postoji indeks $t < i = p$ tako da je

$$\frac{b_t}{a_{tj}} < \frac{b_p}{a_{pj}}, \quad b_t > 0,\ a_{tj} > 0$$

u sledećoj iteraciji b_t^1 postaje negativno:

$$b_t^1 = b_t - \frac{b_i}{a_{ij}}a_{tj} < 0.$$

2. Ako je $p > i$, i u sledećoj iteraciji b_i^1 je negativan, ali može da postoji $b_t < 0$, $t < i$ tako da je

$$\min_{k>t}\left(\left\{\frac{b_t}{a_{tj}},\ a_{tj}<0\right\}\cup\left\{\frac{b_k}{a_{kj}}\mid a_{kj}>0,\ b_k>0\right\}\right)=\frac{b_t}{a_{tj}}.$$

U tom slučaju je moguće izabrati a_{tj} za pivot element i dobijamo

$$b_t^1=\frac{b_t}{a_{tj}}\geq 0.$$

Takodje, kako je

$$\frac{b_t}{a_{tj}}\leq\frac{b_k}{a_{kj}},$$

svaki $b_k > 0$ ostaje pogodan za bazično dopustivo rešenje:

$$b_k^1=b_k-\frac{b_t}{a_{tj}}a_{kj}\geq 0.$$

Iz tih razloga predlažemo modifikaciju koraka 4. Glavna ideja je sadržana u sledećoj lemi:

Lema 7 Neka je problem dopustiv i neka je $b_{i_1}, \ldots, b_{i_q} < 0$ i $I = \{i_1, \ldots, i_q\}$. U sledeća dva slučaja:

a) $q = m$,

b) $q < m$ i postoji $r \in I$ i $s \in \{1, \ldots, n\}$ tako da važi:

$$\min_{h \notin I} \left\{ \frac{b_h}{a_{hs}} \mid a_{hs} > 0 \right\} \geq \frac{b_r}{a_{rs}}, \; a_{rs} < 0,$$

moguće je dobiti novo bazično rešenje $x^1 = \{x^1_{B,1}, \ldots, x^1_{B,m}\}$ sa najviše $q - 1$ negativnih koordinata, ako se a_{rs} izabere za pivot element, tj. ako zamenimo nebazičnu promenljivu $x_{N,s}$ bazičnom promenljivom $x_{B,r}$.

Algoritam 4 ModNoBasicMax (Modifikacija algoritma NoBasicMax)

- **Korak 1.** Ako je $b_1, \ldots, b_m \geq 0$ preći na korak 7.
- **Korak 2.** Konstruisati skup

$$B = \{b_{i_1}, \ldots, b_{i_e}\} = \{b_{i_k} \mid b_{i_k} < 0,\ k = 1, \ldots, e\}.$$

- **Korak 3.** Izabrati proizvoljno $b_{i_s} < 0$.
- **Korak 4.** Ako je $a_{i_s,1}, \ldots, a_{i_s,n} \geq 0$ tada STOP. Problem linearnog programiranja nema rešenja. U suprotnom, konstruisati skup

$$Q = \{a_{i_s,j_p} < 0 \mid k = 1, \ldots, t\},$$

i staviti $p = 1$.

- Korak 5. Naći minimum:

$$\min_{1\leq k\leq m}\left\{\frac{b_k}{a_{k,j_p}} \mid a_{k,j_p}>0,\ b_k>0\right\}=\frac{b_u}{a_{u,j_p}}.$$

Ako je

$$\frac{b_{i_s}}{a_{i_s,j_p}}\leq\frac{b_u}{a_{u,j_p}}$$

zameniti nebazičnu i bazičnu promenljivu x_{N,j_p} i x_{B,i_s} i preći na korak 2. Vrednost b_{i_s} postaje pozitivna.

- Korak 6. Ako je $p\leq t$ staviti $p=p+1$ i preći na korak 5. U suprotnom zameniti promenljive x_{N,j_p} i $x_{B,u}$ i preći na korak 4. Vrednost b_{i_s} je i dalje negativna.

- Korak 7. Primeniti simpleks algoritam za bazično dopustive probleme, algoritam BasicMax.

3.3 Poboljšana verzija modifikacije

U našem radu [5], uočili smo da algoritam ModNoBasicMax možemo još malo poboljšati.

Može da se dogodi da uslov leme 7 nije zadovoljen za $i = i_s$ ali da postoji neko drugo $b_i < 0$ tako da je $a_{ij_p} < 0$ i da je uslov leme zadovoljen.

Ovo razmatranje nas navodi na sledeću strategiju izbora pivot elementa:

- Najpre moramo obezbediti da elementi b_l koji su pozitivni takvi i ostanu. Zato mora biti
$$\frac{b_i}{a_{ij}} \leq \min\left\{\frac{b_k}{a_{kj}} \mid b_k > 0, a_{kj} > 0\right\}$$
- Da bi negativan b_l postao pozitivan mora da je ili $a_{lj} > 0$ ili:
$$\frac{b_l}{a_{lj}} \leq \frac{b_i}{a_{ij}}$$

Na osnovu ovoga konstruišemo sledeći algoritam [5].

Algoritam 5 **AdvModNoBasicMax** (Poboljšana verzija algoritma **ModNoBasicMax**)

- **Korak 1.** Ako je $b_1, \ldots, b_m \geq 0$ preći na **korak 5**.
- **Korak 2.** Konstruisati skup

$$B = \{b_{i_1}, \ldots, b_{i_e}\} = \{b_{i_k} \mid b_{i_k} < 0,\ k = 1, \ldots, e\}.$$

- **Korak 3.** Neka je $s = 1$.

  Korak 3.1. Ako je $a_{i_s,1}, \ldots, a_{i_s,n} \geq 0$ onda STOP. Problem je nedopustiv. U suprotnom konstruisati skup

$$Q = \{a_{i_s,j_p} < 0 \mid p = 1, \ldots, t\},$$

  i postaviti $p = 1$.

  Korak 3.2. Naći minimume

$$M(j_p) = \min\left\{\frac{b_k}{a_{k,j_p}} \mid b_k > 0, a_{k,j_p} > 0\right\}.$$

$$p' = \operatorname{argmin}\left\{\frac{b_k}{a_{k,j_p}} \mid b_k < 0, a_{k,j_p} < 0\right\},$$

Ako je $\frac{b_k}{a_{k,j_p}} \leq M(j_p)$ odabrati a_{p',j_p} za pivot element, izvršiti zamenu promenljivih x_{B,j_p} i $x_{N,p'}$ i preći na korak 1. (U sledećoj iteraciji b_k postaje pozitivno).

Korak 3.3. Ako je $p < t$ onda staviti $p = p + 1$ i preći na korak 3.2.

Korak 3.4. Ako je $s < e$ onda staviti $s = s + 1$ i preći na korak 3.1.

- Korak 4. (Uslov leme 7 nije zadovoljen ni za jedno i_s i j_p) Označimo sa $v_{N,j}$ indeks bazične promenljive $x_{N,j}$ (odnosno takvu vrednost da važi $x_{v_{N,j}} = x_{N,j}$). Odredjujemo pivot prema Blandovim pravilima.

  Korak 4.1. Naći $j_0 = \operatorname{argmin}\left\{v_{N,l} \mid a_{i_q,l} < 0\right\}$.

  Korak 4.2. Naći

  $$p'' = \operatorname{argmin}\left\{v_{B,p} \mid \frac{b_p}{a_{p,j_0}} = M(j_0)\right\}.$$

  Korak 4.3. Izvršiti zamenu promenljivih $x_{B,j}$ i $x_{N,p''}$ i preći na korak 3.

- Korak 5. Primeniti simpleks algoritam za bazično dopustive probleme, algoritam BasicMax.

3.4 Modifikacija revidiranog simpleks metoda

Ovde moramo uzeti u obzir da nemamo celu Takerovu tabelu na raspolaganju već samo jedan njen deo.

Lema 8 Neka je problem dopustiv i neka je x bazično nedopustivo rešenje sa q negativnih koordinata. Tada postoji $T_{ij} < 0$. Takodje, u sledeća dva slučaja:

a) $q = m$,

b) $q < m$ and

$$Neg = \left\{ \frac{b_k^*}{T_{kj}} \mid b_k^* < 0, T_{kj} < 0, k = 1, \ldots, m \right\},$$

$$Pos = \left\{ \frac{b_k^*}{T_{kj}} \mid b_k^* > 0, T_{kj} > 0, k = 1, \ldots, m \right\},$$

$$\min\left(Neg \cup Pos\right) = \frac{b_r^*}{T_{rj}} \in Neg$$

moguće je naći novo bazično rešenje sa najviše $q - 1$ negativnih koordinata, ako odaberemo T_{rj} za pivot elemenat.

Algoritam 6 **ModRevNoBasicMax** **(Modifikacija revidiranog simpleks metoda)**

- **Korak 1.** Neka su A_B i A_N bazična i nebazična matrica. Rekonstruisati vektor $b^* = A_B^{-1} b$.
- **Korak 2.** Konstruisati skup

$$B = \{b^*_{i_1}, \ldots, b^*_{i_q}\} = \{b^*_{i_k} \mid b^*_{i_k} < 0,\ k = 1, \ldots, q\}.$$

- **Korak 3.** Izabrati proizvoljno $b^*_{i_s} < 0$.
- **Korak 4.** Rekonstruisati i_s-tu vrstu $T_{i_s \bullet}$. Ako je $T_{i_s \bullet} \geq 0$ onda STOP. Problem je nedopustiv. U suprotnom, odabrati $T_{i_s, j} \leq 0$.
- **Korak 5.** Rekonstruisati j-tu kolonu $T_{\bullet j}$. Izračunati

$$\min_{1 \leq i \leq n} \left\{ \frac{b^*_i}{T_{ij}} \mid b^*_i T_{ij} > 0, i = 1, \ldots, m \right\} = \frac{b^*_p}{T_{pj}},$$

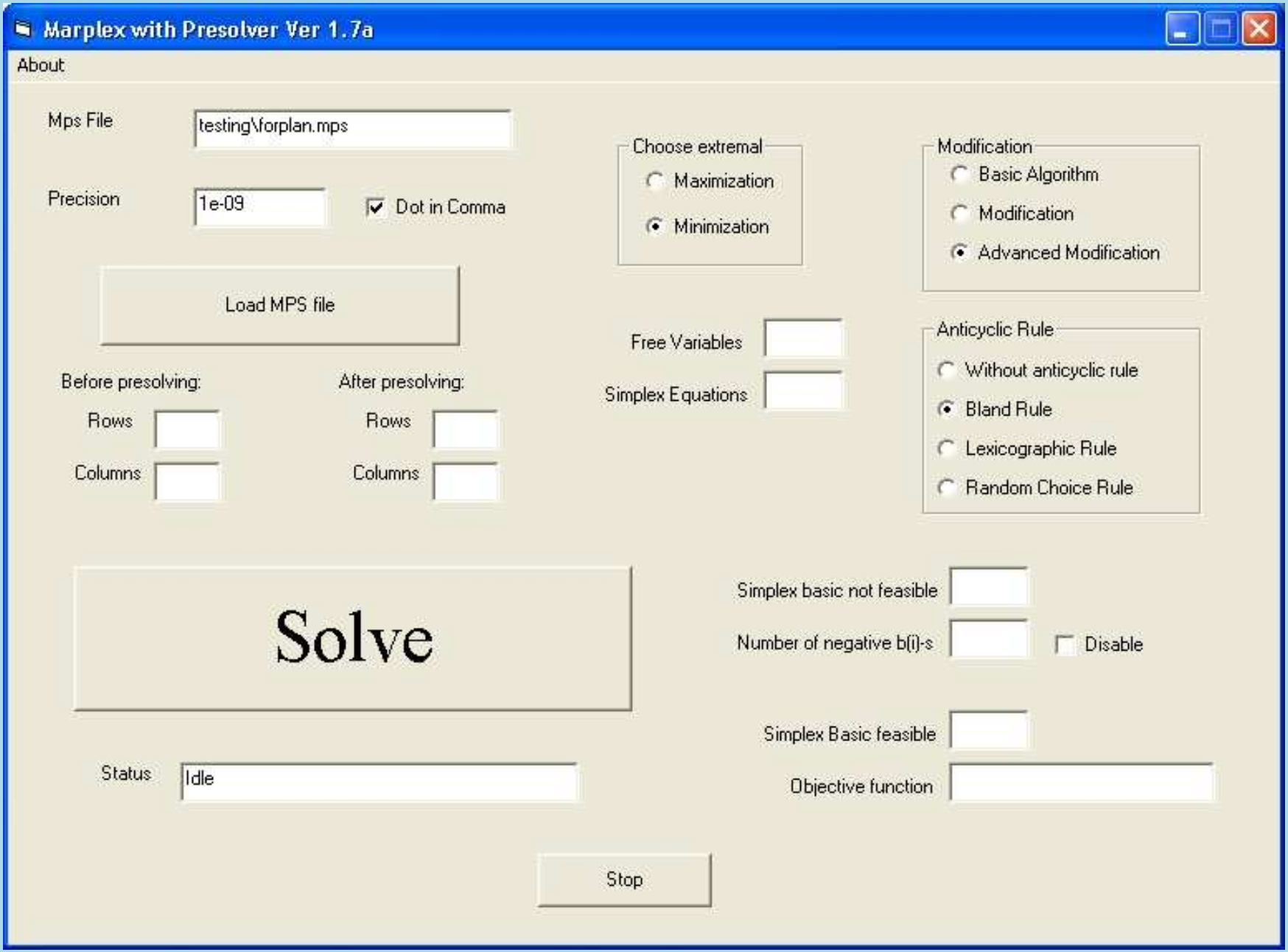
zameniti promenljive $x_{N,j}$ i $x_{B,p}$ i preći na **korak 2**.

3.5 Implementacija simpleks metoda i program MarPlex

- **Simpleks algoritam kao i modifikacije implementirani su u programskom jeziku** Visual Basic 6.0
- **Tako je nastao program** MarPlex **koji vrlo uspešno rešava široku klasu problema linearnog programiranja.**
- MarPlex **je besplatan program i dostupan je na internetu na:**

  `tesla.pmf.ni.ac.yu/people/pecko/myhomepage/Software/MarPlex.zip`

- **Ulazni podaci se zadaju obliku** MPS **fajla. Ovaj tip fajla predstavljaju svetski standard za zadavanje problema linearnog programiranja u vidu simpleks matrica (tablica).**
- **Interface** MarPlex**-a je prvenstveno prilagodjen korisniku i omogućava udoban rad i za razliku od drugih sličnih programa ne opterećuje korisnika brojnim opcijama koje se mahom ne koriste.**
- **Program** MarPlex **smo testirali na referentnim svetskim** *Netlib* **test problemima.**


Marplex with Presolver Ver 1.7a
About
Mps File
testing\forplan.mps
Precision
1e-09
Dot in Comma
Choose extremal
Maximization
Minimization
Modification
Basic Algorithm
Modification
Advanced Modification
Load MPS file
Free Variables
Simplex Equations
Anticyclic Rule
Without anticyclic rule
Bland Rule
Lexicographic Rule
Random Choice Rule
Before presolving:
Rows
Columns
After presolving:
Rows
Columns
Solve
Simplex basic not feasible
Number of negative b(i)-s
Disable
Simplex Basic feasible
Objective function
Status
Idle
Stop

3.6 Implementacija revidiranog simpleks metoda i program RevMarPlex

- **Revidirani simpleks metod kao i modifikaciju implementirali smo u programskom jeziku** MATHEMATICA.
- **Tako je nastao program** RevMarPlex, **varijanta** MarPlex-**a koja koristi revidirani simpleks metod.**
- **Koristili smo** MATHEMATICA-**u zbog integrisanog solvera za sisteme linearnih jednačina (funkcija** `LinearSolve`**).**
- **Pored toga, kod** RevMarPlex-**a koristili smo prednosti** MATHEMATICA-**e po pitanju ulaznih i izlaznih podataka (ovde se funkcija cilja i ograničenja zadaju u simboličkom obliku).**
- **U sledećoj tabeli su prikazani rezultati testiranja programa** RevMarPlex **na** *Netlib* **test primerima.**

Problem	PCx	RevMarPlex	Mod.	Klas. alg.
Adlittle	2.25494963×10^5	225494.963162	32	39
Afiro	-4.64753143×10^2	-464.753142	2	17
Agg	3.59917673×10^7	-35991767.286576	151	151
Agg2	-2.0239251×10^7	-20239252.3559776	75	129
Blend	-3.08121498×10^1	-30.812150	-	-
Sc105	$-5.2202061212 \times 10^1$	-52.202061	-	-
Sc205	-5.22020612×10^1	-52.202061	-	-
Sc50a	$-6.4575077059 \times 10^1$	-64.575077	-	-
Sc50b	-7.000000000×10^1	-70	-	-
Scagr25	-1.47534331×10^7	-14753433.060769	520	> 1500
Scagr7	-2.33138982×10^6	-2331389.824330	55	74
Stocfor1	$-4.1131976219 \times 10^4$	-41131.976219	14	17
LitVera	1.999992×10^{-2}	0	-	-
Kb2	-1.7499×10^3	-1749.9001299062	-	-
Recipe	-2.66616×10^2	-266.6160	13	15
Share1B	-7.65893186×10^2	76589.3185791859	498	> 1500

Oba programa smo testirali na ekstemno loše uslovljenih primera KBAPAH.

Program PCx nije uspeo da reši ove primere.

Problem	RevMarPlex	MarPlex
07-20-02	0	0
15-30-03	7.2345×10^{-9}	2.16968×10^{-5}
15-30-04	2.16968×10^{-5}	1.3984×10^{-3}
15-30-06	4.90359×10^{-6}	1.671×10^{-3}
15-30-07	3.2807×10^{-4}	1.749×10^{-3}
15-60-07	0	-6.29305×10^{-7}
15-60-09	-6.29305×10^{-7}	-1.8353×10^{-6}
20-40-05	0	1.63948×10^{-6}
30-60-05	0	1.64567×10^{-11}
30-60-08	0	5.22527×10^{-5}
LitVera	0	0

Optimalna vrednost ciljne funkcije u svim primerima je jednaka 0.

4 Višekriterijumska optimizacija

- **Proces istovremene optimizacije više ciljnih funkcija naziva se višekriterijumska optimizacija (VKO)**
- **Proučićemo nekoliko metoda za rešavanje problema VKO.**
- **Zajedničko svim metodama je da se svode na nelinearno programiranje.**
- **Proučićemo i neke specijalne slučajeve.**
- **Za svaki metod je data naša originalna implementacija u programskom jeziku** MATHEMATICA**.**

4.1 Definicija i osnovna svojstva

Definicija 5 **Problem višekriterijumske optimizacije** **može da se definiše na sledeći način:**

$$\max [f_1(x), \ldots, f_l(x)]$$
$$x \in \Omega_P$$

gde su $f_i : R^n \Rightarrow R$ **funkcije cilja a** $\Omega_P \subseteq \mathbb{R}^n$ **skup dopustivih rešenja za problem višekriterijumske optimizacije definisan skupovnom jednakošću:**

$$\Omega_P = \{x \in \mathbb{R}^n \mid g_i(x) \leq 0,\ i = 1, ..., m;\ x_j \geq 0,\ j = 1, ..., n\}.$$

Funkcije $g_i(x)$ **se nazivaju funkcijama ograničenja.**

Kao i kod linearnog programiranja, i ovde elemente skupa Ω_P **nazivamo dopustivim rešenjima.**

Teorema 7 **Ako su sve funkcije** $g_i(x)$ **konveksne, tada je i skup** Ω_P **konveksan.**

Definicija 6 **Marginalna rešenja** problema VKO su optimumi svake funkcije cilja pojedinačno:

$$\begin{aligned} \max \quad & f_k(x) = f(x^{(k)}), \\ \text{p.o.} \quad & x \in \Omega_P, \end{aligned}$$

za svako $k = 1, \ldots, l$.

Definicija 7 **Idealne vrednosti** za funkcije cilja f_k definišu se kao vrednosti funkcija cilja za marginalna rešenja: $f_k^* = f_k(x^{(k)}),\ k = 1, ..., l$. Idealne vrednosti funkcija cilja odredjuju **idealnu tačku** u kriterijumskom prostoru, $f^* = (f_1^*, f_2^*, ..., f_l^*)$. **Savršeno rešenje** je takvo rešenje koje istovremeno maksimizira sve funkcije cilja, $f_k(x^*) = f_k^*$

U teoriji višekriterijumske optimizacije, fundamentalan je koncept Pareto optimalnosti.

Definicija 8 Dopustivo rešenje x^* predstavlja Pareto optimum problema VKO ako ne postoji neko drugo dopustivo rešenje x takvo da važi:

$$f_k(x) \geq f_k(x^*), \forall k = 1, ..., l$$

pri čemu bar jedna od nejednakosti prelazi u strogu nejednakost:

$$f_{k_i}(x) > f_{k_i}(x^*), \forall i = 1, ..., m,\ m < l.$$

Definicija 9 Dopustivo rešenje x^* je slabi Pareto optimum ako ne postoji neko drugo dopustivo rešenje x tako da važi:

$$f_k(x) > f_k(x^*), \forall k = 1, ..., l.$$

4.2 Metoda težinskih koeficijenata

Formira se nov jednokriterijumski problem:

$$\begin{aligned} \max \quad & f^M(x) = \sum_{k=1}^{l} w_k f_k^o(x) \\ \text{p.o.} \quad & x \in \Omega_P, \end{aligned}$$

gde je $w_k \geq 0$ i f_k^o je normalizovana k-ta funkcija cilja $f_k(x), k = 1, \ldots, l$.

Lema 9 **Rešenje problema višekriterijumske optimizacije dobijeno primenom metode težinsk koeficijenata je pareto optimalno ukoliko važe uslovi $S_k > 0$ i $w_k > 0$ za svako $k \in \{1, \ldots, l\}$.**

4.3 Leksikografska metoda

U leksikografskoj metodi zadajemo prioritete kriterijumskim funkcijama u vidu strogo definisanog redosleda značajnosti funkcija.

Rešenje problema višekriterijumske optimizacije dobijamo tako što redom rešavamo sledećih l problema jednokriterijumske optimizacije.

$$\begin{aligned} (\max) \quad & f_k(x) \\ \text{p.o.} \quad & f_i(x) = f_i(x^{(i)}),\ i = 1, \ldots,\ k-1,\ k \geq 2, \\ & x \in \Omega_P, \end{aligned}$$

za $k = 1,\ \ldots\ ,\ l$. Pritom su $x^{(i)},\ i = 1, \ldots,\ k-1$, respektivno, rešenja problema na nivoima $i = 1, \ldots,\ k-1,\ k \geq 2$. Rešenje $x^{(l)}$ uzima se kao konačno rešenje problema VKO. Kod leksikografske metode posle svakog koraka smanjuje se dimenzija oblasti dopustivih rešenja.

Lema 10 Rešenje $x^{(l)}$ koje daje leksikografska metoda je Pareto optimalno.

4.4 Relaksirana leksikografska metoda

Ova metoda je modifikacija leksikografske metode.

Parametar $\alpha_k \geq 0,\ k = 1, ..., l-1$,pokazuje dozvoljeno odstupanje od optimalne vrednosti.

$$\begin{aligned} (\max) \quad & f_k(x) \\ \text{p.o.} \quad & f_i(x) \geq f(x^{(i)}) - \alpha_i,\ i = 1, \ldots, k-1,\ k \geq 2, \\ & x \in \Omega_P, x \geq 0, \end{aligned}$$

Lema 11 Rešenje dobijeno relaksiranom leksikografskom metodom je u opštem slučaju slabo Pareto optimalno.

4.5 Metoda ε ograničenja

Izdvaja se kriterijum $f_q(x)$ koji ima najviši prioritet i koji treba maksimizirati,

Ostale funkcije cilja ne smeju imati vrednosti manje od unapred zadatih parametara ε_k,$k = 1, ..., l, k \neq q$.

$$\begin{aligned} (\max) \quad & f_q(x) \\ \text{p.o.} \quad & g_i(x) \leq 0, \quad i = 1, ..., m \\ & f_k(x) \geq \varepsilon_k, \quad k = 1, ..., l, \ k \neq q \\ & x_j \geq 0, \quad j = 1, ..., n \end{aligned}$$

čije se rešenje usvaja kao rešenje polaznog zadatka VKO. Ako postavljeni zadatak nema dopustivo rešenje, potrebno je smanjiti vrednosti ε_k.

Lema 12 Rešenje x^* dobijeno metodom ε ograničenja je slabo Pareto optimalno.

4.6 Metode rastojanja

Kriterijumski prostor $S = \{(f_1(x), \ldots, f_l(x) \mid x \in \Omega_P\}$.

Osnovna ideja je da se u kriterijumskom prostoru traži tačka koja je najbliža nekoj unapred odredjenoj tački f^z (najčešće se za tu tačku uzima idealna tačka f^*).

$$\begin{aligned} (\min) \quad & d(f^z, f(x)) \\ \text{p.o.} \quad & x \in \Omega_P, \qquad i = 1, \ldots, m \end{aligned}$$

Možemo, na primer, koristiti sledeće metrike:

$$d_{l_1}(f^z, f(x)) = \sum_{k=1}^{l} w_k |f_k^z - f_k(x)|, \qquad \text{za pravougaonu metriku,}$$

$$d_{l_2}(f^z, f(x)) = \sqrt{\sum_{k=1}^{l} w_k (f_k^z - f_k(x))^2}, \qquad \text{za Euklidovu metriku,}$$

$$d_{l_\infty}(f^z, f(x)) = \max_{1 \le k \le l} w_k |f_k^z - f_k(x)|, \qquad \text{za Čebiševljevu metriku.}$$

Lema 13 Rešenje dobijeno primenom metode rastojanja, korišćenjem pravougaone metrike, je Pareto optimalno, ukoliko važi $w_k > 0$, za svako $k = 1, \ldots, l$, i ukoliko je $f^z = (f_1^z, ..., f_l^z)$ idealna vrednost funkcije f.

4.7 Linearna višekriterijumska optimizacija

Pretpostavimo da su sve funkcije $g_i(x)$ i $f_i(x)$ linearne. Neka je:

$$g_i(x) = \sum_{j=1}^{n} a_{ij}x_j - b_i, \qquad f_i(x) = \sum_{j=1}^{n} c_{ij}x_j + d_i$$

Ako označimo matrice $A = [a_{ij}]$, $C^T = [c_{ij}]$ formata $m \times n$ i $l \times n$ dobijamo problem linearne višekriterijumske optimizacije:

$$\max C^T x + d$$
$$p.o.\ Ax \leq b \quad x \geq 0$$

Simetrični oblik

$$\max C^T x + d$$
$$p.o.\ Ax = b \quad x \geq 0$$

Standardni oblik

Primetimo da ako je polazni problem višekriterijumske optimizacije linearan, da su onda linearni i odgovarajući jednokriterijumski problemi kod svih prethodno razmatranih metoda, osim metode rastojanja.

Ova činjenica je vrlo zgodna za primenu simpleks metoda za rešavanje odgovarajućih jednokriterijumskih problema.

Neka je T^k optimalna Takerova tabela k-tog problema i neka je dodatni uslov:

$$r_1 x_1 + \ldots + r_n x_n - e \leqq 0 \iff (r_B^k)^T x_B^k + (r_N^k)^T x_N^k - e \leqq 0$$

Sada iz $x_B + T^k x_N = b^k$ dobijamo:

$$\left((r_N^k)^T - (r_B^k)^T T^k\right) x_N^k + (r_B^k)^T b^k - e \leqq 0$$

Odnosno:

$$\tilde{T}^{k+1} = \begin{bmatrix} T^k \\ (r_N^k)^T - (r_B^k)^T T^k \end{bmatrix} \quad \tilde{b}^{k+1} = \begin{bmatrix} b^k \\ -(r_B^k)^T b^k + e \end{bmatrix}$$

Početno rešenje je u ovom slučaju, najčešće vrlo "blizu" optimalnom.

Razmotrimo sada metodu rastojanja u slučaju linearnog problema višekriterijumske optimizacije.

Neka je odabrana metrika euklidska i neka su sve težine $w_k = 1$. U ovom slučaju rešavamo problem kvadratnog programiranja

$$\begin{aligned} &\min \sqrt{\sum_{i=1}^{l} \left(C_{i\bullet}x - f_i^z\right)^2} = \|Cx - f^z\| \\ &p.o\ Ax = b \\ &x \geq 0 \end{aligned}$$

Posmatrajmo na šta se svodi ovaj problem ako pretpostavimo da ograničenja ne postoje. U ovom slučaju je potrebno naći:

$$\min_{x \in \mathbb{R}^n} \|Cx - f^z\|$$

Rešenje se dobija kao $x^* = C^\dagger f^z$ gde je $C^\dagger$ Moore-Penrose-ov pseudoinverz matrice C. U našim radovima [2, 3, 7, 4] prikazani su metodi za simboličko izračunavanje Moore-Penrose-ovog i drugih pseudoinverza polinomijalnih matrica.

5 Primene linearnog programiranja i višekriterijumske optimizacije

- Pokazaćemo dve praktične primene prethodno izloženih metoda matematičkog programiranja (linearnog programiranja i višekriterijumske optimizacije) u:

  astronomiji(fizici) i

  telekomunikacijama(elektronici).

- Linearno programiranje i višekriterijumska optimizacija, kao i ostale mnogobrojne metode matematičkog programiranja imaju primene u skoro svim oblastima nauke, tehnike i uopšte u svakodnevnom životu.

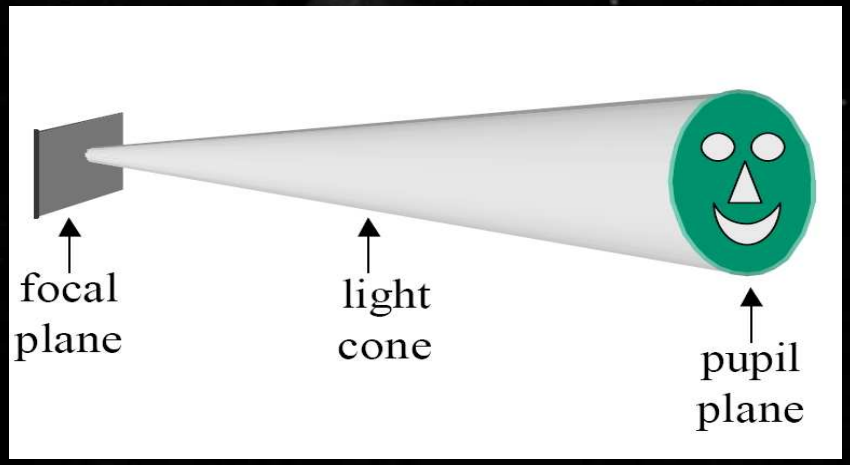
- Sledeća dva primera lepo ilustruju tu činjenicu.

5.1 Izračunavanje optimalne maske teleskopa

Razmotrićemo primenu linearnog programiranja na problem projektovanja teleskopa pomoću kog ćemo moći da utvrdimo da li oko neke zvezde kruže planete.

Svetlost sa udaljene zvezde pada na ravan objektiva teleskopa (eng. pupil plane), prelama se kroz sočivo objektiva i pada na žižnu ravan (eng. focal plane).

Svi svetlosni zraci koji pod istim uglom padnu na objektiv, posle prelamanja padaju u istu tačku na žižnoj ravni.

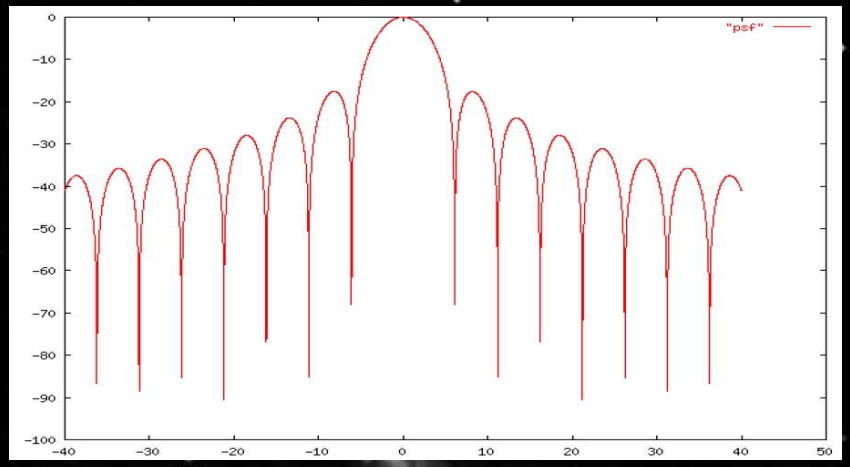


Principijelna šema teleskopa

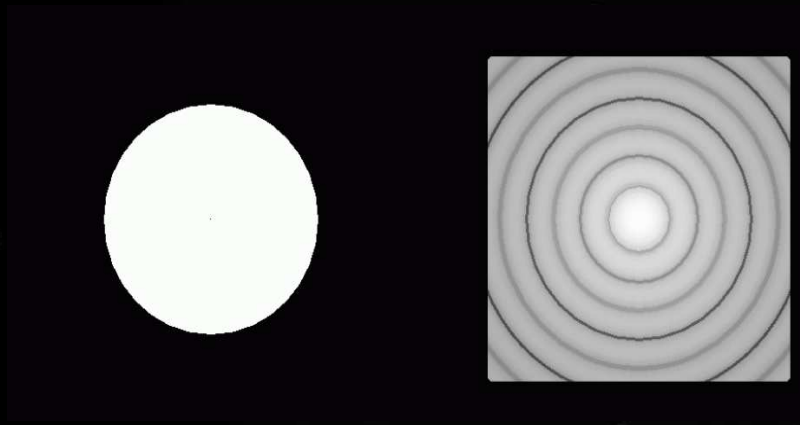
Profil objektiva i difrakciona slika

Medjutim, usled talasne prirode svetlosti, dolazi do difrakcije na otvoru objektiva i umesto jedne svetle tačke, vidimo mrlju konačne veličine koju okružuju prstenovi.

Ovi prstenovi, iako su tamniji od centralne mrlje, ipak su intenzivniji od svetlosti bilo koje planete koja kruži oko zvezde.



Intenzitet svetlosti na x koordinatnoj osi.



Profil objektiva i difrakciona slika, pri čemu je nula na -110dB.

Ideja je da se ovi prstenovi eliminišu pogodnim izborom oblika maske objektiva $A(x)$.

Pretpostavimo da je profil maske iskazan funkcijom $A(x)$, tj da je otvoreni deo objektiva:

$$\left\{(x,y) \mid -\frac{1}{2} \leq x \leq \frac{1}{2},\ -A(x) \leq y \leq A(x)\right\}$$

Naravno, prethodno je izvršena odgovarajuća normalizacija dužina.

Ako usvojimo Fraunhoferovu aproksimaciju, imamo sledeći izraz za jačinu električnog polja u tački (ξ,ζ) u žižnoj ravni:

$$E(\xi,\zeta) = \int_{-1/2}^{1/2}\int_{-A(x)}^{A(x)} e^{i(x\xi+y\zeta)}dydx = 4\int_{0}^{1/2}\cos(x\xi)\frac{\sin(A(x)\zeta)}{\zeta}dx$$

Uslov koji ćemo sada nametnuti je da u prethodno zadatoj oblasti $\mathcal{O}$ važi sledeći uslov:

$$-\alpha E(0,0) \leq E(\xi,\zeta) \leq \alpha E(0,0), \quad (\xi,\zeta) \in \mathcal{O}$$

Pretpostavimo da je sada skup $\mathcal{O}$ podskup x ose. Tada je uslov:

$$\int_0^{1/2} A(x)\left[\cos(x\xi) - \alpha\right] dx \leq 0$$

$$\int_0^{1/2} A(x)\left[\cos(x\xi) + \alpha\right] dx \leq 0$$

Cilj nam je da vidljiva površina objektiva bude što je moguće veća, odnosno da maksimizujemo

$$\int_{-1/2}^{1/2} 2A(x)dx = 4\int_0^{1/2} A(x)dx$$

Znači, posmatramo sledeći optimizacioni problem:

$$\max 4 \int_0^{1/2} A(x)dx$$

$$p.o. \int_0^{1/2} A(x)\left[\cos(x\xi) - \alpha\right] dx \leq 0$$

$$\int_0^{1/2} A(x)\left[\cos(x\xi) + \alpha\right] dx \leq 0$$

$$0 \leq A(x) \leq \frac{1}{2}$$

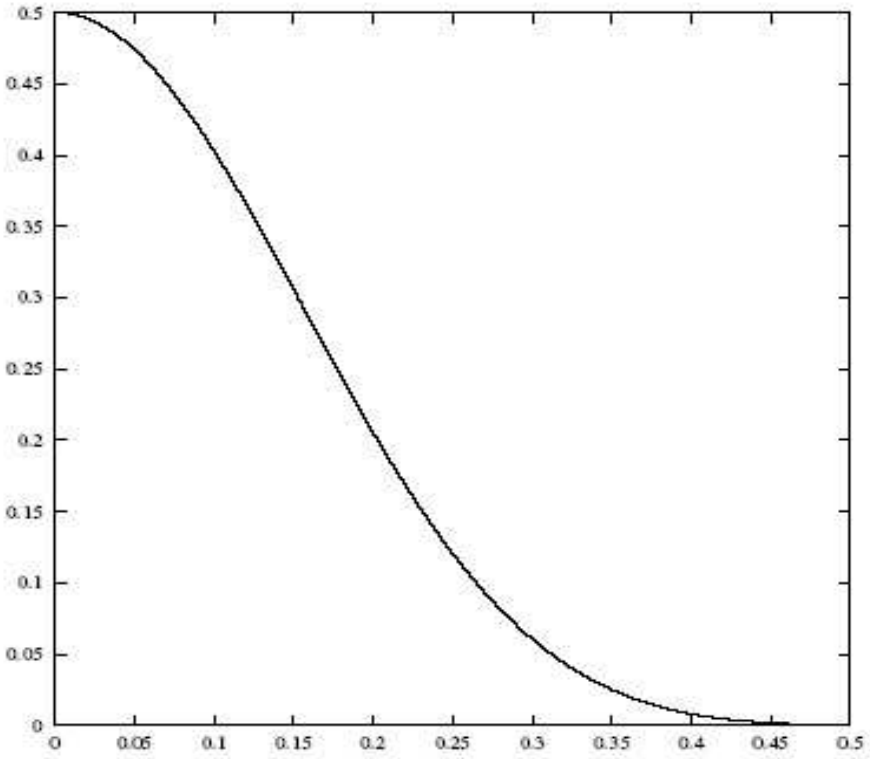
Ako sada izvršimo diskretizaciju integrala i skupa $\mathcal{O}$ dobijamo sledeći problem linearnog programiranja:

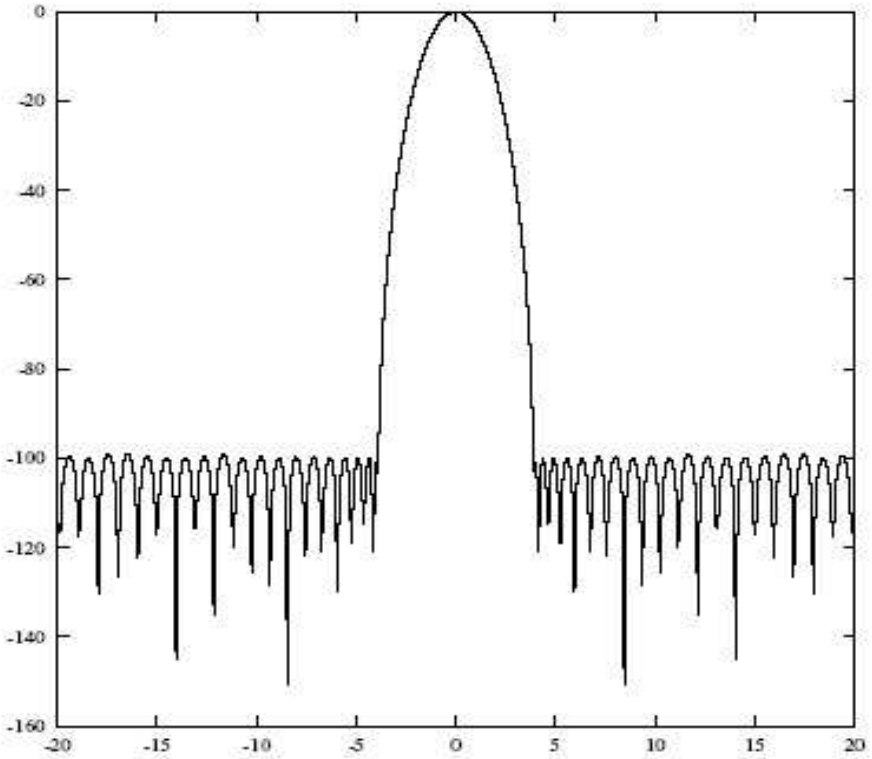
$$\max 4 \sum_{i=1}^{N_x} B_i A(x_i)$$

$$p.o. \sum_{i=1}^{N_x} C_i A(x_i) \left[\cos(x_i \xi_j) - \alpha\right] \leq 0, \quad j = 1, \ldots, N_\xi$$

$$\sum_{i=1}^{N_x} D_i A(x_i) \left[\cos(x_i \xi_j) - \alpha\right] \leq 0 \quad j = 1, \ldots, N_\xi$$

$$0 \leq A(x_i) \leq \frac{1}{2}$$





Profil optimalne maske $A(x)$ i intenzitet svetlosti na x osi ako se koristi optimalna maska

Sa slike vidimo, da je intenzitet svetlosti za $\xi > 4$ manji ili jednak -110dB, tako da je u ovom pojasu moguće zapaziti lik neke planete koja kruži oko zvezde.

5.2 Projektovanje FIR filtara

U ovom odeljku razmotrićemo primenu linearnog programiranja u projektovanju digitalnih FIR filtara.

FIR filtri, kao i uopšte digitalni filtri imaju veliku ulogu u obradi digitalnih signala.

Digitalni signali su za razliku od kontinualnih signala definisani samo u diskretnim vremenskim trenucima i predstavljaju se kao niz realnih vrednosti $x(t)$, $t = 1, 2, \ldots$.

U analizi digitalnih signala najčešće se koriste diskretna Furijeova i z-transformacija.

$$X(f) = \sum_{n=-\infty}^{+\infty} x(n) e^{2\pi f i n} \qquad X(z) = \sum_{n=-\infty}^{+\infty} x(n) z^{-n}$$

Frekvencijska karakteristika idealnog filtra može se prikazati pomoću izraza:

$$|H(f)| = \left| \frac{Y(f)}{X(f)} \right| = \begin{cases} 1, & f \in \mathcal{O} \\ 0, & \text{U suprotnom} \end{cases}$$

Neka je $h(t)$ inverzna diskretna Furijeova transformacija funkcije $H(f)$, data pomoću:

$$h(t) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(f) e^{2\pi i f t} df.$$

Imamo da su signal na izlazu i signal na ulazu povezani sledećom relacijom:

$$y(t) = \sum_{s=-\infty}^{+\infty} h(s) x(t-s)$$

koji predstavlja konvoluciju nizova $x(t)$ i $h(t)$.

Za FIR (Finite Impulse Response) filtre važi da postoji broj n_0 takav da je $h(t) = 0$ ako je $|t| > n_0$. Broj n_0 nazivamo red filtra. U suprotnom, u pitanju je IIR filtar (Infinite Impulse Response).

Pretpostavićemo da je $h(t)$ simetrična funkcija po t.

Sada je $|H(f)|$, takodje, simetrična funkcija i važi $|H(f)| = |A(f)|$ gde je:

$$A(f) = h(0) + 2 \sum_{t=1}^{n_0} h(t) \cos(2\pi f t)$$

Znači, potrebno je minimizovati:

$$\min \int_0^{f_0} |A(f) - 1| df$$

$$\min \int_{f_0}^{1/2} |A(f)| df$$

Ovim smo dobili problem bezuslovne višekriterijumske nelinearne optimizacije. Uvedimo sada pomoćne funkcije $\tau(f)$ i $\eta(f)$. Problem možemo ekvivalentno predstaviti u obliku:

$$\begin{aligned} &\min \int_0^{f_0} \tau(f) df \\ &\min \int_{f_0}^{1/2} \eta(f) df \\ &p.o. \quad -\tau(f) \leq A(f) - 1 \leq \tau(f), \quad f \in \mathcal{O} \\ &\qquad -\eta(f) \leq A(f) \leq \eta(f), \quad f \notin \mathcal{O} \end{aligned}$$

Ako sada izvršimo diskretizaciju po frekvenciji, slično kao u prethodnom odeljku, dobijamo problem linearne višekriterijumske optimizacije.

U praksi se najčešće fiksira jedno odstupanje (npr. na skupu $\mathcal{O}^C$) i traži se minimum drugog. Sada imamo:

$$\begin{aligned} &\min \int_0^{f_0} \tau(f)df \\ &p.o. \quad -\tau(f) \leq A(f) - 1 \leq \tau(f), \quad f \in \mathcal{O} \\ &\qquad -\epsilon \leq A(f) \leq \epsilon, \quad f \notin \mathcal{O}, \quad \tau(f) \geq 0 \end{aligned}$$

a ako uvedemo diskretizaciju po f dobijamo problem linearnog programiranja:

$$\min \sum_{j=1}^{N'_f} B_j \tau(f_j)$$

$$\text{p.o.} \quad -\tau(f_j) \leq h(0) + 2\sum_{t=1}^{n_0} h(t)\cos(2\pi f_j t) - 1 \leq \tau(f_j), \quad j = 1, \ldots, N'_f$$

$$-\epsilon \leq h(0) + 2\sum_{t=1}^{n_0} h(t)\cos(2\pi f_j t) \leq \epsilon, \quad j = N'_f + 1, \ldots, N_f + N'_f$$

Primetimo da je ovo svodjenje isto kao kod metode ϵ ograničenja iz prethodne glave.

Razmotrimo sada problem reprodukcije signala pomoću tri FIR filtra, pri čemu svaki propušta različiti opseg frekvencija.

Imamo da je spektar paralelne veze ovih filtara:

$$H(f) = H_w(f) + H_m(f) + H_t(f)$$

gde su $H_w(f)$, $H_m(f)$ i $H_t(f)$ gde su redom spektri svakog filtra (indeksi potiču od engleskih reči w-woofer, m-midrange, t-tweetier).

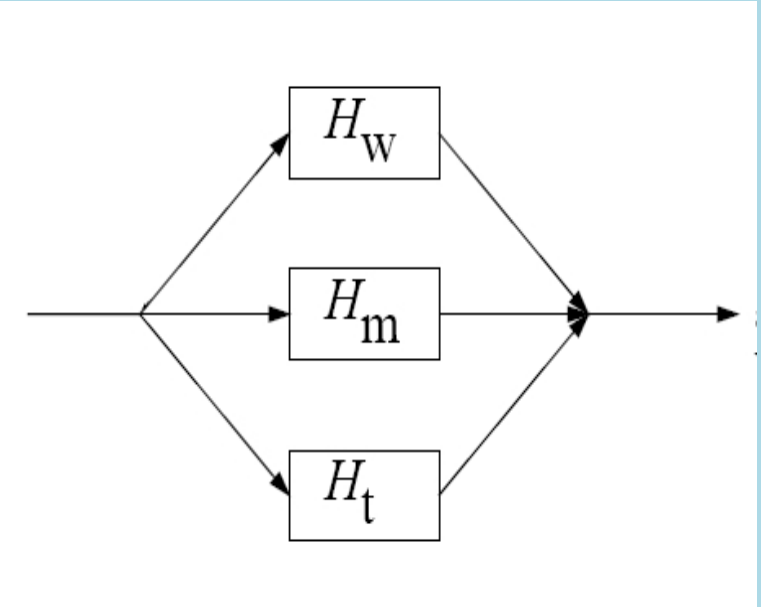
Propusni opsezi ($\mathcal{O}$) za ove filtre su redom $\mathcal{O}_w = [0, f_w]$,$\mathcal{O}_m = [0, f_{m1}] \cup [f_{m2}, \frac{1}{2}]$ i $\mathcal{O}_t = [f_t, \frac{1}{2}]$.

U ovom slučaju nam je cilj da funkcija $A(f)$ bude što bliža jedinici za $f \in [0, \frac{1}{2}]$.

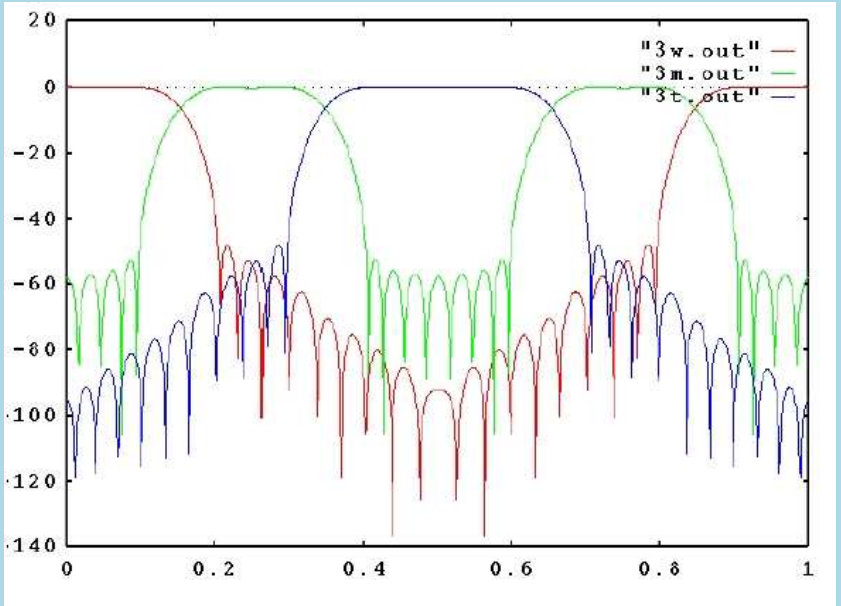
$$\min \int_0^{1/2} |A_w(f) + A_m(f) + A_t(f) - 1|$$
$$p.o. \ -\epsilon \leq A_i(f) \leq \epsilon, \quad f \in \mathcal{O}_i, \quad i = w, m, t$$

koji na sličan način svodimo na problem linearnog programiranja.

Na slici su prikazane funkcije $A_w(f)$, $A_m(f)$ i $A_t(f)$ za sledeće vrednosti parametara: $N_f = 1000$, $f_w = 0.2$, $f_t = 0.7$, $f_{m1} = 0.1$ i $f_{m2} = 0.4$.



Blok šema paralelne veze tri filtra.



Grafici funkcija $A_w(f)$, $A_m(f)$ i $A_t(f)$.

6 Zaključak

Izložićemo kratak pregled svih izloženih rezultata kao i neke ideje za dalja istraživanja:

A. Detaljno smo proučili problem linearnog programiranja.

[A.1.] Definisali smo matematički model i osnovne oblike problema linearnog programiranja.

[A.2.] Definisali smo i proučili osnovne osobine skupa dopustivih rešenja.

[A.3.] Formulisali smo geometrijski metod za rešavanje problema linearnog programiranja. Pokazali smo kako u specijalnim slučajevima izgleda skup dopustivih rešenja.

[A.4.] Prezentovali smo našu originalnu implementaciju geometrijskog metoda u programskom jeziku MATHEMATICA.

B. Proučili smo sve faze simpleks metoda linearnog programiranja.

[B.1.] Posmatrali smo kanonski oblik i nekoliko slučajeva koji mogu da nastupe u zavisnosti od znaka elemenata vektora funkcije cilja.

[B.2.] Definisali smo pojam Takerove tabele i pokazali smo kako se vrši zamena bazične i nebazične promenljive u Takerovoj tabeli.

[B.3.] Formulisali smo metod za nalaženje prvog bazično dopustivog rešenja.

[B.4.] Ukazali smo na problem nagomilavanja računske greške kod simpleks metoda i prezentovali revidirani simpleks metod kojim se taj problem rešava.

[B.5.] Proučili smo problem cikliranja simpleks metoda.

[B.6.] Dokazali smo da je složenost simpleks metoda eksponencijalna.

C. Prezentovali smo nekoliko naših originalnih modifikacija simpleks [6, 5] i revidiranog simpleks metoda [1].

[C.1.] Najpre smo uočili dva nedostatka klasičnog algoritma za nalaženje prvog bazično dopustivog rešenja [6].

[C.2.] Teorijskim razmatranjima smo zatim došli do metoda koji nema ove nedostatke.

[C.3.] Pokazali smo kako se prethodno izložena modifikacija može poboljšati.

[C.4.] Na sličan način, uočili smo i nedostatke revidiranog simpleks metoda i teorijski došli do metoda kojim se oni prevazilaze.

[C.5.] Sve prethodno izložene algoritme simpleks metoda kao i originalne modifikacije implementriali smo u programskom jeziku Visual Basic 6.0. Tako je nastao program MarPlex. Program je testiran na referentnim svetskim test primerima, i pritom smo pokazali kako se primenom izloženih modifikacija sa manjim brojem iteracija dolazi do optimalnog rešenja.

[C.6.] Revidirani simpleks metod smo implementirali u programskom jeziku MATHEMATICA. Tako je nastao još jedan naš originalni softver RevMarPlex.

[C.7.] Oba naša programa smo testirali na klasi veoma loše uslovljenih test primera koje program PCx nije bio u stanju da reši. I u ovom slučaju su oba programa dali korektna rešenja, pri čemu je preciznost RevMarPlex-a bila znatno veća.

D. Proučili smo problem višekriterijumske optimizacije koji je takodje, kao problem linearnog programiranja, veoma primenljiv u praksi.

[D.1.] Definisali smo problem višekriterijumske optimizacije (VKO) kao i osnovne pojmove vezane za njega.

[D.2.] Formulisali smo nekoliko klasičnih metoda za rešavanje problema VKO. Za svaki metod smo dali originalnu implementaciju [8] u programskom jeziku MATHEMATICA.

[D.3.] Razmatrali smo neke specijalne slučajeve problema VKO i pokazali smo kako se, pod odredjenim uslovima, na njih mogu primeniti simpleks metod i teorija generalisanih inverza. Takodje smo ukazali na moguću primenu naših originalnih rezultata [2, 3, 7, 4] iz teorije generalisanih inverza u tim slučajevima problema VKO.

E. Pokazali smo dve moguće primene problema linearnog programiranja i višekriterijumske optimizacije. Prva primena je u astronomiji (optici) a druga u telekomunikacijama (elektronici).

E.1. Prva primena se odnosi na projektovanje optimalne maske objektiva . Formulisali smo problem i pokazali kako se on svodi, prvo na problem VKO, a onda i na problem linearnog programiranja. Grafički su prikazani rezultati dobijeni u konkretnom slučaju.

E.2. Druga primena se odnosi na projektovanje FIR filtara. I ovde smo formulisali problem i sveli ga na problem linearnog programiranja. Razmotrili smo i modifikovanu verziju problema koju smo takodje sveli na problem linearnog programiranja, a zatim smo prikazali rezultate u jednom konkretnom slučaju.

Objavljeni radovi

[1] M.D. Petković, P.S. Stanimirović, N.V. Stojković, Two modifications of revised simplex metod, Matematicki Vesnik, 54 (2002), pp. 163–169.

[2] M.D. Petković, P.S. Stanimirović, Partitioning method for two-variable rational and polynomial matrices, Mathematica Balcanica vol. 19, 2005, pp. 185–194.

[3] M.D. Petković, P.S. Stanimirović, Symbolic computation of the Moore-Penrose inverse using partitioning method, International Journal of Computer Mathematics, 82(March 2005), pp. 355–367.

[4] M.D. Petković, P.S. Stanimirović, Interpolation algorithm of Leverrier-Faddev type for polynomial matrices, Numerical Algorithms, prihvaćeno za štampu.

[5] P.S. Stanimirović, N.V. Stojković, M.D. Petković, Modification and implementation of two phases simplex method, biće objavljeno.

[6] P.S. Stanimirović, N.V. Stojković, M.D. Petković, Several Modifications of Simplex Method, Filomat, 2003.

[7] P.S. Stanimirović, M.D. Petković, Computation of generalized inverses of polynomial matrices by interpolation, Appl. Math. Comput., Vol 172/1 (January 2006), pp 508-523.

[8] P.S. Stanimirović, N.V. Stojković, M.D. Petković, Run-time transformations in implementation of linear multi-objective optimization, PRIM, Budva 2004.

[9] M. Tasić, P.S. Stanimirović, I.P. Stanimirović, M.D. Petković, N.V. Stojković, Some useful MATHEMATICA teaching examples, Facta Universitatis(Niš) Series Electronics and Energetics, vol. 18, No. 2, August 2005, pp. 329-344.

Predlozi za dalja istraživanja

1. Prezentovane modifikacije metoda za nalaženje prvog bazično dopustivog rešenja mogle bi se primeniti na algoritme za eliminaciju jednačina i slobodnih promenljivih.
2. Pokazali smo da teorija generalisanih inverza ima primenu kod problema VKO. Ovu vezu dve fundamentalno različite oblasti bi trebalo detaljnije proučiti.
3. U delu vezanom za složenost simpleks metoda ukazali smo na očekivan broj iteracija simpleksa na uopštenom Minti-Kli primeru, ukoliko se pivot element bira slučajno. Sličan rezultat mogao bi da se izvede i za ostale izložene faze simpleks metoda kao i za modifikacije.

Zahvalnost

- Zahvaljujem se Dr. Nebojši Stojkoviću na nesebičnoj pomoći kako pri izradi ovog rada, tako i uopšte.
- Zahvalnost takodje dugujem i profesoru Dr. Predragu Rajkoviću koji mi je takodje pružio nesebičnu pomoć i uveo me u jednu drugu oblast matematike (teoriju polinoma i polinomnih identiteta).
- Sa posebnim zadovoljstvom zahvaljujem se mom profesoru i mentoru, Prof. Dr. Predragu Stanimiroviću, ne samo na pomoći pri izradi ovog rada, već i na velikoj pažnji i vremenu koje mi je posvetio uvodeći me u probleme matematičkog programiranja, a time i u naučni rad.

Hvala na pažnji!