

Διοίκηση Παραγωγής & Συστημάτων Υπηρεσιών

1η Εργαστηριακή Άσκηση

Γραμμικός Προγραμματισμός



2022 - 2023

Αγγλογάλλος Αναστάσιος

031 18641

Μια κατασκευαστική επιχείρηση θέλει να ελαχιστοποιήσει το συνολικό κόστος (πρώτες ύλες, κόστος εργασίας και αποθήκευσης) του επόμενου χρόνου. Στον παρακάτω πίνακα απεικονίζονται το κόστος των πρώτων υλών μιας μονάδας τελικού προϊόντος, η ζήτηση του τελικού προϊόντος και οι διαθέσιμες ώρες εργασίας για κάθε μήνα. Το κόστος εργασίας είναι 12€ ανά ώρα. Οι εργαζόμενοι πληρώνονται για τις ώρες που εργάστηκαν. Κάθε μονάδα τελικού προϊόντος χρειάζεται 20 λεπτά εργασίας.

Μήνας	1	2	3	4	5	6	7	8	9	10	11	12
Κόστος π.υ. ανά μονάδα τελικού προϊόντος (€)	11	10	13	9	8	7	10	12	12	10	8	9
Ζήτηση (μονάδες)	150	200	100	300	200	400	300	250	150	200	300	350
Διαθέσιμες ώρες εργασίας (ώρες)	250	250	200	150	200	200	150	200	250	150	150	200

Το κόστος αποθήκευσης είναι ίσο με 3€ για κάθε μονάδα που είναι αποθηκευμένη στο τέλος του μήνα. Οποιαδήποτε μονάδα παράγεται σε ένα δεδομένο μήνα μπορεί να χρησιμοποιηθεί για την κάλυψη της ζήτησης του ίδιου μήνα ή να αποθηκευτεί για να καλύψει τη ζήτηση των επόμενων μηνών.

Στις αρχές του πρώτου μήνα η επιχείρηση έχει 100 μονάδες τελικού προϊόντος σε απόθεμα. Σημειώνεται ότι δεν υπάρχει κάποια ελάχιστη απαίτηση αποθεμάτων ανά μήνα.

1. Καταstrώστε και αναλύστε το πρόγραμμα παραγωγής που ελαχιστοποιεί το συνολικό κόστος. Θα πρέπει να χρησιμοποιήσετε υπολογιστικό εργαλείο γραμμικού προγραμματισμού

Διατύπωση Προβλήματος

- Μεταβλητή Απόφασης : x_i , ποσότητα παραγωγής προϊόντος το μήνα i (kg)
- Κριτήριο Απόφασης : Μεγιστοποίηση Κέρδους - Ελαχιστοποίηση Κόστους (€)

Δεδομένα

Αρχικό Απόθεμα $s_0 = 100$ προϊόντα

Περιορισμοί

Περιορισμοί Δυναμικότητας Παραγωγής

Παράγουμε έως $x_{i\max}$ προϊόντα

1ος Μήνας

Περιορισμός :

$$x_1 \leq 750$$

2ος Μήνας

Περιορισμός :

$$x_2 \leq 750$$

3ος Μήνας

Περιορισμός :

$$x_3 \leq 600$$

4ος Μήνας

Περιορισμός :

$$x_4 \leq 450$$

5ος Μήνας

Περιορισμός :

$$x_5 \leq 600$$

6ος Μήνας

Περιορισμός :

$$x_6 \leq 600$$

7ος Μήνας

Περιορισμός :

$$x_7 \leq 450$$

8ος Μήνας

Περιορισμός :

$$x_8 \leq 600$$

9ος Μήνας

Περιορισμός :

$$x_9 \leq 750$$

10ος Μήνας

Περιορισμός :

$$x_{10} \leq 450$$

11ος Μήνας

Περιορισμός :

$$x_{11} \leq 450$$

12ος Μήνας

Περιορισμός :

$$x_{12} \leq 600$$

Διαχείριση Αποθήκης

Αποθηκεύουμε τα προϊόντα που προκύπτουν από την διαφορά των διαθέσιμων προϊόντων του μήνα i με τη ζήτηση του μήνα i

$$\text{Διαθέσιμα προϊόντα}(i) = \text{Αποθηκευμένα προϊόντα}(i - 1) + \text{Παραγωγή}(i)$$

$$\text{Αποθηκευμένα προϊόντα} = \text{Διαθέσιμα προϊόντα}(i) - \text{Ζήτηση}(i)$$

1ος Μήνας

Αποθηκευμένα Προϊόντα

$$s_1 = s_0 + x_1 - 150$$

Περιορισμοί :

$$s_1 \geq 0$$

2ος Μήνας

Αποθηκευμένα Προϊόντα

$$s_2 = s_1 + x_2 - 200$$

Περιορισμοί :

$$s_2 \geq 0$$

3ος Μήνας

Αποθηκευμένα Προϊόντα

$$s_3 = s_2 + x_3 - 100$$

Περιορισμοί :

$$s_3 \geq 0$$

4ος Μήνας

Αποθηκευμένα Προϊόντα

$$s_4 = s_3 + x_4 - 300$$

Περιορισμοί :

$$s_4 \geq 0$$

5ος Μήνας

Αποθηκευμένα Προϊόντα

$$s_5 = s_4 + x_5 - 200$$

Περιορισμοί :

$$s_5 \geq 0$$

6ος Μήνας

Αποθηκευμένα Προϊόντα

$$s_6 = s_5 + x_6 - 400$$

Περιορισμοί :

$$s_6 \geq 0$$

7ος Μήνας

Αποθηκευμένα Προϊόντα

$$s_7 = s_6 + x_7 - 300$$

Περιορισμοί :

$$s_7 \geq 0$$

8ος Μήνας

Αποθηκευμένα Προϊόντα

$$s_8 = s_7 + x_8 - 250$$

Περιορισμοί :

$$s_8 \geq 0$$

8ος Μήνας

Αποθηκευμένα Προϊόντα

$$s_8 = s_7 + x_8 - 250$$

Περιορισμοί :

$$s_8 \geq 0$$

9ος Μήνας

Αποθηκευμένα Προϊόντα

$$s_9 = s_8 + x_9 - 150$$

Περιορισμοί :

$$s_9 \geq 0$$

10ος Μήνας

Αποθηκευμένα Προϊόντα

$$s_{10} = s_9 + x_{10} - 200$$

Περιορισμοί :

$$s_{10} \geq 0$$

11ος Μήνας

Αποθηκευμένα Προϊόντα

$$s_{11} = s_{10} + x_{11} - 300$$

Περιορισμοί :

$$s_{11} \geq 0$$

12ος Μήνας

Αποθηκευμένα Προϊόντα

$$s_{12} = s_{11} + x_{12} - 350$$

Περιορισμοί :

$$s_{12} \geq 0$$

Αντικειμενική Συνάρτηση

min: $\sum (\text{Κόστος Πρώτος υλών}(i) + \text{Εργατικό Κόστος}) \cdot x_i + \text{Κόστος Αποθήκευσης} \cdot \text{Αποθηκευμένα Προϊόντα}(i)$

min:

$$15 \cdot x_1 + 14 \cdot x_2 + 17 \cdot x_3 + 13 \cdot x_4 + 12 \cdot x_5 + 11 \cdot x_6 + 14 \cdot x_7 + 16 \cdot x_8 + 16 \cdot x_9 + 14 \cdot x_{10} + 12 \cdot x_{11} + 13 \cdot x_{12} + 3 \cdot s_1 + 3 \cdot s_2 + 3 \cdot s_3 + 3 \cdot s_4 + 3 \cdot s_5 + 3 \cdot s_6 + 3 \cdot s_7 + 3 \cdot s_8 + 3 \cdot s_9 + 3 \cdot s_{10} + 3 \cdot s_{11} + 3 \cdot s_{12}$$

Κώδικας :

```

/* Objective function */
min: 15*x1 + 14*x2 + 17*x3 + 13*x4 + 12*x5 + 11*x6 + 14*x7 + 16*x8 + 16*x9 + 14*x10 + 12*x11 + 13*x12
    + 3*s1 + 3*s2 + 3*s3 + 3*s4 + 3*s5 + 3*s6 + 3*s7 + 3*s8 + 3*s9 + 3*s10 + 3*s11 + 3*s12;

/* Initial Storage */
s0 = 100;

/* Variable bounds */

/* Production Bounds */
x1 <= 750;
x2 <= 750;
x3 <= 600;
x4 <= 450;
x5 <= 600;
x6 <= 600;
x7 <= 450;
x8 <= 600;
x9 <= 750;
x10 <= 450;
x11 <= 450;
x12 <= 600;

/* Type Declarations */
integer x1; integer x2;
integer x3; integer x4;
integer x5; integer x6;
integer x7; integer x8;
integer x9; integer x10;
integer x11; integer x12;

integer s1; integer s2;
integer s3; integer s4;
integer s5; integer s6;
integer s7; integer s8;
integer s9; integer s10;
integer s11; integer s12;

/* Demand & Storage Bounds */
s1 = s0 + x1 - 150; s1 >= 0;

s2 = s1 + x2 - 200; s2 >= 0;

s3 = s2 + x3 - 100; s3 >= 0;

s4 = s3 + x4 - 300; s4 >= 0;

s5 = s4 + x5 - 200; s5 >= 0;

s6 = s5 + x6 - 400; s6 >= 0;

s7 = s6 + x7 - 300; s7 >= 0;

s8 = s7 + x8 - 250; s8 >= 0;

s9 = s8 + x9 - 150; s9 >= 0;

s10 = s9 + x10 - 200; s10 >= 0;

s11 = s10 + x11 - 300; s11 >= 0;

s12 = s11 + x12 - 350; s12 >= 0;

```

Result :

Variables	MILP Feasible	result ▲
	37500	37500
x3	0	0
s1	0	0
s3	0	0
s4	0	0
s5	0	0
s7	0	0
s8	0	0
s9	0	0
s10	0	0
s11	0	0
s12	0	0
x1	50	50
x7	100	100
s2	100	100
s0	100	100
x9	150	150
x5	200	200
x10	200	200
s6	200	200
x8	250	250
x2	300	300
x4	300	300
x11	300	300
x12	350	350
x6	600	600

Αποτελέσματα

Παρατηρούμε ότι το πρόγραμμα παραγωγής που ελαχιστοποιεί το συνολικό κόστος είναι το εξής :

Προϊόντα	Μήνας (i)											
	1ος	2ος	3ος	4ος	5ο	6ος	7ος	8ος	9ος	10ος	11ος	12ος
Παραγωγή x_i	50	300	0	300	200	600	100	250	150	200	300	350
Αποθήκευση s_i	0	100	0	0	0	200	0	0	0	0	0	0

Συνολικό Κόστος = **37.500** ευρώ

2. Τροποποιήστε το μοντέλο της προηγούμενης ενότητας, εάν υποθέσουμε ότι η επιχείρηση επιβαρύνεται και με ένα σταθερό κόστος 1.000€ για κάθε μήνα που υπάρχει παραγωγή. Αυτό το κόστος, δηλαδή, λαμβάνεται υπόψη μόνο αν η παραγωγή του εξεταζόμενου μήνα είναι διαφορετική του μηδενός.

Θα πρέπει να σκεφτούμε ένα τρόπο να ελέγξουμε εάν η παραγωγή είναι διαφορετική του μηδενός. Δυστυχώς, το πρόγραμμα Lpsolve δεν είναι γλώσσα προγραμματισμού οπότε δεν έχει If - Statements. Όμως με χρήση μιας βοηθητικής μεταβλητής b_i κατασκευάζουμε την συγκεκριμένη διάταξη.

Συγκεκριμένα ελέγχουμε εάν το x_i είναι μικρότερο του $M \cdot b_i$,
όπου M ένας αριθμός μεγαλύτερος από κάθε x_i
Έπειτα ελέγχουμε εάν είναι μεγαλύτερο $2 \cdot b_i - 1$,
Με αυτό το τρόπο κατασκευάσαμε το δικό μας λειτουργικό if-statement

Προσθέτουμε την επιπλέον συνθήκη στον κώδικά μας

```
/* If Statements */

b1 <= 1; x1 <= 1000*b1; x1 >= 2*b1 - 1;
b2 <= 1; x2 <= 1000*b2; x2 >= 2*b2 - 1;
b3 <= 1; x3 <= 1000*b3; x3 >= 2*b3 - 1;
b4 <= 1; x4 <= 1000*b4; x4 >= 2*b4 - 1;
b5 <= 1; x5 <= 1000*b5; x5 >= 2*b5 - 1;
b6 <= 1; x6 <= 1000*b6; x6 >= 2*b6 - 1;
b7 <= 1; x7 <= 1000*b7; x7 >= 2*b7 - 1;
b8 <= 1; x8 <= 1000*b8; x8 >= 2*b8 - 1;
b9 <= 1; x9 <= 1000*b9; x9 >= 2*b9 - 1;
b10 <= 1; x10 <= 1000*b10; x10 >= 2*b10 - 1;
b11 <= 1; x11 <= 1000*b11; x11 >= 2*b11 - 1;
b12 <= 1; x12 <= 1000*b12; x12 >= 2*b12 - 1;

integer b1; integer b2;
integer b3; integer b4;
integer b5; integer b6;
integer b7; integer b8;
integer b9; integer b10;
integer b11; integer b12;
```

Προκύπτει η παρακάτω αντικειμενική συνάρτηση :

Εάν παραγωγή(i) $\neq 0$ τότε $b_i = 1$

```
/* Objective function */
min: 15*x1 + 14*x2 + 17*x3 + 13*x4 + 12*x5 + 11*x6 + 14*x7 + 16*x8 + 16*x9 + 14*x10 + 12*x11 + 13*x12
+ 3*s1 + 3*s2 + 3*s3 + 3*s4 + 3*s5 + 3*s6 + 3*s7 + 3*s8 + 3*s9 + 3*s10 + 3*s11 + 3*s12
+ 1000*b1 + 1000*b2 + 1000*b3 + 1000*b4 + 1000*b5 + 1000*b6 + 1000*b7 + 1000*b8 + 1000*b9 + 1000*b10 + 1000*b11 + 1000*b12;
```

Έξοδος :

Variables	MILP Feasible	MILP Better	MILP Better	MILP Better	MILP Better	result
	48500	47950	47750	47550	47350	47350
x1	50	50	50	50	50	50
x2	300	300	300	300	300	300
x3	0	0	0	0	0	0
x4	300	300	300	300	300	300
x5	200	200	200	250	300	300
x6	600	600	600	600	600	600
x7	100	100	350	450	0	0
x8	250	400	0	0	400	400
x9	150	0	150	0	0	0
x10	200	200	200	200	200	200
x11	300	300	300	300	300	300
x12	350	350	350	350	350	350
s1	0	0	0	0	0	0
s2	100	100	100	100	100	100
s3	0	0	0	0	0	0
s4	0	0	0	0	0	0
s5	0	0	0	50	100	100
s6	200	200	200	250	300	300
s7	0	0	250	400	0	0
s8	0	150	0	150	150	150
s9	0	0	0	0	0	0
s10	0	0	0	0	0	0
s11	0	0	0	0	0	0
s12	0	0	0	0	0	0
b1	1	1	1	1	1	1
b2	1	1	1	1	1	1
b3	0	0	0	0	0	0
b4	1	1	1	1	1	1
b5	1	1	1	1	1	1
b6	1	1	1	1	1	1
b7	1	1	1	1	0	0
b8	1	1	0	0	1	1
b9	1	0	1	0	0	0
b10	1	1	1	1	1	1
b11	1	1	1	1	1	1
b12	1	1	1	1	1	1
x0	100	100	100	100	100	100

Αποτελέσματα

Παρατηρούμε ότι το πρόγραμμα παραγωγής που ελαχιστοποιεί το συνολικό κόστος είναι το εξής :

Προϊόντα	Μήνας (i)											
	1ος	2ος	3ος	4ος	5ο	6ος	7ος	8ος	9ος	10ος	11ος	12ος
Παραγωγή x_i	50	300	0	300	300	600	0	400	0	200	300	350
Αποθήκευση s_i	0	100	0	0	100	300	0	150	0	0	0	0

Συνολικο Κόστος = **47.350** ευρώ