

# Lezioni su AMPL & CPLEX

Corso di Ricerca Operativa · Prof. Gianpaolo Oriolo

Gianmaria Leo

Università di Roma “Tor Vergata”

31 Ottobre 2014

# Concludere l'esercitazione del 24 Ottobre... (1/2)

Per rispondere all'ultima domanda dell'esercitazione 24/10 (parte 2):

- introdurre una variabile dello stesso tipo (q.tà dolcificante artificiale)
- modificare il vincolo di miscelazione
- aggiungere il vincolo di qualità

Possibili azioni:

- aggiornare l'insieme delle materie prime
- aggiornare l'insieme dei fornitori (miscele)
- aggiornare la matrice di miscelazione
- aggiornare i vincoli della formulazione

## Concludere l'esercitazione del 24 Ottobre... (2/2)

- Modifiche al file `.mod`:
  - nuovo insieme per il dolcificante artificiale
  - nuovo insieme per le materie prime (unione di insiemi)
  - ridefinire il parametro di miscelazione (insieme materie prime)
  - nuovo parametro per la qualità e la quantità di prodotto richiesto
  - aggiornare il controllo sul parametro di miscelazione
  - modificare il vincolo di miscelazione
  - aggiungere il vincolo di qualità
- Modifiche al file `.dat`:
  - ridefinire l'insieme dei fornitori (miscele)
  - ridefinire i parametri interessati dall'insieme dei fornitori
  - definire i parametri di qualità e quantità totale
- Usare il file `.run` che risponde alla prima domanda dell'esercizio

# AMPL: introduzione all'uso avanzato

1 Insiemi

2 Espressioni

# Tipi di insiemi

- Gli insiemi definiscono gli elementi in base ai quali si indicizzano variabili, parametri e vincoli del modello.
- AMPL permette di definire diversi tipi di insiemi:
  - ① insieme non ordinato:  
`set <setName>;`
  - ② insieme numerico:  
`set <setName> := <startVal> .. <endVal> by <interval>;`
  - ③ insieme ordinato:  
`set <setName> ordered;`
  - ④ insieme ordinato e ciclico:  
`set <setName> circular;`

**N.B.:** gli insiemi numerici non hanno bisogno di un'assegnazione

# Operatori e funzioni su insiemi generici

Operatore/Funzione	risultato
$A \cup B$	unione tra $A$ e $B$
$A \cap B$	intersezione tra $A$ e $B$
$A \setminus B$	differenza di $A$ rispetto a $B$
$A \operatorname{symdiff} B$	differenza simmetrica tra $A$ e $B$
$A \times B$	prodotto cartesiano di $A$ con $B$
$\operatorname{card}(A)$	cardinalità di $A$

# Funzioni su insiemi ordinati

Funzione	risultato
<code>first(A)</code>	primo elemento di $A$
<code>last(A)</code>	ultimo elemento di $A$
<code>next(a,A)</code>	primo elemento che segue $a \in A$
<code>next(a,A,k)</code>	$k$ -esimo elemento che segue $a \in A$
<code>prev(a,A)</code>	primo elemento che precede $a \in A$
<code>prev(a,A,k)</code>	$k$ -esimo elemento che precede $a \in A$
<code>ord(a,A)</code>	posizione di $a \in A$ ( $\text{ord}(a,A) \geq 1$ )
<code>ord0(a,A)</code>	restituisce 0 se $a \notin A$
<code>member(k,A)</code>	elemento in $k$ -esima posizione

Le espressioni di indicizzazione permettono di definire insiemi impliciti, evitando la loro definizione.

- coppie ordinate di elementi, presi rispettivamente da  $A$  e da  $B$

$\{i \text{ in } A, j \text{ in } B\}$

- coppie ordinate di elementi, entrambi presi da  $A$

$\{i \text{ in } A, j \text{ in } A\}$

- elementi di  $A$  che soddisfano un'espressione

$\{i \text{ in } A: \text{<expr>}\}$

- esempio:

$\{i \text{ in } A: \text{costi}[i] \geq 10\}$



- Ogni elemento dell'insieme è caratterizzato da  $d$  attributi
  - dichiarazione (file .mod): `set INS dimen d ;`
  - assegnazione (file .dat): `set INS:= (a1,a2, <...> ad) <...> ;`
- In alternativa (file .mod):
  - `set INS1; set INS2; <...> ; set INSd;`
  - `set INS:= INS1 cross INS2 cross <...> INSd;`
  - `set INS:= {INS1,INS2, <...> ,INSd};`
- L'operatore `setof` effettua l'operazione inversa di `cross` (file .mod):  

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

# Espressioni: operandi

- Le espressioni sono fondamentali nella costruzione di funzione obiettivo, vincoli e condizioni (aggiuntive) su parametri e variabili.

Simbolo	Tipo operando	Tipo risultato
<code>+</code> <code>-</code> <code>*</code> <code>/</code> <code>div</code> <code>^</code> <code>mod</code>	numerico	numerico
<code>&lt;</code> <code>&lt;=</code> <code>==</code> <code>!=</code> <code>&gt;</code> <code>&gt;=</code>	numerico	logico
<code>if-then-else</code>	logico	numerico
<code>or</code> <code>and</code> <code>not</code>	logico	logico

# Espressioni: funzioni matematiche

Funzione	Risultato
<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>sin(x)</code>	seno di $x$
<code>cos(x)</code>	coseno di $x$
<code>tan(x)</code>	tangente di $x$

# Espressioni: funzioni indicizzate

Funzione	Risultato
$\text{sum}\{a \text{ in } A\} \langle \dots \rangle$	somma estesa ai soli elementi in $A$
$\text{prod}\{a \text{ in } A\} \langle \dots \rangle$	produttoria estesa ai soli elementi in $A$
$\text{min}\{a \text{ in } A\} \langle \dots \rangle$	minimo esteso ai soli elementi in $A$
$\text{max}\{a \text{ in } A\} \langle \dots \rangle$	massimo esteso ai soli elementi in $A$