

Elementi base di AMPL

Runaway Spa

Problema di Produzione

- ▶ 3 tipi di scarpe da corsa
- ▶ 2 reparti di produzione
- ▶ Ritmi di produzione per ogni tipo di scarpa in ogni reparto:
il tempo è una risorsa limitata in ogni reparto (1000h 850h)
- ▶ La domanda massima di ogni tipo di scarpa è limitata

PROD	Ricavo (euro)	DomMax	Rit R1 (num/h)	Rit R2 (num/h)
Abibas	60	2000	6	7
Naic	75	3000	5	6
Nero_Giardini	85	3000	4	5

Runaway Spa: Formulazione

$$\begin{aligned} \max_{Quantita} \quad & \sum_{i \in PROD} Quantita_i Ricavo_i \\ \text{s.t.} \quad & \sum_{i \in PROD} Quantita_i (1 / Ritmo_{ij}) \leq TempoMax_j \quad j \in \{REPARTI\} \\ & 0 \leq Quantita_i \leq DomMax_i \quad i \in \{PROD\} \end{aligned}$$

N.B. Non utilizzare lettere accentate in AMPL!

Struttura codice AMPL

AMPL richiede tre file:

- ▶ File Mod (*Runaway.mod*) per la dichiarazione
- ▶ File Dat (*Runaway.dat*) per l'assegnazione
- ▶ File Run (*Runaway.run*) per i comandi

Manuale AMPL: <https://dev.ampl.com/en/latest/book.html>

Insiemi

Gli insiemi definiscono gli elementi in base ai quali si indicizzano variabili, parametri e vincoli del modello

È necessario distinguere tra dichiarazione e assegnazione di un insieme

- ▶ dichiarazione (*Runaway.mod*):

```
set PROD;  
set REPARTI;
```

- ▶ assegnazione (*Runaway.dat*):

```
set PROD      := Abibas Naic Nero_Giardini;  
set REPARTI := R1 R2;
```

Insiemi: dichiarazione diverse tipologie

1. insieme non ordinato:

```
set PROD;
```

2. insieme ordinato:

```
set REPARTI ordered;
```

funzione	risultato
<code>first (REPARTI)</code>	primo elemento di REPARTI
<code>last (REPARTI)</code>	ultimo elemento di REPARTI
<code>next (r, REPARTI)</code>	elemento di REPARTI dopo di r
<code>prev (r, REPARTI)</code>	elemento di REPARTI prima di r
<code>ord (r, REPARTI)</code>	posizione di r in REPARTI
<code>member (k, REPARTI)</code>	elemento di REPARTI in k-esima posizione

3. insieme numerico:

```
set ANNI := 2018 .. 2022;
```

gli insiemi numerici non hanno bisogno di un'assegnazione

Parametri

I parametri rappresentano i dati del problema che non vengono modificati dal solutore

► dichiarazione (*Runaway.mod*):

```
set PROD; set REPARTI;
```

```
param Ricavo{PROD};
```

```
param DomMax{PROD};
```

```
param TempoMax{REPARTI};
```

```
param Ritmo{PROD, REPARTI};
```

► uso:

```
...Ricavo[i]...; # i in PROD
```

```
...Ritmo[i,j]...; # i in PROD e j in REPARTI
```

Parametri

► assegnazione (*Runaway.dat*):

```
set PROD:= Abibas Naic Nero_Giardini;  
set REPARTI:= R1 R2;
```

```
param Ricavo:= Abibas 60 Naic 75 Nero_Giardini 85;  
param DomMax:= Abibas 2000 Naic 3000 Nero_Giardini 3000;  
param TempoMax:= R1 1000 R2 850;  
param Ritmo:  
    Abibas      6      7  
    Naic        5      6  
    Nero_Giardini 4      5;
```


Variabili

Le variabili rappresentano le incognite del modello, i cui valori ottimi vengono determinati dal solutore

- dichiarazione (*Runaway.mod*):

```
set PROD; set REPARTI;  
param Ricavo{PROD};  
param DomMax{PROD};  
param TempoMax{REPARTI};  
param Ritmo{PROD, REPARTI};
```

```
var Quantita{i in PROD} >= 0, <= DomMax[i];
```

- uso: come per i parametri
- si possono aggiungere altre restrizioni per le (*Runaway.mod*):

```
var Quantita{i in PROD} >= 0, <= DomMax[i], integer;  
anche binary se le variabili sono {0,1}
```

Funzione obiettivo

La funzione obiettivo definisce un ranking delle soluzioni ammissibili

► dichiarazione:

```
minimize obj: espressione_aritmetica;
```

oppure

```
maximize obj: espressione_aritmetica;
```

Per il nostro problema (*Runaway.mod*):

```
maximize Profitto:  
    sum{i in PROD}Quantita[i]*Ricavo[i];
```

I vincoli del modello definiscono le caratteristiche che una possibile soluzione deve soddisfare

► dichiarazione vincolo:

```
subject to nomevincolo {DIMENSIONE}:  
    espressione_logica;
```

Per il nostro problema (*Runaway.mod*):

```
subject to Tempo{j in REPARTI}:  
    sum{i in PROD}Quantita[i]*(1/Ritmo[i,j]) <= TempoMax[j];
```

Caricamento e soluzione del modello (*Runaway.run*)

► Caricamento del Modello :

Reset:

```
reset;
```

Struttura:

```
model Runaway.mod;
```

Dati:

```
data Runaway.dat;
```

► Soluzione del Modello:

Solutore (*opzionale*):

```
option solver minos;
```

Ottimizzazione:

```
solve;
```

► Visualizzazione risultati:

Visualizza valore ottimo e variabili ottime:

```
display Profitto;    display Quantita;
```

Solutori disponibili in AMPL

Solutore	Problemi Supportati
-----------------	----------------------------

CBC	PL, PLI
-----	---------

HiGHS	PL, PLI, Quadratico Convesso (PNL)
-------	------------------------------------

Ipopt	PNL (Calcolo condizioni necessarie)
-------	-------------------------------------

Bonmin	PNL Misto Intero (Calcolo condizioni necessarie)
--------	--

Couenne	PNL Misto Intero (Calcolo soluzione)
---------	--------------------------------------