

Προβλήματα Γραμμικού Προγραμματισμού – Αλγόριθμος SIMPLEX

Στα προβλήματα γραμμικού προγραμματισμού (ΠΓΠ) ζητείται να υπολογιστεί η βέλτιστη τιμή μιας αντικειμενικής συνάρτησης (ΑΣ), υπό τους περιορισμούς (ΥΤΠ) που θέτει το πρόβλημα. Στα προβλήματα μεγιστοποίησης αναζητούμε τη μέγιστη τιμή της ΑΣ (max) ενώ στα προβλήματα ελαχιστοποίησης την ελάχιστη τιμή της (min).

Αν το πρόβλημα έχει μόνο δύο μεταβλητές τότε μπορεί να λυθεί και γραφικά. Όμως στη γενική τους μορφή τα ΠΓΠ έχουν πολλές μεταβλητές και πολλούς περιορισμούς, και η λύση τους είναι δυνατή μόνο με τη βοήθεια του αλγορίθμου SIMPLEX και ενός Η/Υ που θα τον υλοποιεί ταχύτατα. Υπάρχουν αρκετές εφαρμογές λογισμικού που λύνουν προβλήματα βελτιστοποίησης υλοποιώντας τον αλγόριθμο SIMPLEX (π.χ., MathCAD, MS-Excel(Solver), κ.ά.). Το MatLab εκτός από τα προγράμματα ποσοφ δημιουργεί ο χρήστης διαθέτει τις τυπικές συναρτήσεις ελαχιστοποίησης (fminsearch, fminbnd, κλπ.) και διαθέτει το Optimization Toolbox το οποίο περιλαμβάνει επιπλέον συναρτήσεις για την επίλυση προβλημάτων βελτιστοποίησης (linprog, optimtool, κλπ.).

Στο παράδειγμα που ακολουθεί θα δούμε πως μπορούμε να δημιουργήσουμε στο MatLab ένα απλό πρόγραμμα το οποίο να υλοποιεί τον αλγόριθμο SIMPLEX και να λύνει ένα τυπικό ΠΓΠ μεγιστοποίησης καθώς επίσης και πως το λύνουμε με τα εργαλεία του MatLab.

1) Το πρόγραμμα μας αποτελείται από τρία τμήματα: Α) μια συνάρτηση **Spivot.m** που όταν κληθεί βρίσκει το οδηγό στοιχείο (pivot) για το επόμενο βήμα της Simplex, Β) μια συνάρτηση **Sstep.m** που όταν κληθεί εκτελεί ένα βήμα της μεθόδου Simplex, και, Γ) το κυρίως πρόγραμμα **Simplex.m** που διαβάζει τον Αρχικό πίνακα Simplex και καλεί τις δύο πιο πάνω συναρτήσεις όσες φορές χρειαστεί (βήματα) μέχρι να βρεθεί η βέλτιστη λύση.

Έστω ότι μας δίνεται το παρακάτω ΠΓΠ:

$$\begin{aligned} \text{ΑΣ: maximize } P &= 5x_1 + 2x_2 + 3x_3 \\ \text{ΥΤΠ: } x_1 + 2x_2 + 3x_3 &\leq 9 \\ 3x_1 + 2x_2 + 2x_3 &\leq 5 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

Ο Αρχικός πίνακας Simplex που περιέχει και τις μεταβλητές περιθωρίου (slack) θα έχει τη μορφή:

	x_1	x_2	x_3	S_1	S_2	P	
S_1	1	2	3	1	0	0	9
S_2	3	2	2	0	1	0	5
P	-5	-2	-3	0	0	1	0

Μεταφέρουμε τις τιμές του αρχικού πίνακα στο πρόγραμμα simplex.m στη θέση Initialization και στη συνέχεια το εκτελούμε.

M-file: **simplex1.m**

```
% Program simplex1.m (Simplex Method for Linear Progr.)
% Initialization
%
%      X1 X2 X3 S1 S2 P   C
SimplexInitMatrix = [ 1 2 3 1 0 0   9   % S1
                     3 2 2 0 1 0   5   % S2
                     -5 -2 -3 0 0 1   0 ]; % P
% Simplex Iterations
Bima = 0
Smat = SimplexInitMatrix
[PivRow, PivCol, Telos] = Spivot(Smat);
while ~Telos
    Bima = Bima + 1
    Smat = Sstep(Smat, PivRow, PivCol)
    [PivRow, PivCol, Telos] = Spivot(Smat);
end
```

M-file: Sstep.m

```
function NewSmat = Sstep(Smat, PivotRow, PivotCol)
% Simplex Method - Function for One Step Simplex on Smat
% using Pivot element: Smat(PivotRow,PivCol)
[P, C] = size(Smat);
for i = 1:P,
    NewSmat(i,:)=Smat(i,:)- ...
        Smat(i,PivotCol).*Smat(PivotRow,:)./Smat(PivotRow,PivotCol);
end
NewSmat(PivotRow,:) = Smat(PivotRow, :)./Smat(PivotRow,PivotCol);
```

M-file: Spivot.m

```
function [PivRow, PivCol, Telos] = Spivot(Smat)
% Simplex Method - Spivot Function finds Pivot element in Smat
[P, C] = size(Smat);
if sum(Smat(P,1:C-1)<0)==0,
    Telos = 1; % Ola ta P einai (+,0) TELOS epanelipsewn Simplex.
    PivRow = 0; PivCol = 0;
else
    [minP, minPindex] = min(Smat(P,1:C-1)); % Find Pivot_Col.
    if (sum(Smat(1:P-1,minPindex)>0)==0),
        Telos = -1; % Pivot_Col den exei kanena (+) (ALYTO)
        PivRow = 0; PivCol = 0;
    else
        Telos = 0; % Yparxei P (-) & Pivot_Col (+), brethike Pivot.
        for i = 1:P-1,
            if Smat(i,minPindex)>0,
                SmatCdivided(i) = Smat(i,C)./Smat(i,minPindex);
            else
                SmatCdivided(i) = 1/eps; % to kanei HUGE an element <=0
            end
        end
        [minC, minCindex] = min(SmatCdivided);
        PivRow = minCindex;
        PivCol = minPindex;
    end
end
```

```
>> simplex1
Bima =
    0
Smat =
    1     2     3     1     0     0     9
    3     2     2     0     1     0     5
   -5    -2    -3     0     0     1     0
Bima =
    1
Smat =
    0    4/3    7/3     1   -1/3     0   22/3
    1    2/3    2/3     0    1/3     0    5/3
    0    4/3    1/3     0    5/3     1   25/3
```

Το αποτέλεσμα είναι ότι η βέλτιστη λύση επιτυγχάνεται για $x_1 = 5/3$ & $x_2 = x_3 = 0$, και ότι η βέλτιστη τιμή της ΑΣ θα είναι $P = 25/3$. (Για να πάρουμε τα αποτελέσματα σε μορφή κλάσματος έχουμε δώσει προηγουμένως την εντολή: **format rat**).

2) Τα αποτελέσματα του προγράμματος μπορούμε να τα συγκρίνουμε και με τα αποτελέσματα των **έτοιμων συναρτήσεων** που διαθέτει το MatLab μέσω του **Optimization Toolbox**. Η συνάρτηση που θα χρησιμοποιήσουμε είναι η **linprog()** και ο πιο συνηθισμένος τρόπος που συντάσσεται είναι ο εξής: **[x, fval] = linprog(f, A, b, Aeq, beq, lb, ub)**, όπου, x είναι οι μεταβλητές που αναζητάμε, f είναι η αντικειμενική συνάρτηση που θα ελαχιστοποιηθεί, fval τη τιμή που παίρνει τελικά, $A*x \leq b$ είναι οι περιορισμοί (ανισότητες) του προβλήματος, $Aeq*x = beq$ είναι άλλοι τυχόν περιορισμοί (ισότητες) του προβλήματος, και, lb & ub είναι τα κάτω και άνω όρια των μεταβλητών x. Όσα δεδομένα δεν υφίστανται δίνονται σαν κενοί πίνακες [], ή παραλείπονται αν είναι στο τέλος. Το ΠΓΠ που λύσαμε μετασχηματίζεται ελαφρά, για να δοθεί προς λύση στην linprog:

ΑΣ: $\max P = 5x_1 + 2x_2 + 3x_3$ (ή ισοδύναμα $\min f(x) = -5x_1 - 2x_2 - 3x_3$)

ΥΠΠ: $x_1 + 2x_2 + 3x_3 \leq 9$
 $(A^*x \leq b) \quad 3x_1 + 2x_2 + 2x_3 \leq 5 \quad \&$
 $x_1, x_2, x_3 \geq 0 \quad (x \geq lb=0)$

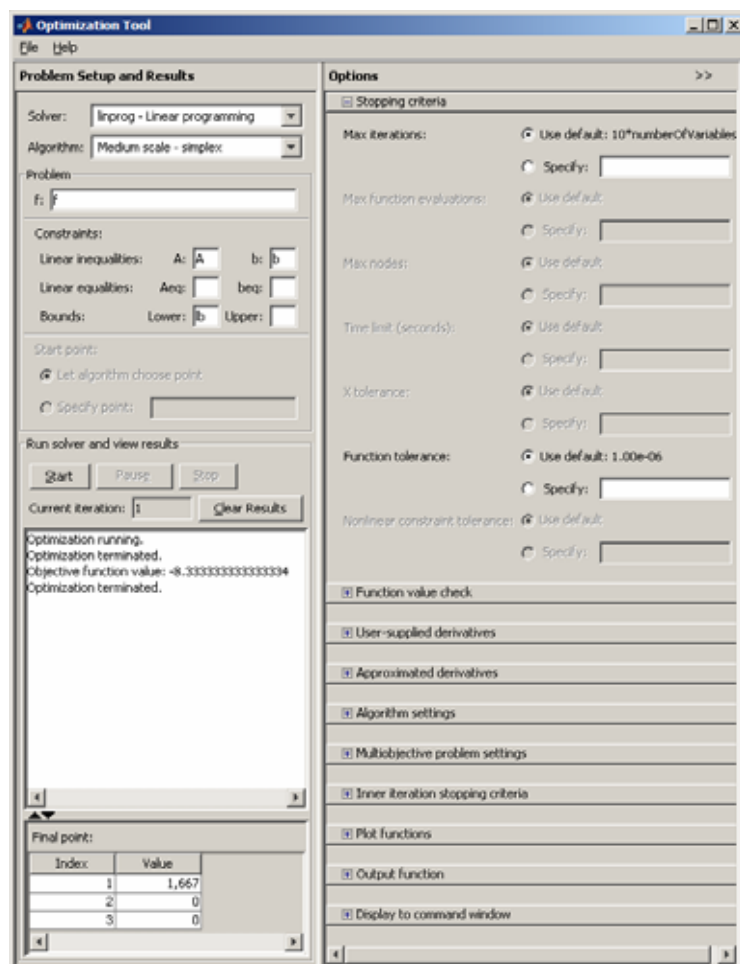
```
>> f = [-5; -2; -3];
>> A = [ 1  2  3 ; 3  2  2];    b = [ 9; 5];
>> lb = zeros(3,1);

>> [x,fval] = linprog(f,A,b,[],[],lb)

Optimization terminated.
x =
    1.6667
    0.0000
    0.0000
fval =
   -8.3333
```

(=5/3)
(=25/3)

3) Για τη **διαδραστική επίλυση** των προβλημάτων βελτιστοποίησης, το Optimization Toolbox διαθέτει το ειδικό εργαλείο **Optimization Tool**. Στο εργαλείο αυτό επιλέγουμε τη μέθοδο επίλυσης (Solver) και τον αλγόριθμο (Algorithm), και στη συνέχεια δίνουμε τα ονόματα των πινάκων που περιέχουν τα στοιχεία του προβλήματος. Με το πλήκτρο Start εκτελείται ο αλγόριθμος και τα αποτελέσματα παρουσιάζονται στα δυο παράθυρα που ακολουθούν.



Όπως είναι αναμενόμενο, τα αποτελέσματα είναι τα ίδια με τις προηγούμενες μεθόδους, αλλά αλλάζει ο τρόπος χρήσης και παρουσίασης.