



Ricerca Operativa

Laboratorio: utilizzo di solver
per programmazione
matematica

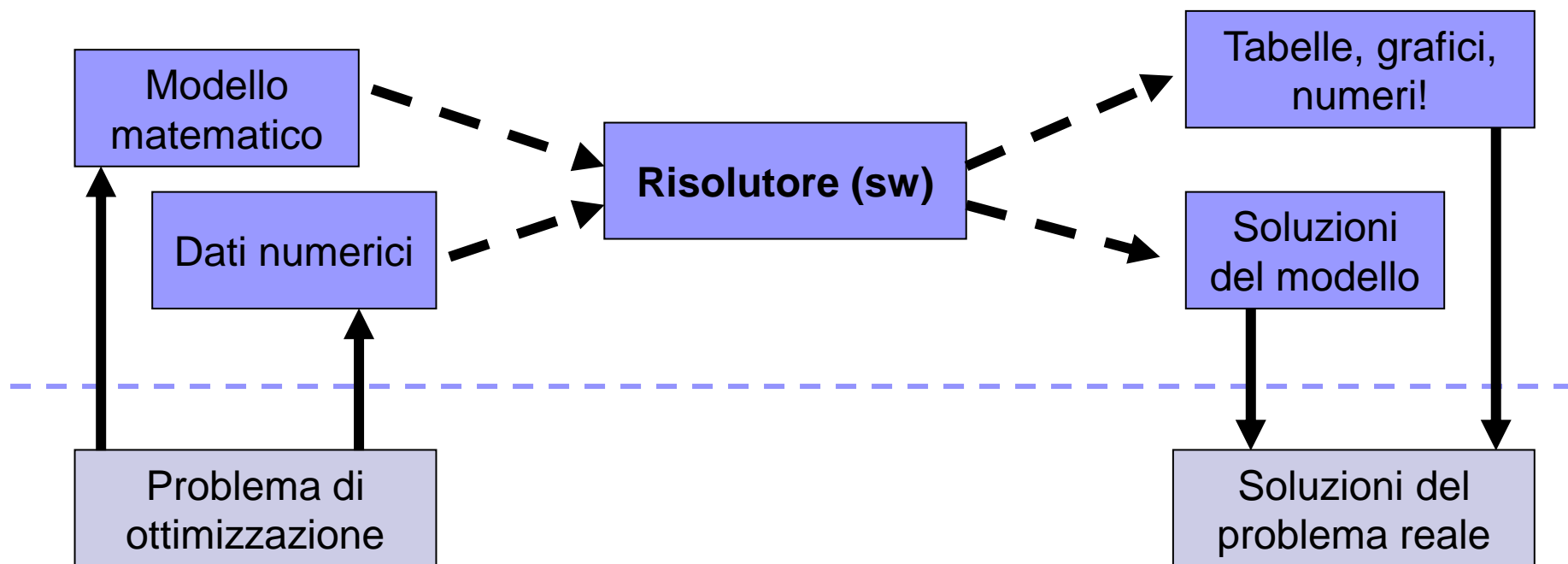
Elementi di un modello di Programmazione Matematica

- **Insiemi:** elementi del sistema;
- **Parametri:** dati del problema;
- **Variabili decisionali o di controllo:** grandezze sulle quali possiamo agire;
- **Vincoli:** relazioni matematiche che descrivono le condizioni di ammissibilità delle soluzioni.
- **Funzione obiettivo:** e la quantità da massimizzare o minimizzare.

Un modello **dichiara** le caratteristiche della soluzione ottima in linguaggio matematico

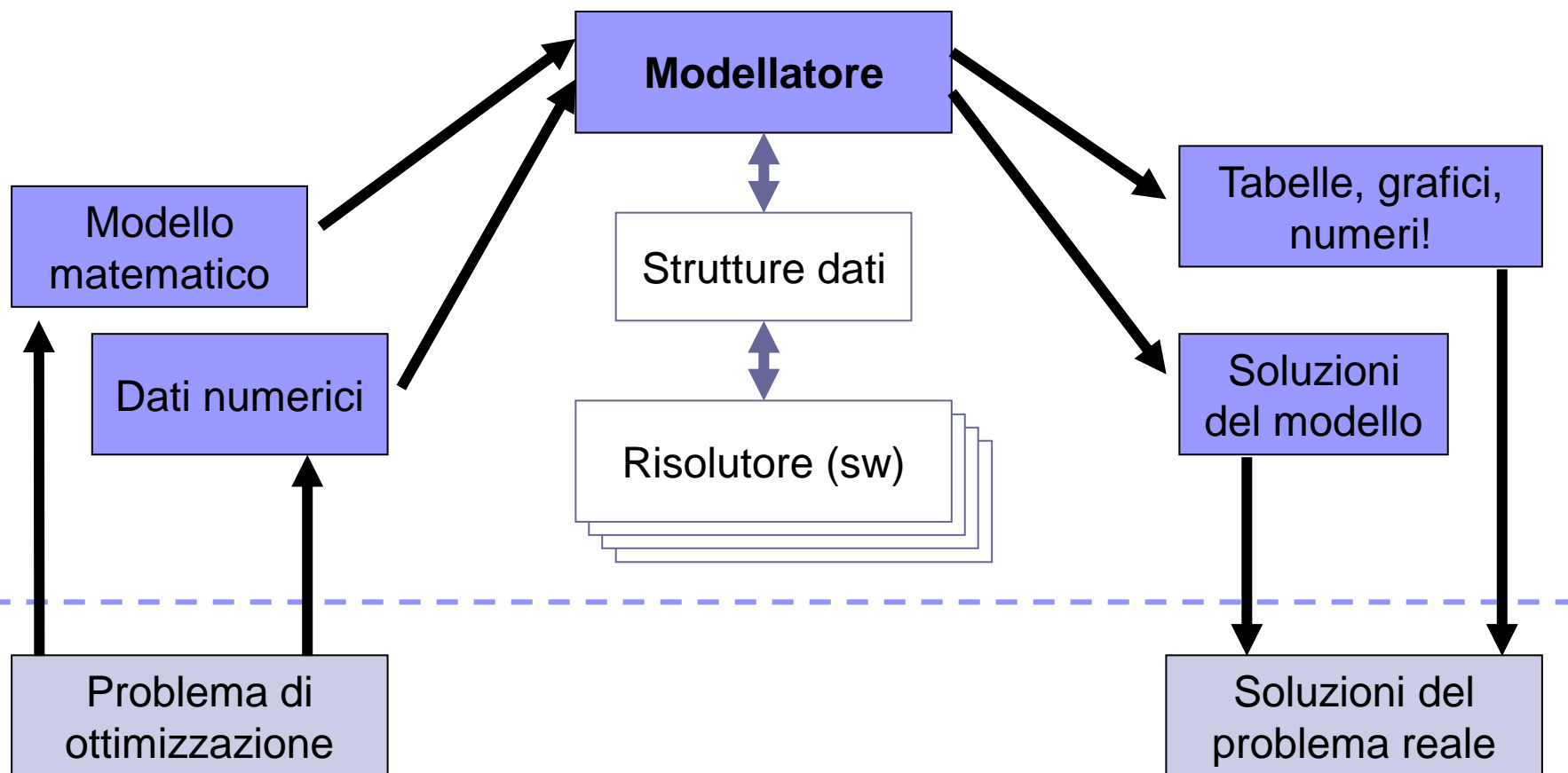
Utilizzo di solver

Un **solver** (o risolutore) è un software che riceve in **input** una descrizione di un problema di ottimizzazione e produce in **output** la soluzione ottima del modello e informazioni ad essa collegate.



Ruolo dei modellatori

Un **modellatore** fornisce un'interfaccia verso un risolutore.



Obiettivi dei modellatori

- Disporre di un **linguaggio** semplice:
 - ad alto livello;
 - simile a quello di modellazione (linguaggio matematico);
 - formalmente strutturato;
 - possibilità di commenti.
- Consentire la **separazione** tra **implementazione** del **modello** e implementazione delle tecniche di **soluzione**
- Dialogare con **diversi solver** (strutture di I/O standard).
- Mantenere la **separazione tra modello e dati** del problema: per cambiare l'istanza basta cambiare i dati, non il modello.
- Linguaggio per **script**.

Possibili configurazioni (alcune)

Modellatori:

- ☐ Foglio elettronico
- ☐ AMPL
- ☐ ZIMPL
- ☐ Lingo
- ☐ OPL
- ☐ Mosel
- ☐ Python, C, Java etc.
- ☐ ...

Solver:

- ☐ Risolutore per fogli elettronici
- ☐ Cplex
- ☐ Gurobi
- ☐ Soplex + SCIP
- ☐ Xpress
- ☐ Minos (Progr. Non Lineare)
- ☐ Lindo
- ☐ GPLK (librerie)
- ☐ Baron
- ☐ Lp_solve
- ☐ ...

Risolutori in fogli elettronici

- Descrivono il modello sotto utilizzando le formule di un foglio elettronico
- Caratteristiche:
 - 👍 Facilità di utilizzo (diffusione, interfaccia «familiare»)
 - 👍 Integrazione con tool di presentazione
 - 👎 Rigidità di utilizzo (no separazione modello/dati)
 - 👎 Motori di ottimizzazione poco efficienti
- Esempi
 - Risolutore (solver) integrato in Microsoft Excel
<http://www.solver.com/excel-solver-help>
 - Solver di LibreOffice
<https://help.libreoffice.org/latest/it/text/scalc/01/solver.html?DbPAR=CALC>

Risolutore di Excel

■ Procedura di attivazione:

- File → Opzioni → Componenti Aggiuntivi → Risolutore → Vai
- Il solver si trova negli strumenti «Dati»

■ Interfaccia intuitiva per indicare

- Funzione obiettivo
- Variabili («by changing cells»)
- Vincoli (inclusa non negatività)

■ Motori di ottimizzazione disponibili

- **Simplesso LP** (per modelli LINEARI): efficiente, esatto
- GRG Non Lineare (per modelli «smooth», cioè funzioni «derivabili» per obiettivo e vincoli): meno efficiente, ottimi locali
- Evolutionary (per modelli «qualsiasi»): metodo euristico (basato su algoritmi genetici)

Esempio

Un coltivatore ha a disposizione 11 ettari di terreno da coltivare a lattuga o a patate. Le risorse a sua disposizione, oltre al terreno, sono: 70 kg di semi di lattuga, 18 t di tuberi, 145 t di concime. Supponendo che il mercato sia in grado di assorbire tutta la produzione e che i prezzi siano stabili, la resa stimata per la coltivazione di lattuga è di 3000 €/ettaro e quella delle patate è di 5000 €/ettaro. L'assorbimento delle risorse per ogni tipo di coltivazione è di 7 kg di semi e 10 t di concime per ettaro di lattuga, e 3 t di tuberi e 20 di concime per le patate. Stabilire quanto terreno destinare a lattuga e quanto a patate in modo da massimizzare la resa economica e sfruttando al meglio le risorse disponibili.

Modello matematico

■ Variabili decisionali:

x_L : quantità in ettari da destinare a lattuga

x_P : quantità in ettari da destinare a patate

■ Funzione obiettivo:

$$\max 3000 x_L + 5000 x_P$$

■ Sistema dei vincoli:

$$x_L + x_P \leq 11 \quad (\text{ettari disponibili})$$

$$7 x_L \leq 70 \quad (\text{semi disponibili})$$

$$3 x_P \leq 18 \quad (\text{tuberi disponibili})$$

$$10 x_L + 20 x_P \leq 145 \quad (\text{concime disponibile})$$

$$x_L \geq 0, x_P \geq 0 \quad (\text{dominio})$$

Soluzione

- Soluzione empirica con foglio elettronico
- Soluzione con metodo grafico (modello lineare, due variabili)
- Soluzione ottima con il *Risolutore*
- [Risorse: **risolutore.xls**]

Esercizio 1.

Per l'assemblaggio di telecomandi, si hanno a disposizione 10 moduli display, 18 moduli di logica di controllo, 12 moduli di trasmissione, 21 tastierini, 9 moduli di navigazione e 10 moduli led. I telcomandi sono di due tipi. Il tipo A richiede un display, un modulo di navigazione, 2 tastierini, 2 moduli di logica, un modulo di trasmissione e un led. Il tipo B richiede 2 display, 3 tastierini, 2 moduli di logica e 3 moduli di trasmissione. Considerando che il tipo A permette un guadagno netto di 3 euro e il tipo B di 6 euro, determinare la produzione che massimizza il guadagno.

Risolvere il problema con il Risolutore di Excel

Modello PLI

Siano x_A e x_B le quantità di telefoni di tipo A e B

$$\max 3 x_A + 6 x_B \quad (\text{guadagno complessivo})$$

s.t.

$$x_A + 2 x_B \leq 10 \quad (\text{display})$$

$$x_A \leq 9 \quad (\text{navigazione})$$

$$2 x_A + 3 x_B \leq 21 \quad (\text{tastierini})$$

$$2 x_A + 2 x_B \leq 18 \quad (\text{logica})$$

$$x_A + 3 x_B \leq 12 \quad (\text{trasmissione})$$

$$x_A \leq 10 \quad (\text{led})$$

$$x_A, x_B \in \mathbb{Z}_+$$

Esercizi

Risolvere con il Risolutore di Excel i seguenti problemi visti a lezione:

- Magliette e Borse (Young Money Makers)
- Problema della dieta
- Problema dei trasporti

[risorse: [risolutore.xls](#)]

AMPL

- **A Mathematical Programming Language**
- Linguaggio di modellazione algebrica
 - Esprime un problema di ottimizzazione in una forma comprensibile ad un solutore
 - Linguaggio algebrico: contiene diverse primitive per esprimere la notazione matematica normalmente utilizzata per problemi di ottimizzazione (es. sommatorie, funzioni matematiche, etc.)
- Caratteristiche:
 - 👍 Flessibilità: separazione modello / dati, script
 - 👍 Integrazione con motori di ottimizzazione allo stato dell'arte
 - 👍 Linguaggio «semplice»: traduzione del modello matematico
 - 👍 Nuovo linguaggio, integrabilità nelle applicazioni

AMPL: versioni disponibili e installazione

- AMPL è disponibile su <http://ampl.com>
 - Software commerciale [a pagamento o trial]
 - Full per scopi didattici (course edition) [a pagamento o trial]
 - «**free demo**» con **max 500 variabili e 500 vincoli** [**gratuita**, include le versioni «free» dei principali solver]
 - <http://ampl.com/try-ampl/>
 - Link [Download free size-limited demos](#)
 - Sezione “AMPL IDE download for [Windows | Linux | Mac]”
 - Ad esempio: <http://ampl.com/demo/amplide.mswin64.zip>
<https://ampl.com/demo/amplide.linux64.tgz>
- **Documentazione**
 - Manuale sintetico sulla pagina del corso
 - The AMPL book:
<http://ampl.com/resources/the-ampl-book/chapter-downloads/>

Accesso ad AMPL «free demo»

- **Installare AMPL** o accedere a una macchina del laboratorio [vedi istruzioni su *moodle*]
- Lanciare **amplide**
 - Installazione propria nel percorso `folder`, eseguire
 - `folder\amplide\amplide.exe` [windows]
 - `folder/amplide/amplide &` [linux]
 - Macchina di laboratorio linux
 - da menu, **Ampl IDE**
 - `/usr/local/bin/amplide &`

Esempio base «contadino»: modello

Comandi per la creazione del modello (da console):

```
#DICHIARAZIONE VARIABILI (# commento fine linea)
```

```
var xL;      #ettari a lattuga
```

```
var xP;      #ettari a patate
```

```
#MODELLO
```

```
maximize      resa:          3000 * xL + 5000 * xP;
```

```
subject to    ettari:        xL + xP <= 11;
```

```
subject to    semi:          7 * xL <= 70;
```

```
s.t.          tuberi:        3 * xP <= 18;
```

```
s.t.          conc:          10 * xL + 20 * xP <= 145;
```

```
s.t.          dominio1:      xL >= 0;
```

```
s.t.          dominio2:      xP >= 0;
```

Esempio base «contadino»: soluzione

Comandi per la soluzione del modello (da console):

```
option solver cplex; # scelta del motore di  
                        ottimizzazione (*)
```

```
solve; # risolve il modello
```

```
display xL, xP; # visualizza il valore  
                  (ottimo) delle variabili
```

(*) possibili solver (vedi eseguibili disponibili nella cartella di installazione):

PLI: `cplex (o cplexamp), gurobi, xpress ...`

PNL: `minos (default solver), baron ...` (non gestiscono variabili intere)

AMPL: file `.mod` e `.dat`

- Memorizzare i **comandi di creazione del modello** in un file di testo con estensione `.mod`
 - Si può usare l'IDE (posizionarsi sul percorso desiderato, menu File → new → `.mod: contadino`) o prepararlo esternamente
- Memorizzare i **dati** (se presenti) in file di testo `.dat`
 - Si può usare l'IDE (posizionarsi sul percorso desiderato, menu File → new → `.dat: contadino`) o prepararlo esternamente
- Caricare il modello con i **comandi**
`model contadino.mod; # carica il modello`
`data contadino.dat; # carica i dati`
- Per **cancellare** modello e dati precedentemente caricati
`reset;`

AMPL: file **.run**

- Memorizzare i **comandi** per **caricare e risolvere** il modello in file di testo con estensione **.run (script)**
 - Si può usare l'IDE (posizionarsi sul percorso desiderato, menu File → new → **.run: contadino**) o prepararlo esternamente
- Per **eseguire** lo script, invocare da console, il comando:
include `contadino.run`;
- **Percorso di lavoro** (*current directory*)
 - I comandi `model`, `data` e `include` fanno riferimento alla directory corrente.
 - La directory corrente è indicata nel file browser o può essere cambiata con il **comando cd** (e.g. `cd /home/luigi/ampl`)
 - I nomi dei file possono essere completati con il percorso, e.g.,
`include ./script/contadino.run`
`model model/contadino.mod`

Esempio base: contadino [risorse]

- `contadino.mod`
- `contadino.dat`
- `contadino.run`

Esercizio 2.

Risolvere il problema dei telecomandi con AMPL

Per le variabili intere:

```
var xA integer;
```

[Risorse]

- **telecomandi.mod**
- **telecomandi.dat**
- **telecomandi.run**

Esempio: sintassi comando **var**

- Dichiarazione di variabili:

```
var nomeVar    [ $\geq$  LB]    [,  $\leq$  UB]  
                                     [integer/binary];
```

- Esercizio: ... al massimo 5 telecomandi di tipo A.

```
var xA  $\geq$  0 ,  $\leq$  5 integer;
```


Modelli generali

- I modelli precedenti includono i «dati» del problema:
 - Se cambiano i dati bisogna cambiare il modello (`.mod`)
 - Poca leggibilità
 - Difficile riportare modifiche del modello
- Separare modello e dati
 - File « `.mod` » con il modello «generale» e la definizione dei parametri del problema
 - File « `.dat` » con i dati attribuiti ai parametri
- Per uno stesso modello possiamo utilizzare diversi file dati (ad esempio «contadino» e «telecomandi»)

Modello generale: mix ottimo di produzione

I insieme dei beni che possono essere prodotti;

J insieme delle risorse disponibili;

P_i profitto (unitario) per il bene $i \in I$;

Q_j quantità disponibile della risorsa $j \in J$;

A_{ij} quantità di risorsa j necessaria per la produzione di un'unità del bene i .

$$\begin{aligned} \max \quad & \sum_{i \in I} P_i x_i \\ \text{s.t.} \quad & \sum_{i \in I} A_{ij} x_i \leq Q_j \quad \forall j \in J \\ & x_i \in \mathbb{R}_+ \ [\mathbb{Z}_+ \mid \{0, 1\}] \quad \forall i \in I \end{aligned}$$

Modello in AMPL: sintassi base

#DICHIARAZIONE INSIEMI

set Prodotti;

set Risorse;

#DICHIARAZIONE PARAMETRI

param maxNumProd; # massimo numero prodotti

param P {Prodotti}; # profitto unitario

param Q {Risorse}; # disponibilità risorsa

param A {Prodotti,Risorse};
risorsa per unità di pr.

var x {Prodotti} >=0 , <= maxNumProd;

maximize profitto: **sum** {i in Prodotti} P[i]*x[i];

subject to disponib {j in Risorse}:

sum {i in Prodotti} A[i,j]*x[i] <= Q[j];

Espressioni indicizzanti

Dati in AMPL: sintatti base

```
set Prodotti := lattuga patata;  
set Risorse  := ettari semi tuberi concime;
```

```
param maxNumProd := 7;
```

```
param          P  :=  
lattuga        3000  
patata         5000  
;
```

```
param          Q  :=  
ettari         11  
semi           70  
tuberi         18  
concime        145  
;
```

```
param A : ettari semi tuberi concime :=  
lattuga  1      7      0      10  
patata   1      0      3      20  
;
```

A{Prodotti=righe , Risorse=colonne}

Esercizio 3.

Risolvere il problema dei telecomandi con AMPL, usando il modello generale.

- ... ogni prodotto ha uno specifico limite superiore!

- .mod

```
param maxNumProd {Prodotti};
```

```
var x {i in Prodotti} >=0, <= maxNumProd[i];
```

- .dat

```
param MaxNumProd := telA 5 telB 999;
```

- [risorse]

- mixOpt.mod - mixOpt.run

- mixOpt.contadino.dat - mixOpt.telecomandi.dat

- Comando **expand** (visualizza modello esteso)

Definizione e dichiarazione di set e param

■ Dichiarazione di insiemi

`set nome;`

`[i in] Set1, [j in] Set2, ...`

■ Dichiarazione di parametri

`param nome [{index_expr}] [default value];`

■ Definizione generale di set e param

`set nome := elem1 elem2 ... elemN;`

`param nome := index1 index2 ... indexN value;`

■ Accesso a parametri indicizzati (parentesi quadre []):

`NomeSet[i] #i-esimo elemento di NomeSet`

`NomeParam[elem1, elem2, ..., elemN]`

Definizione sintetica di parametri

■ Dichiarazioni sintetiche: più parametri con stessi indici

```
param : P   MaxNumProd := #notare i due punti dopo "param"  
      telA   3   5  
      telB   6   999;
```

■ Dichiarazioni sintetiche: tabelle

```
param A : ettari semi tuberi concime := #due punti dopo A  
lattuga   1       7       0       10  
patata    1       0       3       20;
```

■ Dichiarazioni sintetiche: tabelle trasposte

```
param A (tr) : lattuga patata :=  
ettari        1         1  
semi          7         0  
tuberi        0         3  
Concime       10        20;
```

Esercizio 4. Dieta economica

Un dietologo deve preparare una dieta che garantisca un apporto giornaliero di proteine, ferro e calcio di almeno 20 mg, 30 mg e 10 mg, rispettivamente. Il dietologo è orientato su cibi a base di verdura (5 mg/kg di proteine, 6 mg/Kg di ferro e 5 mg/Kg di calcio, al costo di 4 €/Kg), carne (15 mg/kg di proteine, 10 mg/Kg di ferro e 3 mg/Kg di calcio, al costo di 10 €/Kg) e frutta (4 mg/kg di proteine, 5 mg/Kg di ferro e 12 mg/Kg di calcio, al costo di 7 €/Kg). Determinare la dieta di costo minimo.

Risolvere il problema con AMPL (file .mod e .dat separati)

Modello generale: dieta

I insieme delle risorse disponibili;

J insieme delle domande da coprire;

C_i costo (unitario) per l'utilizzo della risorsa $i \in I$;

D_j ammontare della domanda di $j \in J$;

A_{ij} capacità (unitaria) della risorsa i di soddisfare la domanda j .

$$\begin{array}{ll} \min & \sum_{i \in I} C_i x_i \\ \text{s.t.} & \\ & \sum_{i \in I} A_{ij} x_i \geq D_j \quad \forall j \in J \\ & x_i \in \mathbb{R}_+ \text{ [} \mathbb{Z}_+ \text{ | } \{0, 1\} \text{] } \quad \forall i \in I \end{array}$$

Modello PL

- Siano x_1 , x_2 e x_3 le quantità di cibi a base di verdura, carne e frutta, rispettivamente

$$\min \quad 4x_1 + 10x_2 + 7x_3 \quad (\text{costo giornaliero dieta})$$

s.t.

$$5x_1 + 15x_2 + 4x_3 \geq 20 \quad (\text{proteine})$$

$$6x_1 + 10x_2 + 5x_3 \geq 30 \quad (\text{ferro})$$

$$5x_1 + 3x_2 + 12x_3 \geq 10 \quad (\text{calcio})$$

$$x_i \in \mathbb{R}_+, \quad \forall i \in \{1, 2, 3\}$$

Esercizio 5.

Il dietologo vuole inserire alimenti a base di pesce azzurro (10 mg/kg di proteine, 15 mg/kg di ferro e 2 mg/kg di calcio, al costo di 3 euro/kg) nella dieta.

Modificare opportunamente i file relativi al problema.

■ [Risorse]

- `diet.mod - diet.run - diet.1.dat - diet.2.dat`

■ Comandi

- `include diet.run;`
- `reset data; data diet.2.dat;`
- `solve; display x;`

Esercizio 6. Indagine di mercato

Un'azienda pubblicitaria deve svolgere un'indagine di mercato per lanciare un nuovo prodotto. Si deve contattare telefonicamente un campione significativo di persone: almeno 150 donne sposate, almeno 110 donne non sposate, almeno 120 uomini sposati e almeno 100 uomini non sposati. Le telefonate possono essere effettuate al mattino (al costo operativo di 1.1 euro) o alla sera (al costo di 1.6 euro). Le percentuali di persone mediamente raggiunte sono riportate in tabella.

	Mattino	Sera
Donne sposate	30%	30%
Donne non sposate	10%	20%
Uomini sposati	10%	30%
Uomini non sposati	10%	15%
Nessuno	40%	5%

Si noti come le telefonate serali sono più costose, ma permettono di raggiungere un maggior numero di persone: solo il 5% va a vuoto. Si vuole minimizzare il costo complessivo delle telefonate da effettuare (mattina/sera) in modo da raggiungere un campione significativo di persone

Risolvere il problema con AMPL (usare soluzione Es. 4)

Modello PLI

- Siano x_1 e x_2 il numero di telefonate da fare al mattino e alla sera, rispettivamente

$$\min \quad 1.1 x_1 + 1.6 x_2 \text{ (costo totale telefonate)}$$

s.t.

$$0.3x_1 + 0.3x_2 \geq 150 \quad (\text{donne sposate})$$

$$0.1x_1 + 0.2x_2 \geq 110 \quad (\text{donne non sposate})$$

$$0.1x_1 + 0.3x_2 \geq 120 \quad (\text{uomini sposati})$$

$$0.1x_1 + 0.15x_2 \geq 100 \quad (\text{uomini non sposati})$$

$$x_i \in \mathbb{Z}_+, \quad \forall i \in \{1, 2\}$$

[Risorse] [diet.indagine.dat](#)

Esempio: Localizzazione di servizi

Una città è divisa in sei quartieri, dove si vogliono attivare dei centri unificati di prenotazione (CUP) per servizi sanitari. In ciascun quartiere è stata individuata una possibile località di apertura. Le distanze medie in minuti da ciascun quartiere a ciascuna delle possibili località è indicata in tabella. Si desidera che nessun utente abbia un tempo medio di spostamento superiore a 15 minuti per arrivare al CUP più vicino e si vuole minimizzare il numero di CUP attivati.

	Loc. 1	Loc. 2	Loc 3	Loc. 4	Loc. 5	Loc. 6
Q.re 1	5	10	20	30	30	20
Q.re 2	10	5	25	35	20	10
Q.re 3	20	25	5	15	30	20
Q.re 4	30	35	15	5	15	25
Q.re 5	30	20	30	15	5	14
Q.re 6	20	10	20	25	14	5

Modello PLI

Sia $x_i = 1$, se viene aperto il CUP nel quartiere i , 0 altrimenti

$$\min \quad x_1 + x_2 + x_3 + x_4 + x_5 + x_6$$

s.t.

$$x_1 + x_2 \geq 1 \text{ (esigenze q.re 1)}$$

$$x_1 + x_2 + x_6 \geq 1 \text{ (esigenze q.re 2)}$$

$$x_3 + x_4 \geq 1 \text{ (esigenze q.re 3)}$$

$$x_3 + x_4 + x_5 \geq 1 \text{ (esigenze q.re 4)}$$

$$x_4 + x_5 + x_6 \geq 1 \text{ (esigenze q.re 5)}$$

$$x_2 + x_5 + x_6 \geq 1 \text{ (esigenze q.re 6)}$$

$$x_i \in \{0,1\}$$

Modello PLI generale

- Come modello di copertura («dieta»): richiede data preprocessing (**[Risorse]**: `diet.mod` – `diet.CUP.dat`)
- Modello specifico a partire dai dati «grezzi»:

Modello PLI generale

- Come modello di copertura («dieta»): richiede data preprocessing ([Risorse]: diet.mod – diet.CUP.dat)
- Modello specifico a partire dai dati «grezzi»:

I : insieme dei quartieri
 c_{ij} : distanza da $i \in I$ a $j \in I$
 T : distanza soglia

$$\min \sum_{i \in I} x_i$$

$$\text{s.t.} \quad \sum_{\substack{j \in I: \\ c_{ij} \leq T}} x_j \geq 1, \quad \forall i \in I$$

$$x_i \in \{0, 1\}, \quad \forall i \in I$$

Esercizio 7. Trasporto di frigoriferi

Una ditta di produzione di elettrodomestici produce dei frigoriferi in tre stabilimenti e li smista in quattro magazzini intermedi di vendita. La produzione settimanale nei tre stabilimenti A, B e C è rispettivamente di 50, 70 e 20 unità. La quantità richiesta dai 4 magazzini è rispettivamente di 10, 60, 30 e 40 unità. I costi per il trasporto di un frigorifero tra gli stabilimenti e i magazzini 1, 2, 3 e 4 sono i seguenti:

- dallo stabilimento A: 6, 8, 3, 4 euro;
- dallo stabilimento B: 2, 3, 1, 3 euro;
- dallo stabilimento C: 2, 4, 6, 5 euro.

Utilizzare AMPL per determinare il piano di trasporti di costo minimo, considerando che non sono ammesse rimanenze alla fine della settimana e che lo stesso modello dovrà essere utilizzato per diverse settimane.

Modello PLI

- Sia x_{ij} il numero di frigoriferi prodotti nello stabilimento i e smistati nel magazzino j

$$\begin{aligned} \min \quad & 6x_{A1} + 8x_{A2} + 3x_{A3} + 4x_{A4} + \\ & + 2x_{B1} + 3x_{B2} + 1x_{B3} + 3x_{B4} + \\ & + 2x_{C1} + 4x_{C2} + 6x_{C3} + 5x_{C4} \end{aligned}$$

s.t.

$$x_{A1} + x_{A2} + x_{A3} + x_{A4} \leq 50 \quad (\text{capacità produttiva stabilimento A})$$

$$x_{B1} + x_{B2} + x_{B3} + x_{B4} \leq 70 \quad (\text{capacità produttiva stabilimento B})$$

$$x_{C1} + x_{C2} + x_{C3} + x_{C4} \leq 20 \quad (\text{capacità produttiva stabilimento C})$$

$$x_{A1} + x_{B1} + x_{C1} \geq 10 \quad (\text{domanda magazzino 1})$$

$$x_{A2} + x_{B2} + x_{C2} \geq 60 \quad (\text{domanda magazzino 2})$$

$$x_{A3} + x_{B3} + x_{C3} \geq 30 \quad (\text{domanda magazzino 3})$$

$$x_{A4} + x_{B4} + x_{C4} \geq 40 \quad (\text{domanda magazzino 4})$$

$$x_{ij} \in \mathbb{Z}_+ \quad \forall i \in \{A, B, C\}, j \in \{1, 2, 3, 4\}$$

Modello generale: trasporti

I insieme dei centri di offerta; O_i ammontare dell'offerta in $i \in I$;

J insieme dei centri di domanda; D_j ammontare della domanda in $j \in J$.

C_{ij} costo (unitario) per il trasporto da $i \in I$ a $j \in J$;

$$\begin{array}{ll} \min & \sum_{i \in I} \sum_{j \in J} C_{ij} x_{ij} \\ \text{s.t.} & \\ & \sum_{j \in J} x_{ij} \leq O_i \quad \forall i \in I \\ & \sum_{i \in I} x_{ij} \geq D_j \quad \forall j \in J \\ & x_{ij} \in \mathbb{R}_+ \text{ [} \mathbb{Z}_+ \text{ | } \{0, 1\} \text{] } \quad \forall i \in I, j \in J \end{array}$$

display e espressioni indicizzanti

- Visualizza elementi del modello e della soluzione

```
display elemento1[, elemento2, ...];
```

```
display {ind_expr} elemento[indici]
```

- Esempi di espr. indicizzanti *ind_expr* (condizionate)

- `display {i in I} C[i, "m3"];`

- `display {i in I, j in J: C[i,j] >= 5} C[i,j];`

- `display {i in I: origine[i].body - O[i] != 0}
O[i] - origine[i].body;`

il «corpo» (**body**) di un vincolo è la parte che include le variabili (tutto ciò che non è «termine noto»); l'espressione `vincolo.body` indica il valore di *body* in corrispondenza di una soluzione

- Le espressioni indicizzanti possono essere usate anche per la definizione di variabili, parametri, vincoli etc.

- **[Risorse]:** `trasporto.mod - trasporto_frigo.dat -
trasporto_frigo.run`

Script: controllo del flusso

- Sequenza: ordine di esecuzione standard

- Iterazione:

- ☐ **for**{ *espressione indicizzante* } { ... }
- ☐ **repeat while**(*condizione logica true*) { ... }
- ☐ **repeat** { ... } **while**(*condizione logica true*) ;
- ☐ **repeat** { ... } **until**(*condizione logica false*) ;
- ☐ **repeat until**(*condizione logica false*) { ... }
- ☐ Uso di **break** e **continue**

- Selezione

- ☐ **if** (*condizione*) **then** { ... } **else** { ... }
- ☐ Uso di **break**

Script: altri comandi utili

- **let** *parametro := valore;*

- Anche altri parametri definiti nel `.run` (all'inizio)

- **printf** " *stringa formato* " , *elenco valori*

- C – like

- ```
printf "il valore di x[%d] è %7.2f\n", indice, x[indice]
```

- **fix** (o **unfix**) *variabile := valore;*

- ... e molto altro ...

- I comandi e il controllo di flusso nei file **.run**

# Esercizio: distribuzione PC

Un'azienda assembla dei PC in tre diversi stabilimenti con diverso costo unitario di produzione. I PC sono venduti a cinque clienti bancari e si sopportano dei costi di trasporto (inclusi gli oneri di importazione) per spedire un PC da ciascuno stabilimento a ciascun cliente. Sono definite le richieste di PC di ogni cliente e la produzione di ciascuno stabilimento è limitata. Non sono ammessi eccessi di produzione. I dati sono riassunti nella tabella seguente.

Scrivere in AMPL un modello del problema e fornire la soluzione, in termini di costo complessivo di trasporto e di quantità trasportate tra stabilimenti e sedi bancarie.



# Esercizio: distribuzione PC (dati)

| Produzione     |             |          | Costi di trasporto |            |              |               |            |
|----------------|-------------|----------|--------------------|------------|--------------|---------------|------------|
| Unità          | costo unit. | Capacità | Banca Intesa       | Uni Credit | Anton Veneta | Credit Suisse | Banca Cina |
| Italia         | 220         | 10000    | 5,5                | 7,5        | 6,9          | 8,0           | 10,3       |
| Cina           | 180         | 20000    | 15,0               | 14,3       | 13,0         | 16,4          | 5,0        |
| Francia        | 200         | 10000    | 6,0                | 7,8        | 6,3          | 6,8           | 11,0       |
| <b>Domanda</b> |             |          | 7100               | 3400       | 9700         | 5200          | 3050       |

# Esercizio: distribuzione PC (modello base)

Insiemi:  $S$  (stabilimenti) e  $B$  (banche)

Parametri:  $w_i$  (costi prod.),  $c_{ij}$  (costi trasp.),  $a_i$  (capacità prod.),  $b_j$  (richieste)

$$\min \sum_{i \in S, j \in B} (w_i + c_{ij}) x_{ij}$$

$$\text{s. t. } \sum_{j \in B} x_{ij} \leq a_i, \quad \forall i \in S$$

$$\sum_{i \in S} x_{ij} = b_j, \quad \forall j \in B \quad (\text{no eccessi di produzione} \Rightarrow \text{uguaglianza})$$

$$x_{ij} \in Z_+ \quad \forall i \in S, j \in B$$

# Esercizio: distribuzione PC (scenari)

Per bilanciare la produzione, l'azienda richiede che nello stabilimento italiano si assemblino almeno il 25% dei PC.

1. Inoltre, l'azienda richiede che nello stabilimento italiano si assemblino **almeno il 30% dei PC (ipotesi 1)** [o **il 40% dei PC (ipotesi 2)**] prodotti in ciascuno degli altri stabilimenti
2. Produrre un elenco che permetta di individuare i casi in cui una banca riceve forniture da un solo paese.
3. Visualizzare l'utilizzo delle capacità produttive per paese.
4. Conviene, nell'ipotesi 2, potenziare di 5000 unità la produzione in Cina, al costo di 4.000 euro?
5. Tornare alla situazione senza bilanciamenti e studiare gli effetti della diminuzione (a intervalli di 6 euro) del costo di produzione in Italia (diminuzione massima di 40 euro), indicando in quali casi in Italia la produzione complessiva supera quella della Francia.

# Esercizio: distribuzione PC (modello esteso)

Insiemi:  $S$  (stabilimenti) e  $B$  (banche)

Parametri:  $w_i$  (costi prod.),  $c_{ij}$  (costi trasp.),  $a_i$  (capacità prod.),  $b_j$  (richieste),  
 $\alpha$  (bilanciamento generale),  $\beta$  (bilanciamento singolo)

$$\min \sum_{i \in S, j \in B} (w_i + c_{ij}) x_{ij}$$

$$\text{s. t. } \sum_{j \in B} x_{ij} \leq a_i \quad \forall i \in S$$

$$\sum_{i \in S} x_{ij} = b_j \quad \forall j \in B \quad (\text{no eccessi di produzione} \Rightarrow \text{uguaglianza})$$

$$\sum_{j \in B} x_{\text{Italia } j} \geq \alpha \sum_{i \in S, j \in B} x_{ij}$$

$$\sum_{j \in B} x_{\text{Italia } j} \geq \beta \sum_{j \in B} x_{ij} \quad \forall i \in S \setminus \{\text{Italia}\}$$

$$x_{ij} \in \mathbb{Z}_+ \quad \forall i \in S, j \in B \quad \text{[Risorse] PC.mod - PC.dat - PC.run}$$

# Esercizio: fonderia (testo)

Un'acciaieria acquista rottame di quattro tipi differenti (T1, T2, T3, T4) per ottenere due leghe (L1, L2) con caratteristiche chimiche differenti. I quattro tipi di rottame hanno i seguenti contenuti in percentuale di Piombo, Zinco e Stagno, e il seguente prezzo unitario di acquisto (in migliaia di € a tonnellata).

|               | <b>T1</b> | <b>T2</b> | <b>T3</b> | <b>T4</b> |
|---------------|-----------|-----------|-----------|-----------|
| <b>Piombo</b> | 40%       | 30%       | 25%       | 38%       |
| <b>Zinco</b>  | 35%       | 40%       | 35%       | 32%       |
| <b>Stagno</b> | 25%       | 30%       | 40%       | 30%       |
| <b>prezzo</b> | 2.5       | 1.8       | 2         | 2.2       |

La lega L1 deve avere un contenuto non superiore al 30% di piombo, al 60% di zinco e al 42% di stagno.

La lega L2 deve avere un contenuto non superiore al 46% di piombo, al 38% di zinco e al 56% di stagno.

Definire le quantità di ciascun tipo di rottame da utilizzare in ciascuna delle leghe in modo da minimizzare il costo complessivo e soddisfare esattamente un ordine di 1500 tonnellate di lega L1 e 2000 tonnellate di lega L2.

## Esercizio: fonderia (modello specifico)

$$\begin{aligned} \min \quad & 2.5(x_{11} + x_{12}) + 1.8(x_{21} + x_{22}) + 2(x_{31} + x_{32}) + 2.2(x_{41} + x_{42}) \\ \text{s.t.} \quad & x_{11} + x_{21} + x_{31} + x_{41} = 1500 \\ & x_{12} + x_{22} + x_{32} + x_{42} = 2000 \\ & 0.4x_{11} + 0.3x_{21} + 0.25x_{31} + 0.38x_{41} \leq 0.3(x_{11} + x_{21} + x_{31} + x_{41}) \\ & 0.35x_{11} + 0.4x_{21} + 0.35x_{31} + 0.32x_{41} \leq 0.6(x_{11} + x_{21} + x_{31} + x_{41}) \\ & 0.25x_{11} + 0.3x_{21} + 0.40x_{31} + 0.3x_{41} \leq 0.42(x_{11} + x_{21} + x_{31} + x_{41}) \\ & 0.4x_{12} + 0.3x_{22} + 0.25x_{32} + 0.38x_{42} \leq 0.46(x_{12} + x_{22} + x_{32} + x_{42}) \\ & 0.35x_{12} + 0.4x_{22} + 0.35x_{32} + 0.32x_{42} \leq 0.38(x_{12} + x_{22} + x_{32} + x_{42}) \\ & 0.25x_{12} + 0.3x_{22} + 0.40x_{32} + 0.3x_{42} \leq 0.56(x_{12} + x_{22} + x_{32} + x_{42}) \\ & x_{ij} \geq 0 \quad i = 1, \dots, 4 \quad j = 1, 2 \end{aligned}$$

# Esercizio: fonderia (*todo*)

## ■ Modello generale (...)

**Insiemi:**  $I$  (ROTTAMI);  $J$  (LEGHE);  $K$  (METALLI).

**Parametri:**  $C_i$  (Prezzo rottame  $i \in I$ );  $R_j$  (Ordine lega  $j \in J$ );  $A_{k,i}$  (contenuto metallo  $k \in K$  in rottame  $i \in I$ );  $U_{k,j}$  (contenuto max di metallo  $k \in K$  nella lega  $j \in J$ ).

**Variabili:**  $x_{ij}$  (acquisti di rottame  $i \in I$  usati per la lega  $j \in J$ )

**Modello PL:**

$$\begin{aligned} \min \quad & \sum_{i \in I} C_i \sum_{j \in J} x_{ij} \\ \text{s.t.} \quad & \sum_{i \in I} A_{ki} x_{ij} \leq U_{kj} \sum_{i \in I} x_{ij} \quad \forall j \in J, k \in K \\ & \sum_{i \in I} x_{ij} = R_j \quad \forall j \in J, \\ & x_{ij} \in \mathbb{R}_+ \quad \forall i \in I, j \in J \end{aligned}$$

## ■ Implementazione in AMPL (file .mod e .dat ...)

[Risorse] rottame.\*

# Esercizio: ovile (testo)

L'azienda Ovile produce due tipi di cibo per animali: granulare e in polvere. Le materie prime utilizzate per la produzione sono: avena, mais e melassa. Tali materie, ad eccezione della melassa, devono essere macinate prima della lavorazione. In seguito si mescolano le varie sostanze e si processa il composto (granulazione o polverizzazione) al fine di ottenere i due diversi tipi di prodotto. La percentuale di proteine, grassi e fibre contenute nelle materie prime e i requisiti nutrizionali (in %) che i prodotti devono soddisfare sono riportati nella seguente tabella.

| <b>Materie prime</b> | Proteine   | Grassi   | Fibre    |
|----------------------|------------|----------|----------|
| Avena                | 13.6       | 7.1      | 7        |
| Mais                 | 4.1        | 2.4      | 3.7      |
| Melassa              | 5          | 0.3      | 25       |
| <b>Requisiti</b>     | $\geq 9.5$ | $\geq 2$ | $\leq 6$ |

Di seguito sono riportati la disponibilità delle materie prime e i costi unitari per il loro acquisto.

| <b>Materie prime</b> | Disponibilità (kg) | Costo (Euro/kg) |
|----------------------|--------------------|-----------------|
| Avena                | 11900              | 0.13            |
| Mais                 | 23500              | 0.17            |
| Melassa              | 750                | 0.12            |

Infine, i costi di produzione (per un kg di materie prime) sono riportati nella seguente tabella.

| Macina | Mescola | Granulazione | Polverizzazione |
|--------|---------|--------------|-----------------|
| 0.25   | 0.05    | 0.42         | 0.17            |

Tenendo conto che la domanda giornaliera (esatta) è di 9 tonnellate per il prodotto granulare e di 12 tonnellate per quello in polvere, determinare il piano produttivo che minimizza il costo totale.



# Esercizio: ovile (modello specifico)

Variabili:  $x_{ij}$  è la quantità (in kg) di materia prima  $i$  (1=avena, 2=mais, 3=melassa) destinata al tipo di prodotto  $j$  (1=granulare, 2=polvere)

$\min 0.13(x_{11} + x_{12}) + 0.17(x_{21} + x_{22}) + 0.12(x_{31} + x_{32})$  costi avena, mais e melassa  
 $+ 0.25(x_{11} + x_{12} + x_{21} + x_{22})$  kg macinati  
 $+ 0.05(x_{11} + x_{12} + x_{21} + x_{22} + x_{31} + x_{32})$  kg mescolati  
 $+ 0.42(x_{11} + x_{21} + x_{31})$  kg in granuli  
 $+ 0.17(x_{12} + x_{22} + x_{32})$  kg in polvere

$0.136x_{11} + 0.041x_{21} + 0.05x_{31} \geq 0.095(x_{11} + x_{21} + x_{31})$  min 9.5% proteine per granulare

$0.071x_{11} + 0.024x_{21} + 0.003x_{31} \geq 0.02(x_{11} + x_{21} + x_{31})$  min 2% grassi per granulare

$0.07x_{11} + 0.037x_{21} + 0.25x_{31} \leq 0.06(x_{11} + x_{21} + x_{31})$  max 6% fibre per granulare

$0.136x_{12} + 0.041x_{22} + 0.05x_{32} \geq 0.095(x_{12} + x_{22} + x_{32})$  min 9.5% proteine per polvere

$0.071x_{12} + 0.024x_{22} + 0.003x_{32} \geq 0.02(x_{12} + x_{22} + x_{32})$  min 2% grassi per polvere

$0.07x_{12} + 0.037x_{22} + 0.25x_{32} \leq 0.06(x_{12} + x_{22} + x_{32})$  max 6% fibre per polvere

$x_{11} + x_{12} \leq 11900$  disponibilità avena

$x_{21} + x_{22} \leq 23500$  disponibilità mais

$x_{31} + x_{32} \leq 750$  disponibilità melassa

$x_{11} + x_{21} + x_{31} = 9000$  domanda granulare

$x_{12} + x_{22} + x_{32} = 12000$  domanda polvere

$x_{ij} \geq 0 \quad i = 1, \dots, 3 \quad j = 1, 2$

# Esercizio: ovile (*todo*)

## ■ Modello generale (...)

**Insiemi:**  $I$  (MATERIE);  $J$  (CIBI);  $K$  (SOSTANZE);  $R$  (LAVORAZIONI).

**Parametri:** ...;  $P_{rij} = 1$  se è richiesta la lavorazione  $r$  sulla materia  $i$  per il cibo  $j$ , 0 altrimenti; ...

**Modello PL:**

$$\begin{aligned} \min \quad & \sum_{i \in I} C_i \sum_{j \in J} x_{ij} + \sum_{r \in R} F_r \sum_{i \in I} \sum_{j \in J} P_{rij} x_{ij} \\ \text{s.t.} \quad & \sum_{i \in I} A_{ik} x_{ij} \geq B_k \sum_{i \in I} x_{ij} & \forall j \in J, k \in K : B_k > 0 \\ & \sum_{i \in I} A_{ik} x_{ij} \leq U_k \sum_{i \in I} x_{ij} & \forall j \in J, k \in K : U_k < 1 \\ & \sum_{j \in J} x_{ij} \leq Q_i & \forall i \in I, \\ & \sum_{i \in I} x_{ij} = D_j & \forall j \in J, \\ & x_{ij} \in \mathbb{R}_+ & \forall i \in I, j \in J \end{aligned}$$

## ■ Implementazione in AMPL (file .mod e .dat ...)

[Risorse] [ovile.\\*](#)