

Esercitazione sul linguaggio di modellazione AMPL

Problema di pianificazione della produzione

Una industria manifatturiera produce due prodotti denominati per semplicità Standard e Deluxe. La materia prima necessaria per questa produzione è il ferro, e per effettuare la produzione occorre utilizzare macchinari per la levigatura e macchinari per la pulitura. In particolare, per produrre un Kg di ognuno dei due prodotti, sono necessarie le risorse riportate nella seguente tabella:

	Standard	Deluxe
Ferro	4 Kg	4 Kg
Levigatura	4 ore	2 ore
Pulitura	2 ore	5 ore

Il prodotto Standard viene venduto a 10 Euro al Kg, mentre quello Deluxe viene venduto a 15 Euro al Kg. L'impianto può disporre di 75 Kg di ferro a settimana, e di 80 ore di levigatura e 60 di pulitura a settimana. Occorre pianificare la produzione dei due prodotti in modo da massimizzare i ricavi.

Ciò che occorre decidere è quindi la quantità di prodotto Standard e quella di prodotto Deluxe da produrre ogni settimana. A tal fine occorre modellare il problema come problema di ottimizzazione, e le variabili del modello saranno

x_{st} = quantità in Kg di prodotto Standard da produrre per settimana

x_{dl} = quantità in Kg di prodotto Deluxe da produrre per settimana

La quantità di prodotto può assumere valori non interi, per cui le variabili possono assumere valori reali. Il modello richiesto è allora:

$$\begin{aligned} \max & 10 x_{st} + 15 x_{dl} \\ 4 x_{st} + 4 x_{dl} & \leq 75 \quad (\text{disponibilità ferro}) \\ 4 x_{st} + 2 x_{dl} & \leq 80 \quad (\text{disponibilità levigatura}) \\ 2 x_{st} + 5 x_{dl} & \leq 60 \quad (\text{disponibilità pulitura}) \\ x_{st}, x_{dl} & \geq 0 \quad (\text{nonnegatività}) \end{aligned}$$

Per risolvere questo modello di Programmazione Lineare esistono strumenti software appositi. Anche se quello dell'esempio è un modello molto semplice, ed è quindi risolubile anche con metodi meno sofisticati, i modelli ottenibili nei casi reali sono generalmente di dimensioni molto maggiori, e sono quindi risolubili solo per mezzo di solutori software.

Per utilizzare un solutore automatico occorre esprimere il modello riportato sopra in un linguaggio di modellazione, cioè un linguaggio interpretabile da un solutore software. Un linguaggio molto diffuso è il linguaggio AMPL (Algebraic Mathematical Programming Language)

In AMPL è buono stile utilizzare un file di modello e un file di dati. Il file di modello è un file di testo con estensione `.mod` in cui viene descritta la struttura del modello matematico, ma senza specificare valori numerici. Il file di dati è invece un file di testo con estensione `.dat` in cui sono riportati tutti i valori numerici, o eventuali stringhe, etc. presenti nel modello stesso. Ciò rende possibile modificare i valori numerici o anche le dimensioni del modello senza modificare il file della struttura del modello. Questa indipendenza concettuale tra modello e dati ha diversi vantaggi pratici.

Per esprimere il modello non basta tradurre 1 a 1 il modello scritto sopra con la simbologia matematica in un semplice file di testo, ma occorre prima comunicare al solutore quali sono gli oggetti che troverà in questo modello (variabili, costanti, insiemi, etc.). Occorre cioè dichiarare tutto ciò che poi si utilizzerà per scrivere il modello, come avviene anche in molti altri linguaggi di programmazione. Prima di scrivere il modello bisogna quindi capire e dichiarare quali sono tutti gli oggetti che serviranno per esprimere il modello.

Osservando bene il modello scritto sopra, notiamo che essenzialmente abbiamo un insieme di prodotti e un insieme di risorse. Abbiamo poi dei valori numerici, cioè delle costanti. In particolare, per ogni prodotto abbiamo un prezzo di vendita, per ogni risorsa abbiamo un limite massimo, e per ogni prodotto e ogni risorsa abbiamo una quantità necessaria per la produzione. Abbiamo quindi tre gruppi di costanti, di cui due sono vettori e uno è una matrice. Abbiamo infine le variabili del modello.

Gli insiemi vanno specificati con il costrutto `set` di AMPL, mentre le costanti (o i gruppi di costanti) col costrutto `param`. Le variabili infine col costrutto `var`.

Una volta dichiarati questi oggetti, siamo in grado di scrivere il modello esattamente come il modello riportato sopra, anche se, ovviamente, in formato solo testo, cioè usando delle parole chiave per i simboli matematici non presenti nel formato testo. Il file di modello è allora il seguente:

file pian.mod

```
set PROD;      # insieme dei prodotti
set RISORSE;   # insieme delle risorse

param b{RISORSE};
param a{RISORSE,PROD};
param c{PROD};

var x{j in PROD} >=0;

maximize profit: sum{j in PROD} c[j]*x[j];

s.t. vincolo{i in RISORSE}: sum{j in PROD} a[i,j]*x[j] <= b[i];
```

Gli elementi appartenenti ad ogni insieme ed i valori numerici delle costanti vanno invece riportati nel file di dati, che è allora il seguente:

file pian.dat

```
set PROD:=STANDARD,DELUXE;
set RISORSE:=FERRO,LEVIGATURA,PULITURA;

param:          b:=
FERRO           75
LEVIGATURA      80
PULITURA       60;

param a:        STANDARD  DELUXE:=
FERRO           4         4
LEVIGATURA      4         2
PULITURA       2         5;

param:          c:=
STANDARD        10
DELUXE          15;
```

Si noti che cambiamenti dei valori numerici richiedono solo cambiamenti del file di dati, e anche l'aggiunta di prodotti o risorse non richiede cambiamenti nel file di modello. Possono essere anche prodotti molti file di dati ed un unico file di modello, per risolvere rapidamente molti problemi aventi la stessa struttura ma diversi valori numerici (caso frequente nella pratica).

Alternativamente, sarebbe stato possibile scrivere tutto in un unico file di modello, per esempio nel modo seguente

```
file pian_naive.mod

var x_st;
var x_dl;

maximize profit: 10*x_st + 15*x_dl;

subject to ferro: 4*x_st + 4*x_dl <= 75;
subject to pulitura: 4*x_st + 2*x_dl <= 80;
subject to levigatura: 2*x_st + 5*x_dl <= 60;
subject to nonnegst: x_st >= 0;
subject to nonnegdl: x_dl >= 0;
```

Questo però comporta una serie di svantaggi pratici e non è buono stile di programmazione, per cui è assolutamente da evitare.

Una volta scritti il file di modello e il file di dati è possibile usare il solutore. Per farlo, avviare l'eseguibile `AMPL.exe`, che rende disponibile una finestra a riga di comando. In questa finestra, digitare nell'ordine i seguenti comandi. Notare il punto e virgola alla fine di ogni comando.

```
option solver cplex;           per scegliere il solutore da utilizzare

model <path>pian.mod;          per scegliere il file di modello da considerare

data <path>pian.dat;          per scegliere il file di dati da considerare

solve;                         per risolvere
```

In caso di errori, per resettare usare il comando `reset`; Errori nei file di modello o di dati tali da rendere i file stessi non comprensibili al solutore vengono segnalati in fase di lettura dei file. Per correggere tali errori, cominciare dalla prima segnalazione di errore (viene indicata la riga del file), individuare il problema, risolverlo e ripetere nuovamente il processo. Da notare che esistono anche errori di cui AMPL non è in grado di accorgersi (ad esempio errati valori numerici dei parametri, etc.).

Per vedere infine la soluzione, cioè i valori delle variabili all'ottimo, o ogni altro oggetto di interesse, usare il comando `display`. Per esempio, per vedere il vettore `x`:

```
display x;
```

Se tutto è stato fatto correttamente, la soluzione del modello in esame dovrebbe essere:

```
x [*] :=
  DELUXE    7.5
  STANDARD 11.25
;
```