

# Lineáris programok megoldása a szimplex algoritmussal (Segédlet)

Rétvári Gábor  
E-mail: [retvari@tmit.bme.hu](mailto:retvari@tmit.bme.hu)

## A GNU Octave alapú szimplex megoldószoftver

A szimplex táblák használatának elsajátításához rendelkezésre áll egy egyszerű Octave/MATLAB alapú szoftver, mely kiszámítja a szimplex algoritmus egyes iterációinak megfelelő szimplex táblákat és ezeket egyszerű formában kiírja.

A primál és a duál szimplex algoritmusok külön szkriptekben érhetőek el:

- `psimplex.m`: primál szimplex,  
<http://lendulet.tmit.bme.hu/~retvari/courses/VITMD097/psimplex.m>
- `dsimplex.m`: duál szimplex,  
<http://lendulet.tmit.bme.hu/~retvari/courses/VITMD097/dsimplex.m>

A kód a GNU Octave matematikai szoftverkörnyezethez készült, de (elvileg) változtatás nélkül futtatható MATLAB alatt is. A fájlok letöltése után helyezzük azokat az aktuális könyvtárba, vagy bárhova, ahol az Octave megtalálja (tehát az Octave path alá, lásd: Octave info pages: `addpath`).

Az installálás sikerességét az alábbi módon ellenőrizhetjük:

```
octave> help psimplex
```

```
PSIMPLEX - simplex algorithm for maximization problems in standard form
```

```
[A, subs, x, z] = psimplex(c, A, b, subs, varargin)
```

```
-----  
DESCRIPTION  
-----
```

```
The simplex algorithm for the LP problem
```

```
          max z = c*x  
Subject to: Ax  = b  
           x  >= 0
```

```
The fifth input parameter is optional. If it is set to 'y', then the  
initial and final tableaux are displayed to the screen.
```



tábla nulladik sora a szkript által kijelzett mátrix utolsó sorában olvasható (ennek program-szervezési okai vannak). A RHS természetesen marad az utolsó oszlopban, a célfüggvény aktuális értéke pedig az utolsó sor utolsó oszlopából olvasható le. A fenti mátrix tehát az alábbi táblának felel meg.

	$z$	$x_1$	$x_2$	$x_3$	$x_4$	RHS
$z$	1	1	-2	0	0	0
$x_3$	0	3	4	1	0	12
$x_4$	0	2	-1	0	1	12

A második lépésben a szkript pivot operációk sorozatával szimplex tábla alakra hozza a kezdeti táblát. Ebben a formában a bázisváltozók oszlopai egységmátrixot alkotnak és a hozzájuk tartozó elemek a nulladik sorban mind nullák. Ez azt jelenti, hogy nem muszáj egységbázisról indítani a megoldást. A folyamat során a szkript ellenőrzi, hogy a kezdeti bázis nemszinguláris és megengedett-e.

```

-----
Initial tableau

A =

    3    4    1    0    12
    2   -1    0    1    12
    1   -2    0    0     0

Initial basis

subs =

    3    4

Press any key to continue ...

```

A kezdeti szimplex tábla most nem változik, mert a bázisunk egységmátrixot alkot és a célfüggvényben nem szerepelnek bázisváltozók. A következő lépésekben a szkript fokozatosan optimalizálja a szimplex táblát.

```

-----
2 enters, 3 leaves the basis

A =

0.75000    1.00000    0.25000    0.00000    3.00000
2.75000    0.00000    0.25000    1.00000   15.00000
2.50000    0.00000    0.50000    0.00000    6.00000

```

```
current basis:
subs =
```

```
2 4
```

```
Press any key to continue ...
```

Minden lépésben meghatározza a bázisba belépő, illetve az onnan kilépő változókat, és elvégzi a pivotot. Ne felejtjük, hogy a kijelzett tábla formátuma speciális, mert nulladik sor az utolsó sorba került. A fenti tábla tehát a „szokásos” formátumban:

	$z$	$x_1$	$x_2$	$x_3$	$x_4$	RHS
$z$	1	2.5	0	0.5	0	6
$x_2$	0	0.75	1	0.25	0	3
$x_4$	0	2.75	0	0.25	1	15

Az iteráció során a szkript folyamatosan ellenőrzi, hogy korlátos pivotot hajtott-e végre, és ha nemkorlátos bázist talál, akkor kiszáll. A degenerációt a szkript nem ellenőrzi, ezért speciális esetekben a futás végtelen ciklusba kerülhet. Végül, ha az aktuális bázis optimális, mint esetünkben, akkor a szkript futása a végeredmény kiírásával megáll.

```
-----
Optimal tableau, end
ans =

0.75000    1.00000    0.25000    0.00000    3.00000
2.75000    0.00000    0.25000    1.00000    15.00000
2.50000    0.00000    0.50000    0.00000    6.00000
```

A mi esetünkben a célfüggvény értéke 6, az optimális megoldás pedig a táblából kiolvasható: a nembázis változók értéke 0, vagyis  $x_1 = x_3 = 0$ , míg a bázisváltozók indexeinek halmaza  $B = \{2, 4\}$ , és az ezeknek megfelelő sorokban a RHS kiolvasásával  $x_2 = 3$ ,  $x_4 = 15$  adódik.

Ha csak a végeredmény érdekel, akkor hagyjuk el az opcionális utolsó paramétert a `psimplex` hívásakor:

```
psimplex(c, A, b, [3,4])
```

Ha az eredményt szeretnénk felhasználni:

```
octave> [T, subs, x, z] = psimplex(c, A, b, [3,4]);
```

Ebben az esetben a szkript az optimális szimplex táblát a `T`, az optimális bázis oszlopainak indexeit a `subs`, az optimális megoldást az `x`, és az optimális célfüggvényértéket pedig a `z` változóban adja vissza.

A duál szimplex algoritmust megvalósító `dsimplex` függvény hasonlóan működik.

(Megjegyzés: vizsgán plusz jegyért lehet a kódot finomítani: szebb output megírása, cycling kezelése, kezdeti bázis kereső módszer implementálása. Mielőtt nekiesnél, írj nekem egy emailt!)