

AMPL Swiss Knife

Gabriel Rovesti

Testo esercizio

6. Si vuole risolvere con **AMPL** un problema di trasporto di alberi da un insieme di origini I a un insieme di destinazioni J . Ciascuna origine i mette a disposizione O_i alberi e ciascuna destinazione richiede D_j alberi. Il costo unitario di trasporto da i a j è C_{ij} e si ha un costo fisso F_i per l'organizzazione dei trasporti da ciascuna origine i . Non è inoltre possibile organizzare il trasporto in più di N origini. Il modello per la minimizzazione dei costi è riportato affianco e utilizza le variabili x_{ij} per indicare il numero di alberi trasportati da i a j , e y_i che vale 1 se si organizza il trasporto da i , 0 altrimenti.

$$\min \sum_{i \in I, j \in J} C_{ij} x_{ij} - \sum_{i \in I} F_i y_i$$

$$\text{s.t. } \sum_{i \in I} x_{ij} \geq D_j \quad , \quad \forall j \in J$$

$$\sum_{j \in J} x_{ij} \leq O_i y_i \quad , \quad \forall i \in I$$

$$\sum_{i \in I} y_i \leq N$$

$$x_{ij} \in \mathbb{Z}_+, \quad y_i \in \{0,1\}, \quad \forall i \in I, \quad j \in J$$

- Si traduca nel linguaggio **AMPL** il modello proposto (**file .mod**).
- Si produca il **file .dat** per l'istanza con origini Croazia, Svezia, Gran Bretagna e Canada (disponibilità di 1000, 2000, 3000 e 4000 alberi rispettivamente), destinazioni Italia, Francia e Germania (con richieste di 5000, 3000 e 2000 rispettivamente), $N = 3$, costi fissi F_i di 1000 euro per tutte le origini, e costi di trasporto verso Italia, Francia e Germania (nell'ordine) pari a: dalla Croazia 10, 20 e 30 euro; dalla Svezia 40, 50 e 60 euro; dalla Gran Bretagna 70, 80 e 90 euro; dal Canada 100, 110 e 120 euro.
- Si scriva uno script di **AMPL** (**file .run**) che risolve l'istanza specificata e visualizza il valore della funzione obiettivo e delle variabili per una soluzione ottima.

Figure 1: Testo esercizio AMPL

```
# Insiemi

set I;
set J;

# Parametri

param O{I}; #origine
param D{J}; #destinazione
param C{I, J}; #costo variabile
param F{I}; #costo fisso

# Variabili

var x{I, J} >=0 integer;
var y{I} binary;

# Funzione obiettivo

minimize fo: sum{i in I, j in J} C[i,j] * x[i,j]
- sum{i in I} F[i] * y[i];
```

```

s.t. d{j in J}: sum{i in I} x[i,j] >= D[j]; #destinazioni

s.t. o{i in I}: sum{j in J} x[i,j] <= O[i] * y[i]; #origini

s.t. n: sum{i in I} y[i] <= N;

# Nel file.dat

set I := Croazia Svezia GranBretagna Canada;
set J := Italia Francia Germania;

# Disponibilità con costi fissi
param:    F:  0 :=
Croazia   1000 1000
Svezia    1000 2000
GranBretagna 1000 3000
Canada;   1000 4000;

param D := Italia 5000 Francia 3000 Germania 2000;

param N := 3;

param C:  Italia  Francia  Germania
Croazia   10   20   30
Svezia    40   50   60
GranBretagna 70   80   90
Canada    100  110  120;

# File .run

reset;
model ampl.mod;
data file.dat;
option solver cplexamp;
solve;
display fo, x, y;

```