Informatyka, studia dzienne, mgr II st. semestr I
Metody obliczeniowe optymalizacji **2011/2012**
Prowadzący: mgr inż. Łukasz Chomątek czwartek, 16:00

Data oddania: _____ Ocena: _____

Paweł Musiał 178726
Łukasz Michalski 178724

Zadanie 4 - Programowanie liniowe*

1. Cel

Napisać program implementujący rozwiązywanie zagadnienia programowania liniowego za pomocą dwufazowej metody sympleksu. Program powinien wykrywać sytuacje patologiczne (brak rozwiązań, nieskończenie wiele rozwiązań).

2. Rozwiązanie zadania

2.1. Problem programowania liniowego

Programowaniem liniowym nazywamy problem optymalizacji wielowymiarowej funkcji liniowej z ograniczeniami. Zapisać możemy go jako minimalizacja wyrażenia :

$$c^T x \tag{1}$$

z ograniczeniami :

$$Ax = b \tag{2}$$

$$x \geq 0 \tag{3}$$

Zakładając, że $b \geq 0$
gdzie :

* SVN: `https://serce.ics.p.lodz.pl/svn/labs/moo/lc_cz1600/lmpm`

- x - zmienne funkcji celu
- c - współczynniki funkcji celu
- A - macierze współczynników ograniczeń
- b - prawa strona ograniczeń

Przedstawioną powyżej postać, nazywamy postacią standardową, aby z dowolnej postaci problemu programowania liniowego

$$c^T x \tag{4}$$
$$Ax \geq b \tag{5}$$
$$x \geq 0 \tag{6}$$

przejść do tej postaci należy : w przypadku maksymalizacji, przekształcamy zadanie na minimalizacje funkcji przeciwnej funkcji celu; w związku z założoną nie ujemnością prawej strony ograniczeń, jeśli wartość b_i jest ujemna, przemnażamy ograniczenie przez -1. Następnie wprowadzamy sztuczne zmienne y_i. A problem w standardowej postawi przyjmuje się jako :

$$c^T x \tag{7}$$
$$[\, A \,, w - I_m \,] \begin{bmatrix} x \\ y \end{bmatrix} = b \tag{8}$$
$$y \geq 0 \tag{9}$$

gdzie I_m jest macierzą jednostkową o wymiarach mxm
lub gdy ograniczenia przybierają formę : $Ax \geq b$
ograniczenia w standardowej formie przybierają formę : $[\, A \,, I_m \,] \begin{bmatrix} x \\ y \end{bmatrix} = b$

2.2. Metoda sympleks

Znając już postać problemu możemy przejść do opisu rozwiązania. W algorytmie tym układ równań liniowych $Ax = b$ (gdzie $rz(A) = m$) zapisujemy w postaci „kanonicznej", którą możemy otrzymać na drodze przekształceń elementarnych :

$$[\, I_m \,, Y_{m,n-m} \,]\, x = y_0$$

Piersze m zmiennych x występuje tylko w jednym równaniu, a współczynniki przy nich są równe 1, zmienne ta nazywamy zmiennymi bazowymi, w innym przypadku niebazowymi.

Związana z tą postacią jest macierz kanoniczna : $[I_m, Y_{m,n-m}, y_0]$.

1. Rozpoczynamy od pewnego dopuszczalnego rozwiązania x= $[x_1, x_2, ..., x_m, 0, ..., 0]^T$ gdzie $x_i \geq 0$ dla $i \in 1..m$. Wyznaczamy macierz kanoniczną dla tego rozwiązania.
2. Obliczamy współczynniki zredukowanego kosztu względem każdej zmiennej nie bazowej - $r_i = c_i - z_i$ gdzie $z_i = c_1 y_{1i} + \cdots + c_m y_{mi}$.
3. Jeśli $\forall\ j\ r_j \geq 0$ to aktualne rozwiązanie jest optymalne. Koniec.
4. Wybierze q dla którego $r_q < 0$.
5. Jeśli nie istnieje $y_{iq} > 0$ problem jest nieograniczony - koniec. W innym przypadku oblicz $p = min\{j : \frac{y_{j0}}{y_{jq}} = min_i\{\frac{y_{i0}}{y_{iq}} : y_{iq} > 0\}\}$.

6. zmień bazę zmiennych p i q w następujący sposób:

$$y'_{ij} = y_{ij} - \frac{y_{pj}}{y_{pq}} y_{iq},\ dla\ i \neq p \tag{10}$$

$$y'_{pj} = \frac{y_{pj}}{y_{pq}} \tag{11}$$

7. Wróć do kroku 2.

Powyższy algorytm opisuje kolejne kroki metody rozwiązywania problemów programowania liniowego.

Jak już powyżej opisano, zmienne w ograniczeniach rozdzielamy na bazowe i niebazowe. Na podstawie tego rozgraniczenia możemy zapisać problem programowania liniowego, oznaczając cześć dotyczącą zmiennych bazowych poprzez B a niebazowych przez D.
Możemy teraz opisać zadanie jako :

$$min\ c_B^T x_B + c_D^T x_D \tag{12}$$

$$\left[\, {}_B\ {}_D \,\right] \left[\begin{smallmatrix} x_B \\ x_D \end{smallmatrix} \right] = b \tag{13}$$

$$x_B \geq 0, x_D \geq 0 \tag{14}$$

Jeśli wektor $x_D \neq 0$, czyli aktualne rozwiązani nie jest optymalnym, możemy wyznaczyć wektor kosztów zredukowanych :

$$r_D^T = c_D^T - c_B^T B^{-1} D \tag{15}$$

Możemy teraz przedstawić postać macierzy sympleksowej :

$$\left[\begin{smallmatrix} A & b \\ c^T & 0 \end{smallmatrix} \right] = \left[\begin{smallmatrix} B & D & b \\ c_B^T & c_D^T & \end{smallmatrix} \right] \tag{16}$$

Algorytm sympleksu działa na macierzy kanonicznej, należy sprowadzić zatem macierz sympleksową do tej formy, po przekształceniach, które można dokładnie poznać w [1]. A w wyniku otrzymujemy :

$$\begin{bmatrix} I_m & B^{-1}D & B^{-1}b \\ 0^T & c_D^T - c_B^T B^{-1} D & -c_B^T B^{-1} b \end{bmatrix} \tag{17}$$

Rozważmy teraz przykład :
maksymalizacja wyrażenia $7x_1 + 6x_2$
z ograniczeniami $2x_1 + x_2 \leq 3$, $x_1 + 4x_2 \leq 4$, $x_1, x_2 \geq 0$

Zadaniem jest maksymalizacja, zatem w tablicy sympleksowej współczynniki funkcji celu c będą zanegowane, oraz wprowadzone zostaną dwie nieujemne sztuczne zmienne. Zatem tablica sympleksowa będzie miała postać :

2	1	1	0	3
1	4	0	1	4
-7	-6	0	0	0

Zauważmy, że tablica ta jest już w postaci kanonicznej, wyznaczamy min $r_i = r_1 = -7 \implies$ q=1, następnie wyznaczamy p zgodnie z algorytmem - p=1. Zatem wykonujemy operacje zmiany bazy (1,1).

1	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{3}{2}$
0	$\frac{7}{2}$	$-\frac{1}{2}$	1	$\frac{5}{2}$
0	$-\frac{5}{2}$	$\frac{7}{2}$	0	$\frac{21}{2}$

Jedynie r_2 jest ujemne, wybieramy q=2, wybieramy zgodnie z algorytmem p=2, Wykonujemy operacje zmiany bazy (2,2).

1	0	$\frac{4}{7}$	$-\frac{1}{7}$	$\frac{8}{7}$
0	1	$-\frac{1}{7}$	$\frac{2}{7}$	$\frac{5}{7}$
0	0	$\frac{22}{7}$	$\frac{5}{7}$	$\frac{86}{7}$

Ponieważ żadna wartość z ostatniego wiersza nie jest ujemna, kończymy działania algorytmu. wyznaczamy rozwiązanie jako $x_1 = \frac{8}{7}$, $x_2 = \frac{5}{7}$, oraz wartość funkcji celu $\frac{86}{7}$.

2.3. Metoda dwu-fazowego sympleksu

W poprzedniej sekcji, założyliśmy, że początkowe rozwiązanie jest dopuszczalne, jednak takie założenie może być błędne. Metoda dwu-fazowego sympleksu wyznacza w pierwszej fazie początkowe rozwiązanie dopuszczalne, a następnie oblicza na jego podstawie rozwiązanie podstawowego problemu. Początkowe rozwiązanie dopuszczalne wyznaczamy rozwiązując zadanie pomocnicze.
Rozwiązujemy zadanie minimalizacji
$y_1 + y_2 + \cdots + y_m$
przy ograniczeniach $[A, I_m]\begin{bmatrix} x \\ y \end{bmatrix} = b$
$x, y \geq 0$
gdzie $y = [y_1 + y_2 + \cdots + y_m]^T$ jest wektorem zmiennych sztucznych, oraz wiemy, że początkowe rozwiązanie dopuszczalne ma postać $\begin{bmatrix} 0 \\ b \end{bmatrix}$. Po rozwiązaniu zadania pomocniczego możemy wyznaczyć tablice sympleksową w postaci kanonicznej problemu podstawowego odrzucając sztuczne zmienne, oraz wracając do podstawowej funkcji celu.

3. Opis programu

3.1. Algorytm podstawowy : metoda sympleks

```
function [subs, A, z]= simplex(A, subs, mm, k)
[m, n] = size(A);
[mi, col] = BlandRule(A(m,1:n-1));
while ~isempty(mi) & mi < 0 & abs(mi) > eps
    t = A(1:m-k,col);
    if all(t <= 0)
        if mm == 0
     z = -inf;
        else
     z = inf;
        end
        fprintf('\nnieograniczone rozwiazanie z= %s\n',z)
        return
    end
    c = 1:m;
    a=A(1:m-k,n);
    b=A(1:m-k,col);
    l = c(b > 0);
    [small, row] = min(a(l)./b(l));
```

```
    row = l(row);
    if ~isempty(row)
        if abs(small) <= 100*eps & k == 1
     [s,col] = BlandRule(A(m,1:n-1));
        end
        A(row,:)= A(row,:)/A(row,col);
        subs(row) = col;
        for i = 1:m
     if i ~= row
         A(i,:)= A(i,:)-A(i,col)*A(row,:);
     end
        end
        [mi, col] = BlandRule(A(m,1:n-1));
    end
end
z = A(m,n);
end
```

simplex

3.2. Dwu fazowy sympleks

```
function  tpsimplex(type, c, A, rel, b)

if (type == 'min')
    mm = 0;
else
    mm = 1;
    c = -c;
end
c=c(:)';b=b(:);
[m, n] = size(A);
n1 = n;
les = 0;
if length(c) < n
    c = [c zeros(1,n-length(c))];
end
for i=1:m
    artificial_var =zeros(m,1);
    artificial_var(i)=1;
    if(rel(i) == '<')
        A = [A artificial_var];
        les = les + 1;
    elseif(rel(i) == '>')
        A = [A -artificial_var];
    end
end
ncol = length(A);

if les == m
    c = [c zeros(1,ncol-length(c))];
    A = [A;c];
    A = [A [b;0]];
    [subs, A, z] = simplex(A, n1+1:ncol, mm, 1);
else
    A = [A eye(m) b];
```

```
    if m > 1
        w = -sum(A(1:m,1:ncol));
    else
        w = -A(1,1:ncol);
    end
    c = [c zeros(1,length(A)-length(c))];
    A = [A;c];
    A = [A;[w zeros(1,m) -sum(b)]];
    subs = ncol+1:ncol+m;
    av = subs;
    [subs, A, z] = simplex(A, subs, mm, 2);

    nc = ncol + m + 1;
    x = zeros(nc-1,1);
    x(subs) = A(1:m,nc);
    xa = x(av);
    com = intersect(subs,av);
    if (any(xa) ~= 0)
        fprintf('Brak rozwiazan\n')
        return
    end
    A = A(1:m+1,1:nc);
    A =[A(1:m+1,1:ncol) A(1:m+1,nc)];
    [subs, A, z] = simplex(A, subs, mm, 1);
end

if (z == inf | z == -inf)
    return
end
[m, n] = size(A);
x = zeros(n,1);
x(subs) = A(1:m-1,n);
x = x(1:n1);
if mm == 0
    z = -A(m,n);
else
    z = A(m,n);
end

disp(x(1:n1))
disp(z)

t = find(A(m,1:n-1) == 0);
if length(t) > m-1
    fprintf('Problem ma nieskonczenie wiele rozwiazan\n');
end

end
```

two–phase simplex

4. Wyniki

Dla danych wejściowych :

```
type = 'min';
c = [7 6];
A = [2 1;1 4];
rel = '<<';
b = [3 4];
```

```
Tableau
       2       1       1       0       3
       1       4       0       1       4
      -7      -6       0       0       0
wymiana bazy 1  -> 1
Tableau
       1       1/2     1/2     0       3/2
       0       7/2          -1/2     1       5/2
       0      -5/2     7/2     0      21/2
wymiana bazy 2  -> 2
Tableau
       1       0       4/7          -1/7     8/7
       0       1      -1/7     2/7     5/7
       0       0      22/7     5/7          86/7
Rozwiazanie optymalne:
       8/7
       5/7
wartosc funkcji celu:
      86/7
```

Dla danych wejściowych :

```
type = 'min';
c = [7 6];
A = [2 1;1 4];
rel = '<<';
b = [3 4];
```

```
Tableau
       2      -1       1       6      -5      -1       0       1       0       6
       1       1       2       1       2       0      -1       0       1       3
       3       4       6       7       1       0       0       0       0       0
      -3       0    - 3      -7       3       1       1       0       0      -9
wymiana bazy 1  -> 1
Tableau
       1      -1/2     1/2     3      -5/2    -1/2     0       1/2     0       3
       0       3/2     3/2    -2       9/2     1/2    -1      -1/2     1       0
       0      11/2     9/2    -2      17/2     3/2     0      -3/2     0      -9
       0      -3/2    -3/2     2      -9/2    -1/2     1       3/2     0       0
wymiana bazy 2  -> 2
Tableau
```

```
       1       0       1       7/3    -1      -1/3   -1/3     1/3     1/3     3
       0       1       1      -4/3     3       1/3   -2/3    -1/3     2/3     0
       0       0      -1      16/3    -8      -1/3   11/3     1/3   -11/3    -9
       0       0       0       0       0       0       0       1       1       0
Koniec fazy 1
Tableau
       1       0       1       7/3    -1      -1/3   -1/3     3
       0       1       1      -4/3     3       1/3   -2/3     0
       0       0      -1      16/3    -8      -1/3   11/3    -9
wymiana bazy 2  -> 3
Tableau
       1      -1       0      11/3    -4     -2/3     1/3     3
       0       1       1      -4/3     3      1/3    -2/3     0
       0       1       0       4      -5      *        3          -9
wymiana bazy 2  -> 5
Tableau
       1       1/3     4/3     17/9     0     -2/9     -5/9      3
       0       1/3     1/3     -4/9     1      1/9     -2/9      0
       0       8/3     5/3     16/9     0      5/9     17/9     -9
Koniec fazy 2

Rozwiazanie optymalne:
       3
       0
       0
       0
       0
       0
       0

 wartosc funkcji celu:
       9
```

5. Wnioski

Przedstawiona powyżej implementacja algorytmu dwu-fazowego sympleksu spełnia swoje zadanie jako metody rozwiązywania problemów programowania liniowego. Rozstrzyga również sytuacje trudne, takie jak : nieskończenie wiele rozwiązań, brak rozwiązań. Również zastosowana modyfikacja podstawowego sposobu doboru zmiennych do operacji zmiany bazy - „regułą Blanda" pozwala uniknąć zablokowania się algorytmu podstawowego - metody sympleksu.

Literatura

[1] „An Introduction to Optimization" , Edwin Kah Pin Chong and Stanislaw H. Zak, Hoboken, EUA : Wiley-Interscience 2008