Comandi e sintassi base di AMPL

Docente: Luigi De Giovanni

Dipartimento di Matematica "Tullio Levi-Civita"
Università degli Studi di Padova

luigi@math.unipd.it
https://www.math.unipd.it/∼luigi/

I comandi base di AMPL

Presentiamo un sottoinsieme minimale di comandi di AMPL per la creazione e la soluzione di modelli di programmazione matematica [1]. Distinguiamo

- comandi per la dichiarazione di modelli (variabili decisionali e domini, funzione obiettivo, vincoli, parametri)
- comandi per la soluzione di modelli con l'ausilio di risolutori esterni
- comandi per la visualizzazione dei risultati

Alcune indicazioni generali:

- AMPL è *case sensitive*
- tutti i comandi terminano con ";"
- si possono inserire commenti preceduti dal simbolo # (viene ignorato il testo fino alla fine della riga)
- i simboli devono essere dichiarati prima del loro utilizzo

Nota: *ci riferiremo in particolare a modelli di Programmazione Lineare (Mista Intera)*

[1] Per la sintassi completa: Robert Fourer, David M. Gay, Brian W. Kernighan. *AMPL A Modeling Language For Mathematical Programming*, Second Edition, Duxbury Thomson, 2003 (disponibile sul sito di AMPL al link `https://ampl.com/resources/the-ampl-book`)

Dichiarazione di variabili decisionali

Le variabili decisionali rappresentano le *incognite* del problema e il loro valore viene calcolato dal risolutore.

- Dichiarazione obbligatoria con il comando `var`

  `var <nome_var> [dominio] ;`

- `dominio` indica il tipo della variabile ed eventuali limiti inferiori e superiori

  `[<tipo>] [>= <lb>] [,] [<= <ub>]`

- `tipo` può essere omesso (variabili reali), `integer` o `binary`

Esempi:

```
var x; #variabile reale tra -infinito e +infinito
var n integer;
var y binary;
var w <=0;
var z integer >= -3 , <= 7;
```

Dichiarazione della funzione obiettivo

Consideriamo una sola funzione obiettivo da minimizzare o massimizzare.

- Dichiarazione obbligatoria con il comando `minimize` o `maximize`

```
minimize <nome_fo> : <espressione> ;
maximize <nome_fo> : <espressione> ;
```

- il nome è obbligatorio
- `espressione` è un'espressione algebrica contenente variabili e può utilizzare anche diversi livelli di parentesi
- si ricorda che i simboli di moltiplicazione e divisione sono rispettivamente `*` (non può essere omesso) e `/`

Esempi:

```
maximize profitto: 3 * x_one + 7.5 * x_two - 12/7 * y;
minimize fo: 12.65 * x1 - sqrt(23) * x2;
minimize fo1: 3*(x+y) + (3-log(23))*(x-(2*y+0.5*z));
```

Dichiarazione di vincoli

I vincoli sono dichiarati uno per volta, ciascuno secondo la seguente sintassi:

- ogni vincolo è dichiarato con il comando `subject to` o `s.t.`

```
subject to <nome_vincolo>: <espressione>
                    <= (oppure >= ) <espressione>;
```

- il nome è obbligatorio

Esempi:

```
subject to v1: 3 * x_one + 7.5 * x_two - 12 * y >= 1000;
s.t. vincolo2: 12.7 * x1 <= sqrt(23) * x2;
s.t. v3: 3*(x+y) >= (3-log(23))*(x-(2*y+0.5*z));
```

Dichiarazione di parametri

È possibile utilizzare dei parametri *numerici* per la dichiarazione di *costanti* da utilizzare nelle espressioni

- Ogni parametro è dichiarato con il comando `param`

  `param <nome_parametro> := <valore>;`

- `valore` è determinato da un numero o da un'espressione che contiene numeri e parametri precedentemente dichiarati

Esempi:

```
param budget := 5000;
param budget_ridotto := sqrt(budget);
param BigM := 1e5;
param epsilon := 1e-4;
param probabilitaScenario1 := 0.35;
```

Scrittura e caricamento di modelli

I modelli possono essere dichiarati inserendo i comandi indicati, uno per volta, direttamente da linea di comando (prompt `ampl:`, che si trova nella finestra *Console* dell'AMPL IDE). In questo caso, il modello viene cancellato uscendo da AMPL (comando `quit`).

Una comoda alternativa è l'utilizzo di un file di testo che contiene (e conserva) tutti i comandi per la dichiarazione di un modello:

- il file che contiene il modello ha estensione `.mod`
- per caricare il file si utilizza il comando `model`

  ```
  ampl: model <nome_file_modello>;
  ```

- `nome_file_modello` deve contenere il percorso completo (ad esempio, in Windows, `C:\Utenti\ModelliAmpl\agricoltore.mod`)
- per evitare di dover indicare il percorso completo, si può utilizzare il comando `cd`

  ```
  ampl: cd C:\Utenti\ModelliAmpl;
  ```

- si può utilizzare il "File Browser" dell'IDE per selezionare la cartella di lavoro (in alternativa a `cd`)

Esempio

Tradurre in AMPL il seguente modello, dove M è una costante:

$$\begin{aligned} \max\ & x_1 + x_2 \\ & x_1 + x_2 \leq 1 + M\,y \\ & x_1 - x_2 \leq 2 \\ & x_1,\, x_2 \geq 0 \\ & x_2 \in \mathbb{Z} \\ & y \in \{0, 1\} \end{aligned}$$

esempio.mod

```
param M := 1000;
var x1 >=0;
var x2 integer >=0;
var y binary;

maximize f: x1 + x2;

subject to v1: x1 + x2 <= 1 + M * y;
s.t. v2: x1 - x2 <= 2;
```

Soluzione di modelli

Una volta caricato, un modello può essere risolto invocando un risolutore. Per risolvere un problema:

1. selezionare un risolutore con il comando `option solver`

   ```
   ampl: option solver <nome_solver> ;
   ```

   nella versione AMPL *free demo* sono disponibili, tra gli altri:
   – `cplex` o `gurobi` per Programmazione Lineare e Programmazione Quadratica Convessa con variabili continue e/o intere
   – `lpsolve` per Programmazione Lineare con variabili continue e/o intere
   – `minos` per Programmazione Non Lineare con variabili continue

2. invocare il comando `solve`

   ```
   ampl: solve ;
   ```

Esempio:

```
ampl: model esempio.mod;
ampl: option solver cplex;
ampl: solve;
```

Visualizzazione dei risultati

Una volta risolto il problema, è possibile visualizzare i risultati utilizzando il comando `display` con seguito dai nomi utilizzati nelle dichiarazioni:

- per visualizzare il valore di un elemento (variabile, funzione obiettivo o parametro)

  `ampl: display <nome_elemento> [, <nome_elemento2>, ...];`

- per un vincolo, è possibile visualizzare lo slack (differenza tra termine noto e valore dell'espressione determinata dalle variabili) usando:

  `ampl: display <nome_vincolo>.slack;`

  in questo modo, ad esempio, è possibile determinare quali vincoli sono saturi (o attivi) e quali invece presentano un margine (slack diverso da 0)

Esempio (continuando da slide precedente):

```
ampl: display f, x1, x2, y;
ampl: display v1.slack, v2.slack;
```

File di comandi

Una sequenza di comandi per il caricamento e la soluzione di modelli e la visualizzazione dei risultati può essere memorizzata in un file testuale di *script*, con estensione `.run`

esempio.run

```
reset;
cd C:\Utenti\ModelliAmpl;
model esempio.mod;
option solver cplex;
solve;
display x1, x2, y;
display f;
```

- Per lanciare un file `.run`, utilizzare il comando `include` dalla console

  `ampl: include <nome_file_comandi>`

- Utilizzare il comando `reset` all'inizio di ogni script per eliminare dati relativi a modelli caricati in precedenza

Esempio:

```
ampl: include C:\Utenti\ModelliAmpl\esempio.run;
```