

## Gestione delle scorte

Siete direttori del magazzino di un grossista che rifornisce negozi comprando la merce dalle fabbriche. Consideriamo una singola merce.

Nei prossimi 5 mesi (il vostro orizzonte temporale) la quantità complessiva che i negozi richiedono ad ogni mese è la seguente:

	Mese 1	Mese 2	Mese 3	Mese 4	Mese 5
Domanda	12	4	18	17	7

La merce comprata alla fabbrica ha un costo che varia mese per mese. Nei prossimi 5 mesi i costi per unità di prodotto sono:

	Mese 1	Mese 2	Mese 3	Mese 4	Mese 5
Cmerce	8	12	7	11	10

In più, ogni volta che si acquista della merce dalla fabbrica si paga una quantità fissa per le spese di trasporto. Nei prossimi 5 mesi è costante e vale:

	Mese 1	Mese 2	Mese 3	Mese 4	Mese 5
Cspedizione	10	10	10	10	10

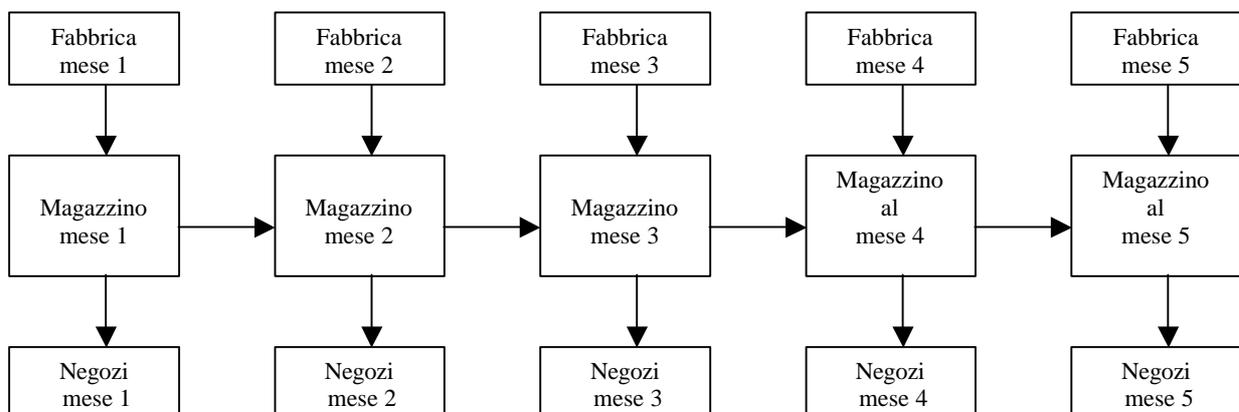
Inoltre, la merce può non essere venduta subito, ma conservata in magazzino. Il magazzino ha un costo per ogni unità di prodotto conservata per ogni mese. Nei prossimi 5 mesi i costi per unità di prodotto sono costanti e valgono:

	Mese 1	Mese 2	Mese 3	Mese 4	Mese 5
Cmagazzino	5	5	5	5	5

Vogliamo ovviamente gestire in magazzino in modo da minimizzare le spese e con il vincolo di soddisfare tutte le richieste dei negozi.

Detto ciò, risulta abbastanza chiaro che questo problema va formulato come un problema di ottimizzazione. Risulta anche chiaro cosa devono esprimere i vincolo e l'obiettivo.

La situazione è la seguente: ad ogni mese possiamo comprare un quantità di merce, che possiamo in parte vendere e in parte conservare in magazzino. La merce che vendiamo ai negozi può essere stata comprata in quello stesso mese oppure comprata nei mesi precedenti e tenuta in magazzino.



## Il modello di programmazione lineare intera

Le grandezze che vogliamo trovare sono le quantità di merce da comprare ad ogni mese e la quantità di merce da conservare in magazzino. Usiamo allora delle variabili

$$\begin{aligned}c_i &= \text{merce comprata al mese } i \\s_i &= \text{merce mantenuta in magazzino al mese } i\end{aligned}$$

Inoltre ci conviene usare delle variabili che possono assumere soltanto due valori per indicare se in quel mese acquistiamo della merce e quindi paghiamo le spese di spedizione.

$$v_i = \begin{cases} 0 & \text{se nel mese } i \text{ non ci sono spese di spedizione} \\ 1 & \text{se nel mese } i \text{ ci sono spese di spedizione} \end{cases}$$

Naturalmente queste variabili non sono tutte indipendenti.

Per esprimere il fatto che al mese 1 la quantità di merce che rimane in magazzino è quella comprata meno quella richiesta scriviamo

$$s_1 = c_1 - \text{Domanda}(1)$$

Per esprimere il fatto che ad ogni altro mese  $i$  la quantità di merce che rimane in magazzino è quella comprata più quella che già era in magazzino meno quella richiesta scriviamo

$$s_i = s_{i-1} + c_i - \text{Domanda}(i)$$

Per esprimere il fatto che ogni volta che si compra della merce bisogna pagare le spese di spedizione scriviamo

$$M v_i \geq c_i$$

L'obiettivo è minimizzare la somma delle spese, e quindi .

$$\min. \sum_{i=1}^5 C_{\text{merce}_i} c_i + \sum_{i=1}^5 C_{\text{spedizione}_i} v_i + \sum_{i=1}^5 C_{\text{magazzino}_i} s_i$$

## Soluzione del modello scelto attraverso AMPL

Il modello che rappresenta il problema descritto sopra è:

```
set O := 1..5;          # orizzonte temporale

param Domanda{O} ;     # quantità di merce richiesta
param Cmerce{O};      # costo della merce
param Cspedizione{O}; # costo di spedizione
param Cmagazzino{O};  # costo di magazzino
param M;              # valore per il vincolo sui costi di spedizione

var c{i in O} >= 0;    # merce comprata nel mese
var s{i in O} >= 0;    # merce che rimane in magazzino nel mese
var v{i in O} binary; # 1 se nel mese ho spese di spedizione, 0 altrimenti

minimize costi: sum{i in O} Cmerce[i] * c[i] +
                sum{i in O} Cspedizione[i] * v[i] +
                sum{i in O} Cmagazzino[i] * s[i];

s.t. merce_mese_1: s[1] = c[1] - Domanda[1];
s.t. merce_mese_i{i in O: i>1}: s[i] = s[i-1] + c[i] - Domanda[i];
s.t. spedizione{i in O}: M * v[i] >= c[i];
```

Il file di dati corrispondente è:

```
param Domanda :=
  1  12
  2   4
  3  18
  4  17
  5   7;

param Cmerce :=
  1   8
  2  12
  3   7
  4  11
  5  10;

param Cspedizione :=
  1  10
  2  10
  3  10
  4  10
  5  10;

param Cmagazzino :=
  1   5
  2   5
  3   5
  4   5
  5   5;

param M := 60;
```

## Soluzione

La soluzione del nostro problema è quindi:

OBJECTIVES:

costi = 571

VARIABLES:

c [\*] :=

1 16

2 0

3 18

4 17

5 7

;

s [\*] :=

1 4

2 0

3 0

4 0

5 0

;

v [\*] :=

1 1

2 0

3 1

4 1

5 1

;