

# Modellazione MIP

---

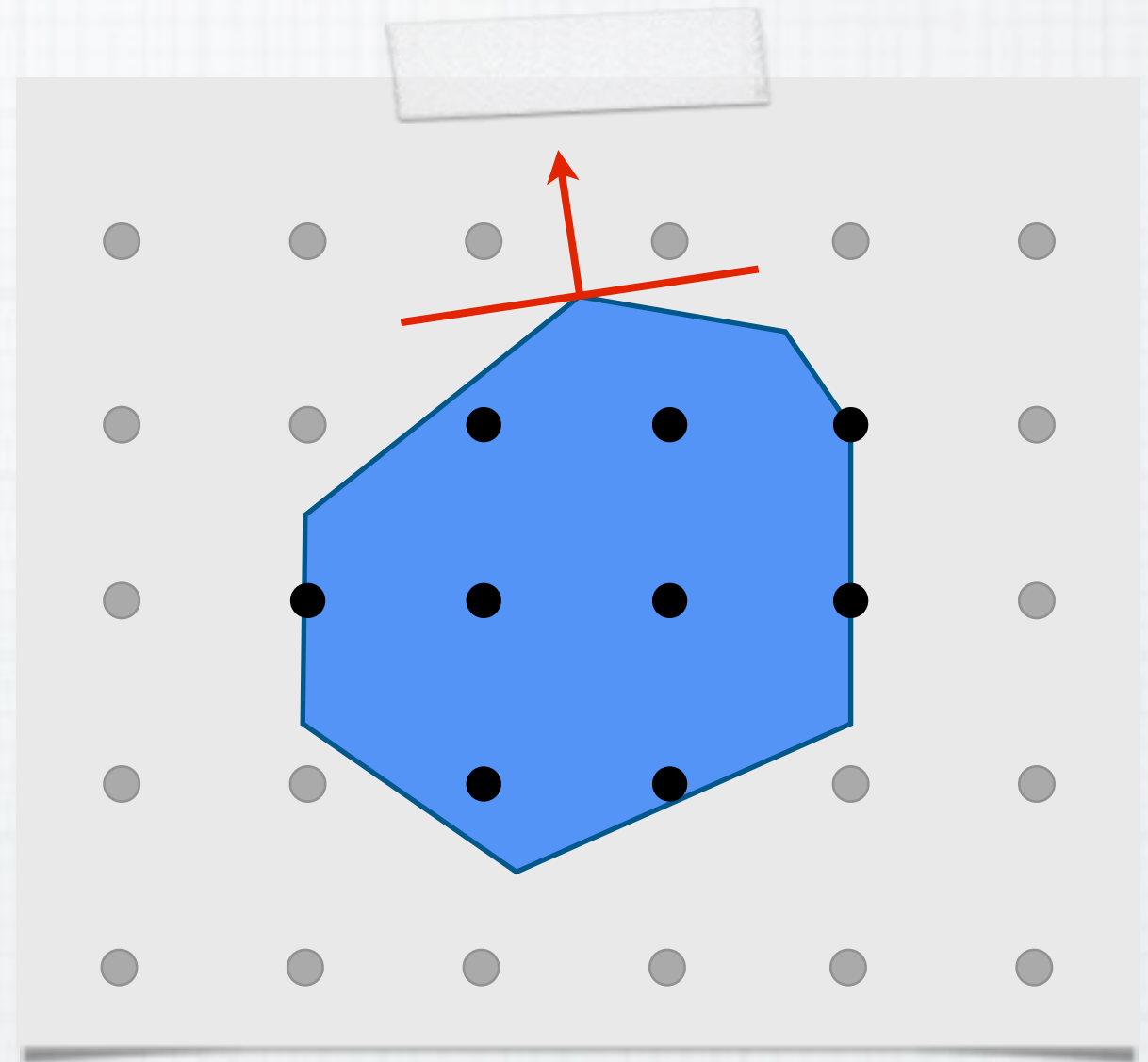
LP, GAMS

Domenico Salvagnin

# Programmazione Lineare Intera

$$\begin{array}{ll}\min & c^T x \\ \text{s.t.} & Ax \leq b \\ & l \leq x \leq u\end{array}$$

$$x_j \in \mathbb{Z}, j \in J$$



# Cosa abbiamo a disposizione?

- \* Linear Programming (LP) solvers
  - \* commerciali (Cplex, Gurobi, Xpress)
  - \* open-source (CLP, SoPlex, GLPK)
- \* Mixed Integer Programming (MIP) solvers
  - \* commerciali (Cplex, Gurobi, Xpress)
  - \* open-source (SCIP, CBC)

# Come si usano?

- \* Due principali modi di interazione:
  - \* scrittura di un modello lineare (intero) in formato LP o MPS, a mano o tramite software di modellazione (GAMS, AMPL), da dare in pasto al solver a scatola chiusa.
  - \* interfacciamento a livello di codice (C/C++/Java/Python) tramite le API fornite dal solver (che viene linkato come libreria). In questo modo è possibile personalizzare il comportamento del solver tramite callbacks.



# Esempio: problema trasporto

$$\min \sum_{i,j} c_{ij} x_{ij}$$

minimizzare

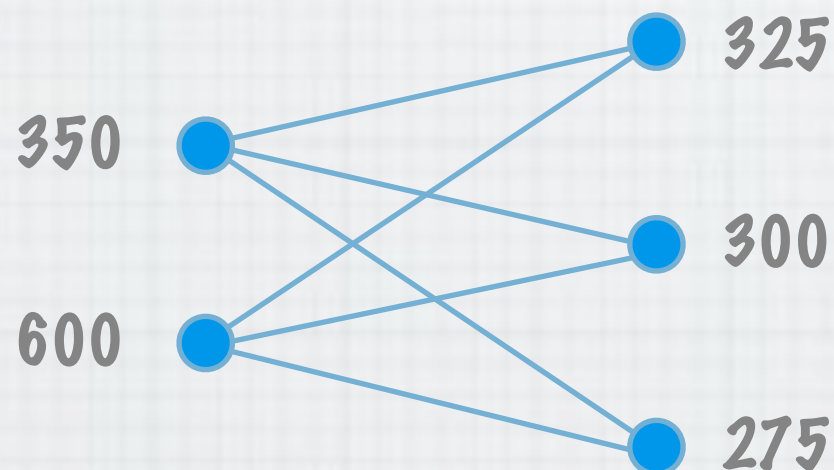
$$\sum_j x_{ij} \leq a_i \quad \forall i$$

soddisfare offerta

$$\sum_i x_{ij} \geq b_j \quad \forall j$$

soddisfare domanda

$$x_{ij} \geq 0$$



	1	2	3
A	2.5	1.7	1.8
B	2.5	1.8	1.4

# Formato LP

**Minimize (Maximize)**

[obj\_name:] funzione obiettivo (es:  $x_1 + 3 x_2 - y$ )

**s.t.**

[name:] vincoli (es:  $x_1 + 2 x_2 \geq 5$ )

...

**Bounds**

... (es:  $0 \leq x_1 \leq 5$ )

**Generals**

... (lista variabili general integers)

**Binaries**

... (lista variabili binarie)

**End**

# Esempio file LP (transport.lp)

funzione obiettivo → Minimize  
obj:  $2.5 x_{a1} + 1.7 x_{a2} + 1.8 x_{a3} + 2.5 x_{b1} + 1.8 x_{b2} + 1.4 x_{b3}$   
s.t.

vincoli → supA:  $x_{a1} + x_{a2} + x_{a3} \leq 350$   
supB:  $x_{b1} + x_{b2} + x_{b3} \leq 600$   
dem1:  $x_{a1} + x_{b1} \geq 325$   
dem2:  $x_{a2} + x_{b2} \geq 300$   
dem3:  $x_{a3} + x_{b3} \geq 275$   
End



# Come si risolve?

## Cplex:

```
# cplex
...
> read transport.lp
> opt
...
> display solution var *
...
> quit
```

## SCIP:

```
# scip
...
> read transport.lp
> optimize
...
> display solution
...
> quit
```

## oppure

```
# scip -f transport.lp
```

**Non dimenticare help in linea! > help**



# GAMS

## General Algebraic Modeling System

- \* descrizione problema di ottimizzazione in notazione “quasi” algebrica
- \* possibilità di riciclare il modello con istanze diverse
- \* modello auto-documentante
- \* flessibilità sulle generazione dell’output (in ogni caso user-friendly)
- \* molti solver diversi a disposizione
- \* ...

# Struttura file GAMS

## Sets

...; // insiemi di indici del problema

## Parameters

...; // parametri numerici del problema

## Variables

...; // variabili del problema

## Equations

...; // definizioni dei vincoli del problema

Model \_modello /all/; // definizione modello

solve \_modello using \_solver minimizing \_var;

display ...;

# Regole del gioco

- \* nomi delle variabili limitate a 10 caratteri
- \* case insensitive
- \* praticamente free-form
- \* bisogna dichiarare qualcosa prima di usarla
- \* statement terminati da ;
- \* si usano simboli strani (=e=, =l= e =g=, al posto di =, ≤ e ≥ e /.../ per delimitare un insieme, al posto di {...})



# Esempio: file GAMS

## Sets

i plants /A, B/

j clients /1, 2, 3/;

## Parameters

a(i) capacity /A 350, B 600/

b(j) demand /1 325, 2 300, 3 275/;

## Table c(i,j) cost

	1	2	3
A	2.5	1.7	1.8
B	2.5	1.8	1.4;

## Variables

x(i,j) shipment quantities

z total cost;

Positive variable x;



# Esempio: file GAMS

Equations

cost objective function

supply(i) supply limit

demand(j) satisfy demand;

cost .. z =e= sum((i,j), c(i,j)\*x(i,j));

supply(i) .. sum(j, x(i,j)) =l= a(i);

demand(j) .. sum(i, x(i,j)) =g= b(j);

Model transport /all/;

solve transport using lp minimizing z;

display x.l;

# Qualche link...

- \* IBM ILOG Cplex: <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>
- \* FICO Xpress: <http://www.fico.com/en/Products/DMTools/Pages/FICO-Xpress-Optimization-Suite.aspx>
- \* GUROBI: <http://www.gurobi.com/>
- \* SCIP: <http://scip.zib.de>
- \* Coin-OR project (CLP, CBC): <http://www.coin-or.org>
- \* GAMS: <http://www.gams.com>
- \* NEOS Servers: <http://neos.mcs.anl.gov/neos/solvers/index.html>