

AMPL Sintassi ed Esempi

F. Rinaldi

Dipartimento di Matematica
Università di Padova

Corso di Laurea Informatica

Outline

AMPL Sintassi ed Esempi

Sintassi

Esempi di Modellazione in AMPL

Elementi di un problema

- ▶ Insiemi
- ▶ Parametri
- ▶ Variabili
- ▶ Funzione Obiettivo
- ▶ Vincoli
- ▶ Dati

Insiemi

- ▶ Gli insiemi definiscono gli indici di variabili, parametri e vincoli del modello;
- ▶ Dichiarazione (file.mod): `set NomeInsieme;`
- ▶ Definizione (file.dat): `set NomeInsieme:= e1 e2 ... eM;`

Diversi tipi di insieme:

- ▶ insiemi non ordinati:
`set NomeInsieme;`
- ▶ insiemi numerici:
`set NomeInsieme:= inizio .. fine by distanza;`
- ▶ insiemi ordinati:
`set NomeInsieme ordered;`
- ▶ insiemi circolari:
`set NomeInsieme circular;`

Insiemi

- ▶ Gli insiemi definiscono gli indici di variabili, parametri e vincoli del modello;
- ▶ Dichiarazione (file.mod): `set NomeInsieme;`
- ▶ Definizione (file.dat): `set NomeInsieme:= e1 e2 ... eM;`

Diversi tipi di insieme:

- ▶ insiemi non ordinati:
`set NomeInsieme;`
- ▶ insiemi numerici:
`set NomeInsieme:= inizio .. fine by distanza;`
- ▶ insiemi ordinati:
`set NomeInsieme ordered;`
- ▶ insiemi circolari:
`set NomeInsieme circular;`

Insiemi Numerici

- ▶ Gli insiemi numerici non devono necessariamente essere interi
- ▶ AMPL cerca di memorizzare ogni elemento dell'insieme utilizzando il fpn più vicino

- ▶ Provare le seguenti istruzioni:

```
set a:= -5/3 .. 5/3 by 1/3;  
display a;
```

- ▶ 0 e 1 non sono elementi dell'insieme!
- ▶ meglio usare interi

Insiemi Numerici

- ▶ Gli insiemi numerici non devono necessariamente essere interi
- ▶ AMPL cerca di memorizzare ogni elemento dell'insieme utilizzando il fpn più vicino

- ▶ Provare le seguenti istruzioni:

```
set a:= -5/3 .. 5/3 by 1/3;  
display a;
```

- ▶ 0 e 1 non sono elementi dell'insieme!
- ▶ meglio usare interi

Insiemi Numerici

- ▶ Gli insiemi numerici non devono necessariamente essere interi
- ▶ AMPL cerca di memorizzare ogni elemento dell'insieme utilizzando il fpn più vicino

- ▶ Provare le seguenti istruzioni:

```
set a:= -5/3 .. 5/3 by 1/3;  
display a;
```

- ▶ 0 e 1 non sono elementi dell'insieme!
- ▶ meglio usare interi

Operazioni tra Insiemi

Dati due insiemi A e B possibile in AMPL effettuare operazioni elementari:

Operazione	Significato
<code>A union B</code>	insieme di elementi che stanno in A o B
<code>A inter B</code>	insieme di elementi che stanno sia in A che in B
<code>A diff B</code>	insieme di elementi che stanno in A ma non in B
<code>A symdiff B</code>	insieme di elementi che stanno in A o in B ma non in entrambi
<code>card(A)</code>	numero di elementi che stanno in A

`within`: serve a dichiarare un insieme come sottoinsieme di un altro:

```
set A;  
set B within A;
```

IMPORTANTE: Nella definizione A e B devono essere coerenti!

Operazioni tra Insiemi

Dati due insiemi A e B possibile in AMPL effettuare operazioni elementari:

Operazione	Significato
<code>A union B</code>	insieme di elementi che stanno in A o B
<code>A inter B</code>	insieme di elementi che stanno sia in A che in B
<code>A diff B</code>	insieme di elementi che stanno in A ma non in B
<code>A symdiff B</code>	insieme di elementi che stanno in A o in B ma non in entrambi
<code>card(A)</code>	numero di elementi che stanno in A

`within`: serve a dichiarare un insieme come sottoinsieme di un altro:

```
set A;  
set B within A;
```

IMPORTANTE: Nella definizione A e B devono essere coerenti!

Esercizio

Generate due insiemi numerici

- ▶ $A \in [40, 80]$ con intervallo 4
- ▶ $B \in [50, 100]$ con intervallo 5

Calcolare $A \text{ union } B$, $A \text{ inter } B$, $B \text{ diff } A$, $A \text{ symdiff } A$, $\text{card}(A \text{ union } B)$.

Ricorda:

- ▶ le operazioni tra insiemi all'interno di una espressione complessa vengono effettuate da sinistra verso destra
- ▶ gerarchia: inter seguito da union, diff, symdiff (stessa priorità)
- ▶ $A \text{ union } B \text{ inter } C \text{ diff } D$ equivale a $(A \text{ union } (B \text{ inter } C)) \text{ diff } D$

Esercizio

Generate due insiemi numerici

- ▶ $A \in [40, 80]$ con intervallo 4
- ▶ $B \in [50, 100]$ con intervallo 5

Calcolare $A \text{ union } B$, $A \text{ inter } B$, $B \text{ diff } A$, $A \text{ symdiff } A$, $\text{card}(A \text{ union } B)$.

Ricorda:

- ▶ le operazioni tra insiemi all'interno di una espressione complessa vengono effettuate da sinistra verso destra
- ▶ gerarchia: inter seguito da union, diff, symdiff (stessa priorità)
- ▶ $A \text{ union } B \text{ inter } C \text{ diff } D$ equivale a $(A \text{ union } (B \text{ inter } C)) \text{ diff } D$

Operazioni tra Insiemi II

Dato un insieme ordinato, possibile in AMPL effettuare operazioni elementari:

Operazione	Significato
<code>first(A)</code>	primo elemento di A
<code>last(A)</code>	ultimo elemento di A
<code>next(a,A)</code>	elemento di A dopo di a
<code>prev(a,A)</code>	elemento di A prima di a
<code>next(a,A,k)</code>	k-esimo elemento di A dopo di a
<code>prev(a,A,k)</code>	k-esimo elemento di A prima di a
<code>ord(a,A)</code>	posizione di a in A
<code>ord0(a,A)</code>	come <code>ord(a,A)</code> , ma restituisce 0 se a non è in A
<code>member(k,A)</code>	elemento di A in k-esima posizione

Insiemi Ordinati

- Provare le seguenti istruzioni:

```
set a ordered;  
data;  
set a:= a1 a2 a3 a4 a5 a6 a7;  
display first(a), last(a);  
display next('a3',a), next('a3',a,2);  
display ord('a3',a);
```

Insiemi a più dimensioni

dimen: Permette di specificare la dimensione di un insieme contestualmente alla sua dichiarazione:

- ▶ dichiarazione(file.mod): `set TUPLE dimen 3 ;`
- ▶ definizione (file.dat): `set TUPLE:= (1,7,1) (7,1,1) (10,5,3) (pippo, pluto, paperino) ;`

cross: modo alternativo per costruire inisemi di dimensione >1

```
set INS1;  
set INS2;  
set INS3;  
set INS:= INS1 cross INS2 cross INS3;
```

setof effettua procedimento inverso rispetto a cross(file.mod):

```
set INS dimen 3;  
set INS1:=setof{(i,j,h) in INS} i ;  
set INS2:=setof{(i,j,h) in INS} j ;  
set INS3:=setof{(i,j,h) in INS} h ;
```

Parametri

- ▶ `param`: per creare i parametri del modello
- ▶ `dichiarazione(file.mod)`: `param T;`
- ▶ `definizione (file.dat)`: `param T:= 10;`
- ▶ possibile anche specificare delle restrizioni: `param T>1;`

Dichiarazione(file.mod):

Definizione(file.dat):

◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡

Variabili

Variabili sono le incognite del modello:

```
set INS;  
param u{INS};  
  
var prova;  
  
var x{INS} integer;  
var y{INS} binary;  
  
var z{INS} >=0, <=M;  
var w{p in INS} >=0, <=u[p];
```

Funzione Obiettivo e Vincoli

La f.o. rappresenta ciò che vogliamo massimizzare o minimizzare(file.mod):

```
minimize(o maximize) nomeobiettivo:  espressionearitmetica;
```

I vincoli rappresentano le specifiche che la soluzione deve soddisfare(file.mod):

```
subject to nomevincolo:  espressionelogica;  
s.t.  nomevincolo:  espressionelogica;
```

Funzioni Aritmetiche e Operatori Indicizzati

Funzioni Aritmetiche:

Funzione	Significato
<code>abs(x)</code>	valore assoluto di x
<code>floor(x)</code>	approssimazione intera per difetto di x
<code>ceil(x)</code>	approssimazione intera per eccesso di x
<code>sqrt(x)</code>	radice quadrata x
<code>log(x)</code>	logaritmo naturale di x
<code>exp(x)</code>	esponenziale di x
<code>sqrt(x)</code>	radice quadrata di x

Operatori Indicizzati:

Funzione	Significato
Sommatoria	<code>sum{ a in A }</code>
Produttoria	<code>prod{ a in A }</code>
Massimo	<code>max{ a in A }</code>
Minimo	<code>min{ a in A }</code>

Possibile utilizzare sommatorie multiple e condizionate

```
sum{r in MPRIM, f in PROD: r<>3}
```

F. Rinaldi

Un'azienda produce fibra di vetro al metro cubo e desidera pianificare la produzione per le prossime sei settimane. La capacità produttiva è limitata e tale limite varia al variare del periodo considerato. La domanda settimanale è nota per l'intero periodo. I costi di produzione ed immagazzinamento variano al variare del periodo. I dati sono riportati in Tabella 1:

Settimana	Cap.	Domanda	Costo Prod.	Costo Imm.
1	140	100	5	0.2
2	100	120	8	0.3
3	110	100	6	0.2
4	100	90	6	0.25
5	120	120	7	0.3
6	100	110	6	0.4

◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡

Ripartizione di Risorse

- ▶ $sett$: Insieme settimane (6 elementi);
- ▶ $prod_j$: costo di produzione $j \in sett$;
- ▶ imm_j : costo di immagazzinamento $j \in sett$;
- ▶ d_j : Domanda prodotto $j \in sett$;
- ▶ c_j : Capacità produttiva $j \in sett$;
- ▶ x_j : Fibra prodotta $j \in sett$;
- ▶ s_j : Fibra immagazzinata $j \in sett$;

Pianificazione della Produzione: Modello PL

Otteniamo il seguente problema di PL:

$$\begin{aligned}
 \min \quad & \sum_{j=1}^6 prod_j x_j + \sum_{j=1}^6 imm_j s_j \\
 & x_1 = s_1 + d_1 \\
 & x_j + s_{j-1} = s_j + d_j \quad j = 2, \dots, n \\
 & 0 \leq x_j \leq c_j \quad j = 1, \dots, n \\
 & 0 \leq s_j \quad j = 1, \dots, n-1, \quad s_n = 0
 \end{aligned} \tag{1}$$

Esempio 2

Produzione di Parquet alla LegnoPadova SpA

LegnoPadova SpA è un'azienda specializzata nella produzione di parquet. I suoi stabilimenti producono vari tipi di parquet che vengono poi venduti sul mercato generando un determinato profitto. Ciascun settore della produzione può lavorare un numero limitato di ore secondo la Tabella 1. Nella Tabella 2 si riporta il profitto unitario e la domanda massima per tipologia di parquet:

Tabella 1

Settore	Ore
Falegnameria	2000
Verniciatura	1500
Assemblaggio	1700
Verifica	300
Imballaggio	500

Tabella 2

Tipo	Profitto	Domanda
Supereconomy	10	100
Economy	18	80
Biocity	20	60
Vintage	25	40
Gaia	27	40
Family	28	20
Extralux	35	20

Riportiamo infine le ore di lavorazione necessarie per ciascuna tipologia di parquet in Tabella 3. Determinare il piano produttivo che massimizza il profitto totale.

Tabella 3

	Supereconomy	Economy	Biocity	Vintage	Gaia	Family	Extralux
Falegnameria	7	5	9	10	10	12	15
Verniciatura	2	2	2	3	3	3	3
Assemblaggio	2	2	4	7	9	15	18
Verifica	1	1	1	2	1	2	2
Imballaggio	1	1	1	1	2	1	0



Ripartizione di Risorse

- ▶ Insieme di prodotti possibili;
- ▶ Insieme di risorse disponibili;
- ▶ Per ogni risorsa nota la quantità disponibile;
- ▶ Per ogni prodotto noto profitto di vendita unitario e domanda massima;
- ▶ Per ogni prodotto nota quantità di risorsa necessaria per realizzare una unità di prodotto;
- ▶ **Obiettivo:** Massimizzare il profitto.

Ripartizione di Risorse

- ▶ `prodotti`: Insieme prodotti (n elementi);
- ▶ `risorse`: Insieme di risorse disponibili (m elementi);
- ▶ b_i : Quantità disponibile di risorsa $i \in \text{risorse}$;
- ▶ c_j : Profitto di vendita unitario prodotto $j \in \text{prodotti}$;
- ▶ d_j : Domanda massima prodotto $j \in \text{prodotti}$;
- ▶ a_{ij} : Quantità di risorsa $i \in \text{risorse}$ necessaria per realizzare una unità di prodotto $j \in \text{prodotti}$;
- ▶ x_j : Quantità di prodotto $j \in \text{prodotti}$ che si vuole realizzare.

Ripartizione di Risorse: Modello PL

Otteniamo il seguente problema di PL:

$$\begin{aligned}
 \max \quad & \sum_{j=1}^n c_j x_j \\
 \sum_{j=1}^n a_{ij} x_j & \leq b_i \quad i = 1, \dots, m \\
 0 \leq x_j & \leq d_j \quad j = 1, \dots, n
 \end{aligned} \tag{2}$$