ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ/ΚΩΝ & ΜΗΧ/ΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Μάθημα: "ΔΙΟΙΚΗΣΗ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΥΠΗΡΕΣΙΩΝ "

Λουκάς Άγγελος 03119877

1η εργασία

Δεδομένα

Di : ζήτηση το μήνα t, όπου t = 1, 2, …, 12

Κόστος παραγωγής 1 τεμαχίου : 50€

Κόστος μηνιαίας διατήρησης αποθέματος 1 τεμαχίου : 0.8€

Κόστος αύξησης ρυθμού παραγωγής από τον μήνα i-1 στον μήνα i : 1.3€

Κόστος μείωσης ρυθμού παραγωγής από τον μήνα i-1 στον μήνα i : 2€

Παραγωγική δυναμικότητα: 1800 τεμάχια

Κόστος υπερωριών: 2.5€

Κόστος υποαπασχόλησης: 4€

Μεταβλητές απόφασης

P_i : αριθμός τεμαχίων που παράγονται το μήνα i

A_i : διαθέσιμο απόθεμα στο τέλος του μήνα i

MAX_P1_i : max($P_i - P_{i-1}$, 0)

MAX_P2_i : max($P_{i-1} - P_i$, 0)

MAX_SP1_i : max(P_i – 1800, 0)

MAX_SP1_i : max(1800 – P_i , 0)

Αρχικές συνθήκες

$P_0 = 1600$

$A_0 = 700$

Αντικειμενική συνάρτηση προς ελαχιστοποίηση

$$minZ = \sum_{i=1}^{12} 50 \cdot P_i + 0.8 \cdot A_i + 1.3 \cdot max(P_i - P_{i-1}, 0) + 2 \cdot max(P_{i-1} - P_i, 0) + 2.5 \cdot max(P_i - 1800, 0) + 4 \cdot max(1800 - P_i, 0)$$

Περιορισμοί

Η παραγωγή του μήνα i και το απόθεμα του μήνα i-1 πρέπει να υπερβαίνει την ζήτηση του μήνα i.

$$A_{i-1} + P_i \geq D_i$$

Το απόθεμα του μήνα i είναι ίσο με το άθροισμα του αποθέματος του μήνα i-1 και της παραγωγής του μήνα i πλιν την ζήτηση του μήνα i.

$$A_i = A_{i-1} + P_i - D_i \; MAX\_P1\_i \geq 0$$

Για τον υπολογισμό των **MAX_P1_i , MAX_P2_i , MAX_SP1_i , MAX_SP2_i** υλοποιήθηκαν με τις παρακάτω συναρτήσεις για τον κάθε μήνα i.

- $MAX\_P1\_i \geq 0$
- $MAX\_P1\_i \geq P_i - P_{i-1}$
- $MAX\_P2\_i \geq 0$
- $MAX\_P2\_i \geq P_{i-1} - P_i$
- $MAX\_SP1\_i \geq 0$
- $MAX\_SP1\_i \geq P_i - 1800$
- $MAX\_SP2\_i \geq 0$
- $MAX\_SP2\_i \geq 1800 - P_i$

Επίλυση του προβλήματος

Η επίλυση του προβλήματος έγινε με την βιβλιοθήκη pulp της python.

```python
import pulp

problem = pulp.LpProblem("LP_Problem", pulp.LpMinimize)

Pv=50
Ch=0.8
ARP=1.3
MRP=2
StandardP=1800
OverP=2.5
UnderP=4
D=[2100, 1900, 1600, 1500, 1550, 1400, 1250, 1700, 2200, 2300, 2100, 1950]

P = [pulp.LpVariable(f"P{i}", lowBound=0) for i in range(0, 13)]
A = [pulp.LpVariable(f"A{i}", lowBound=0) for i in range(0, 13)]
MAX_P1 = [pulp.LpVariable(f"MAX_P1_{i}", lowBound=0) for i in range(0, 13)]
MAX_P2 = [pulp.LpVariable(f"MAX_P2_{i}", lowBound=0) for i in range(0, 13)]
MAX_SP1 = [pulp.LpVariable(f"MAX_SP1_{i}", lowBound=0) for i in range(0, 13)]
MAX_SP2 = [pulp.LpVariable(f"MAX_SP2_{i}", lowBound=0) for i in range(0, 13)]

def custom_function(A,P,MAX_P1,MAX_P2,MAX_SP1,MAX_SP2):
    _return=A[0]-A[0]
    for i in range(1,13):
        _return+= Pv*P[i]+Ch*A[i]+ARP*MAX_P1[i]+MRP*MAX_P2[i]+OverP*MAX_SP1[i]+UnderP*MAX_SP2[i]
    return _return

objective = custom_function(A,P,MAX_P1,MAX_P2,MAX_SP1,MAX_SP2)

problem += objective, "Objective"

constraints=[
    A[0]==700,
    P[0]==1600
]
for i in range(1,13):
    constraints.append(A[i-1]+P[i] >= D[i-1])
    constraints.append(A[i] == A[i-1] + P[i] - D[i-1])
    constraints.append(MAX_P1[i]>=0)
    constraints.append(MAX_P2[i]>=0)
    constraints.append(MAX_SP1[i]>=0)
    constraints.append(MAX_SP2[i]>=0)
    constraints.append(MAX_P1[i]>= P[i]-P[i-1])
    constraints.append(MAX_P2[i]>= -P[i]+P[i-1])
    constraints.append(MAX_SP1[i]>= P[i]-StandardP)
    constraints.append(MAX_SP2[i]>= -P[i]+StandardP)

for i, constraint in enumerate(constraints):
    problem += constraint, f"Constraint{i+1}"
problem.solve()
for var in problem.variables():
    print(f"{var.name}: {var.varValue}")
print(f"Optimal Objective Value: {pulp.value(problem.objective)}")
```

Έξοδος

	0	1	2	3	4	5	6	7	8	9	10	11	12
D		2100	1900	1600	1500	1550	1400	1250	1700	2200	2300	2100	1950
P	1600	1650	1650	1650	1650	1650	1800	1800	1800	1800	1800	1800	1800
A	700	250	0	50	200	300	700	1250	1350	950	450	150	0
MAX_P1		50	0	0	0	0	150	0	0	0	0	0	0
MAX_P2		0	0	0	0	0	0	0	0	0	0	0	0
MAX_SP1		0	0	0	0	0	0	0	0	0	0	0	0
MAX_SP2		150	150	150	150	150	0	0	0	0	0	0	0

Optimal Objective Value: 1050280