

16-3-2026

$$\begin{aligned} T &= 4 \\ d &= (5, 8, 4, 6) \\ c &= (2, 1, 4, 2) \\ h &= 3 \\ I_1 &= 0 \end{aligned}$$

## Model Class

user

- Sets ← ονταδα δεκτιων.
- Parameters ← τιμες ανεξαρτητων
- Variables ← μεταβλ. ανδραση
- Constraints ← περιορισμοι
- Objective ← αυτικ. αναζητηαν.

nx. ① inventory model  
μεταφορας

$$d_t : t = 1, \dots, T$$

$$x_t : t = 1, \dots, T$$

$$I_t : t = 2, \dots, T+1$$

## ② Network model

Set of nodes =  $N$   
" " edges =  $E \subset N \times N$

$c_{ij}$  : costs across  $(i, j) : (i, j) \in E$

$a_i$  : supply  $i$  ,  $i \in N$

$N$  = set of nodes  $i \in N$

$E$  : " " edges  $e \in E$

sets

# Ανοδέματα

|                     | Type in<br>model class | Μορφή<br>δεδομένων  |
|---------------------|------------------------|---------------------|
| $T :$               | parameter              | αριθμός             |
| $1, \dots, T :$     | set "T"                | range, set $\{ \}$  |
| $[d_t] : t \in "T"$ | parameters             | dictionary          |
| $[c_t] : t \in "T"$ | parameters             | "                   |
| $x_t : t \in "T"$   | variables              | "                   |
| objective           | objective              | <u>function</u>     |
| constraints         | constraints            | set of<br>functions |

python  
script

①

Εισαγωγή δεδομένων  
από αρχείο/κείμενο ή  
από αρχείο csv, xls...  
στην κατάλληλη μορφή  
για εισαγωγή στο model

②

Όποιοι model object  
add-set ...  
add-parameter ...

# dictionary in Python

$a = \{ 1:5, 2:4, 3:7, 4:8 \}$   
          ↑    ↑  
          name value

$a[1] \rightarrow 5$

$a[4] \rightarrow 8$

$b = \{ "Ath": 5000, "The": 7000, "Car": 3000 \}$

$b["Ath"] \rightarrow 5000$  :-

---

## Ορισμός Παράδειγμων

$T = 4$  ( $d_1 = 5 \dots \dots \dots$   $d_4 = 6$ )

$d = \{ 1:5, 2:8, 3:4, 4:6 \}$

$c = \{ 1:2, 2:1, 3:4, 4:2 \}$

$h = 3$

$I1 = 2$

---

m = Model()

m.add-set("T", range(1, T+1))

↑  
name (1, 2, ..., T)

m.add-set("T2", range(2, T+2))

m.add-parameter("d", d, index="T")

m.add-parameter("c", c, index="T")

m.add-parameter("h", h)

[Eval. add-par("h", 3)]

→ 2005  
→ 2020

m.add-parameter("I1", I1)

oprotos  
of index given

m.add-variable("x", index="T") → x<sub>1</sub>, ..., x<sub>T</sub>

m.add-variable("I", index="T2") I<sub>2</sub>, ..., I<sub>T+1</sub>

oprotos  
for each function

Objective : function definition

```
def obj(gmodel):
    return sum(gmodel.c[t] * gmodel.x[t]
               for t in gmodel.T)
```

```
+ sum(gmodel.h * gmodel.I[t]
      for t in gmodel.T2)
```

oprotos  
py than

$m.set\_objective(obj, sense = "min")$   
 opções  
 no método

$$\left[ \sum_{t=1}^T c_t x_t + \sum_{t=2}^{T+1} h I_t \right]$$

### Constraints (balance)

$$I(t+1) = I(t) + x(t) - d(t) \quad \underline{t=1, \dots, T}$$

$$t=1 : I(2) = \underbrace{I(1)}_{\text{napes.}} + x(1) - d(1)$$

$$t=2 : I(3) = \underbrace{I(2)}_{\text{fuzo. p. 2. n. 2. i}} + x(2) - d(2)$$

def balance\_constr(gmodel, t):

if t == 1:

return gmodel.I[2] == gmodel.I[1] +  
 + gmodel.x[1] -  
 - gmodel.d[1]

else:

return gmodel.I[t+1] == gmodel.I[t]

$$+ gmodel.x[t]$$
$$- gmodel.d[t]$$

`m.add_constraint("balance_const", balance_const,`  
↑  
name  
↑  
function  
`index = "T")`

---

Avion

`m.solve()`

`m.display()`

`m.slacks()`