Corso di

Modelli e metodi di ottimizzazione

Alberto Bemporad

http://www.dii.unisi.it/~bemporad

Università di Siena

Dipartimento di Ingegneria dell'Informazione

Sommario

- Obiettivo del corso: **risolvere problemi decisionali complessi in maniera ottimale mediante l'ausilio del calcolatore**

- Contenuti del corso:

  1. **Teoria**: come formulare il problema decisionale in forma matematica.

  2. **Esempi applicativi**: come utilizzare risolutori numerici per ottenere la decisione ottima.

http://www.dii.unisi.it/~bemporad/teaching/danieli

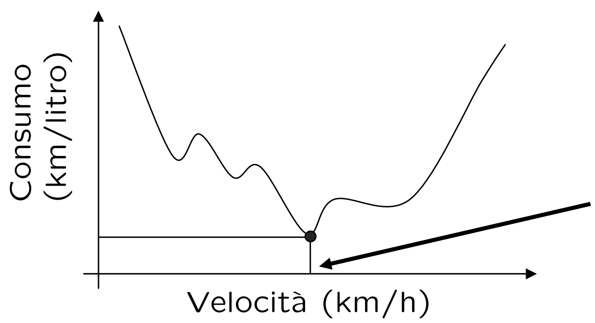
Il problema di ottimizzazione

Cos'è l'ottimizzazione

Ottimizzazione = assegnare ad alcune variabili il miglior valore possibile secondo un criterio assegnato

Esempio:

Date le caratteristiche di un certo autoveicolo, qual'è **la migliore** velocità da tenere per consumare meno carburante possibile ?



velocità ottimale = 80 km/h

Problema di ottimizzazione

Problema di ottimizzazione:

$$\boxed{\min_x f(x)} \qquad \left(\max_x f(x) \right)$$

$\boxed{f(x)}$ MODELLA il nostro obiettivo:

- Consumi
- Costi
- Benefici
- ...

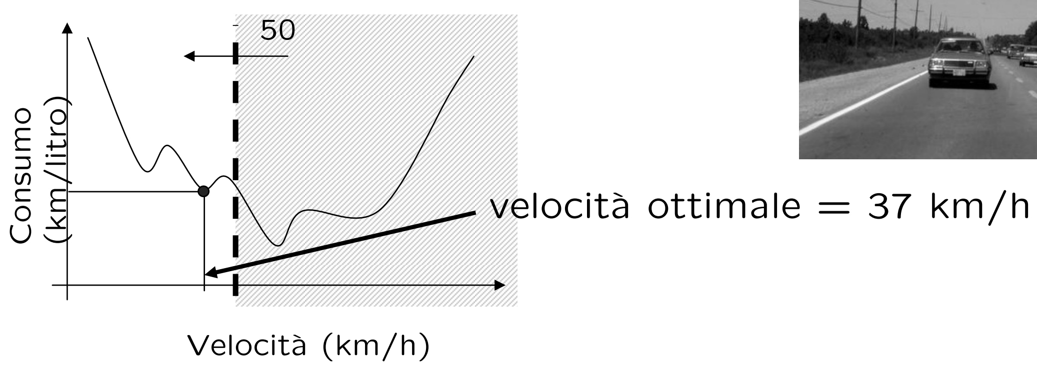
$\boxed{x}$ è la VARIABILE di decisione:

- Velocità
- Vendite
- N. pezzi
- ...

Vincoli (*constraints*)

In genere la variable di decisione x non è completamente libera, ma deve soddisfare alcune condizioni

Es.: La velocità deve essere minore di 50 km/h



Programmazione matematica

$$\boxed{\begin{array}{ll} \min\limits_x & f(x) \\ \text{s.t.} & g(x) \leq 0 \end{array}}$$

$$\left(\begin{array}{ll} \max\limits_x & f(x) \\ \text{s.t.} & g(x) \leq 0 \end{array}\right)$$

$x \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad g(x) = \begin{bmatrix} g_1(x_1, \ldots, x_n) \\ g_2(x_1, \ldots, x_n) \\ \vdots \\ g_m(x_1, \ldots, x_n) \end{bmatrix}$$

In generale è un problema difficile da risolvere !

$\Rightarrow$ **si utilizzano strumenti software**

Software di ottimizzazione

- Rassegna dei risolutori più diffusi, suddivisi per tipologia di problema:

  http://www-fp.mcs.anl.gov/otc/Guide/SoftwareGuide/

- Network Enabled Optimization Server (NEOS) per la risoluzione remota di problemi di ottimizzazione:

  http://neos.mcs.anl.gov/neos/solvers/

- Confronto fra risolutori per vari problemi benchmark:

  http://plato.la.asu.edu/bench.html

- Buon software open-source

  http://www.coin-or.org/

Programmazione lineare

Problemi di tipo LP (**linear programming**)

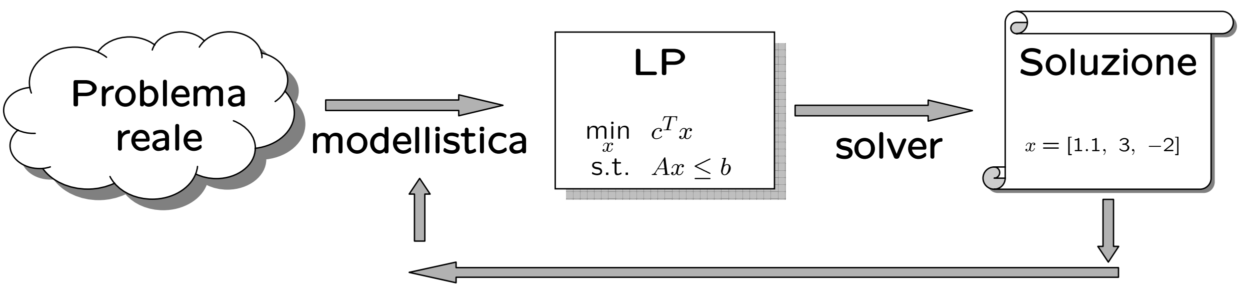
$$\min_x \; \boxed{c^T x} \longrightarrow \text{funzione lineare}$$
$$\text{s.t.} \; \boxed{Ax \leq b} \longrightarrow \text{vincoli lineari}$$

$x \in \mathbb{R}^n$, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$

Per problemi di tipo LP esistono risolutori molto efficienti
(vedi più avanti ...)

Ma a che cosa può servire ?

Decisioni mediante LP

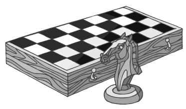


MODELLISTICA: (**modeling**) Definisce un problema di ottimizzazione dall'informazione che abbiamo dal mondo reale

RISOLUTORE: (**solver**) Risolve il problema di ottimizzazione, fornendo una soluzione ottima e ammissibile

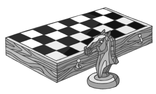
Esempio: Produzione di scacchiere

- Un fabbricante produce due tipi diversi di scacchiere. La più piccola richiede 3 ore di lavoro al tornio, la più grande richiede 2 ore, in quanto meno complessa.
- Si hanno a disposizione 4 torni e altrettanti operai specializzati, ognuno dei quali lavora 40 ore alla settimana.
- La scacchiera più piccola richiede 1 kg di legno, la più grande 3 kg. Il legno però scarseggia e se ne possono avere soltanto 200 kg alla settimana.
- La scacchiera più grande produce un profitto di 20€, la più piccola 5€.
- Problema: quante scacchiere e di quale tipo produrre ogni settimana in modo da massimizzare il profitto ?

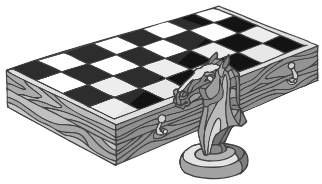


Elementi del problema

Piccole: 3 ore di lavorazione
1 kg di legno
5 € di profitto



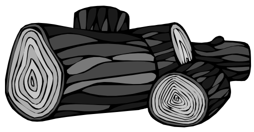
Grandi: 2 ore di lavorazione
3 kg di legno
20 € di profitto



4 operai

40 ore alla settimana

Materiale: soltanto 200 kg alla settimana



Funzione obiettivo

VARIABILI

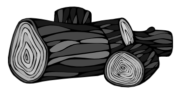
Dobbiamo decidere:
- Prod. scacchiere piccole
- Prod. scacchiere grandi

OBIETTIVO

Vogliamo guadagnare ... max il profitto ! €

Utilizzando le risorse disponibili:
- Tempo
- Legno

VINCOLI



Modello matematico

x_s = n. scacchiere piccole da produrre ≥ 0
x_b = n. scacchiere grandi da produrre ≥ 0

VARIABILI

profitto $= 5{*}x_s + 20{*}x_b$
ore di lavoro $= 3{*}x_s + 2{*}x_b$
legno utilizzato $= 1{*}x_s + 3{*}x_b$

ESPRESSIONI LINEARI

Problema LP

max profitto
- ore di lavoro $\leq 4{*}40$
- legno utilizzato ≤ 200
- $x_s\,,\ x_b \geq 0$

Tentativo di soluzione (euristica)

Soluzioni ammissibili → Soddisfano i vincoli

↓

Profitto

x_s	x_b	Ore	Legno	Ok ?	Profitto
0	0	0	0	Si	0
10	10	50	40	Si	250
50	20	~~190~~	110	No	650
12	62	160	198	Si	1300

È questa la scelta migliore ?

Troveremo la soluzione migliore risolvendo un problema LP

Programmazione Lineare (*Linear Programming, LP*)

Espressioni e disuguaglianze lineari

Espressione lineare

$$a_1x_1 + a_2x_2 + \ldots + a_Nx_N \qquad \text{ovvero:} \qquad \sum_{j=1}^{N} a_jx_j$$

Equazione/disuguaglianza lineare:

$$\sum_{j=1}^{N} a_jx_j \ \{\geq\leq=\}\ b$$

Es.: ore di lavoro = 3*x_s + 2*x_s $\leq$ 4*40

Nota: x = **variabili di decisione (da scegliere)**
a,b = **dati del problema (assegnati)**

Programmazione lineare

George Dantzig
(1914 - 2005)

minimizza o massimizza $\displaystyle\sum_{j=1}^{N} c_jx_j$ (funzione obiettivo)

soggetto a $\displaystyle\sum_{j=1}^{N} a_{ij}x_j \leq b_i$ per $i = 1 \ldots M_1$

$$\sum_{j=1}^{N} a_{ij}x_j \geq b_i \quad \text{per } i = M_1 + 1 \ldots M_2$$

$$\sum_{j=1}^{N} a_{ij}x_j = b_i \quad \text{per } i = M_2 + 1 \ldots M_3$$

$$x_j \geq 0 \quad \text{per } j = 1 \ldots N$$

Programmazione lineare

Trasformazione da max a min:

$$\max_x c^T x = -\left(\min_x -c^T x\right)$$

Cambio di verso disuguaglianze:

$$\sum_{j=1}^{N} a_{ij}x_j \geq b_i \quad \Longrightarrow \quad \sum_{j=1}^{N} -a_{ij}x_j \leq -b_i$$

È sempre possibile formulare problemi LP utilizzando "**min**" e disuguaglianze "$\leq$"

Variabili di scarto (*slack*) e di surplus

Alcuni risolutori accettano problemi LP soltanto nella forma:

$$\begin{array}{ll} \min_x & c^T x \\ \text{s.t.} & \boxed{Ax = b} \\ & x \geq 0 \end{array}$$

solo vincoli di uguaglianza

Variabili di scarto (*slack*)

$$\sum_{j=1}^{N} a_j x_j \leq b \quad \Longrightarrow \quad \sum_{j=1}^{N} a_j x_j + s = b, \;\; s \geq 0$$

Variabili di surplus

$$\sum_{j=1}^{N} a_j x_j \geq b \quad \Longrightarrow \quad \sum_{j=1}^{N} a_j x_j - s = b, \;\; s \geq 0$$

Programmazione lineare

LP in forma "standard":

$$\boxed{\begin{aligned} \min_x \quad & c^T x \\ \text{s.t.} \quad & Ax \le b \\ & Hx = h \end{aligned}}$$

$x \in \mathbb{R}^N$

$A \in \mathbb{R}^{M_2 * N}$

$b \in \mathbb{R}^{M_2}$

$H \in \mathbb{R}^{(M_3 - M_2) * N}$

$h \in \mathbb{R}^{(M_3 - M_2)}$

Questa è la forma standard per la maggior parte dei risolutori (Matlab, Cplex, Xpress-MP, NAG, GLPK, ...)

Alcuni risolutori sottindendono che le variabili $x \ge 0$

Vincoli (*Constraints*)

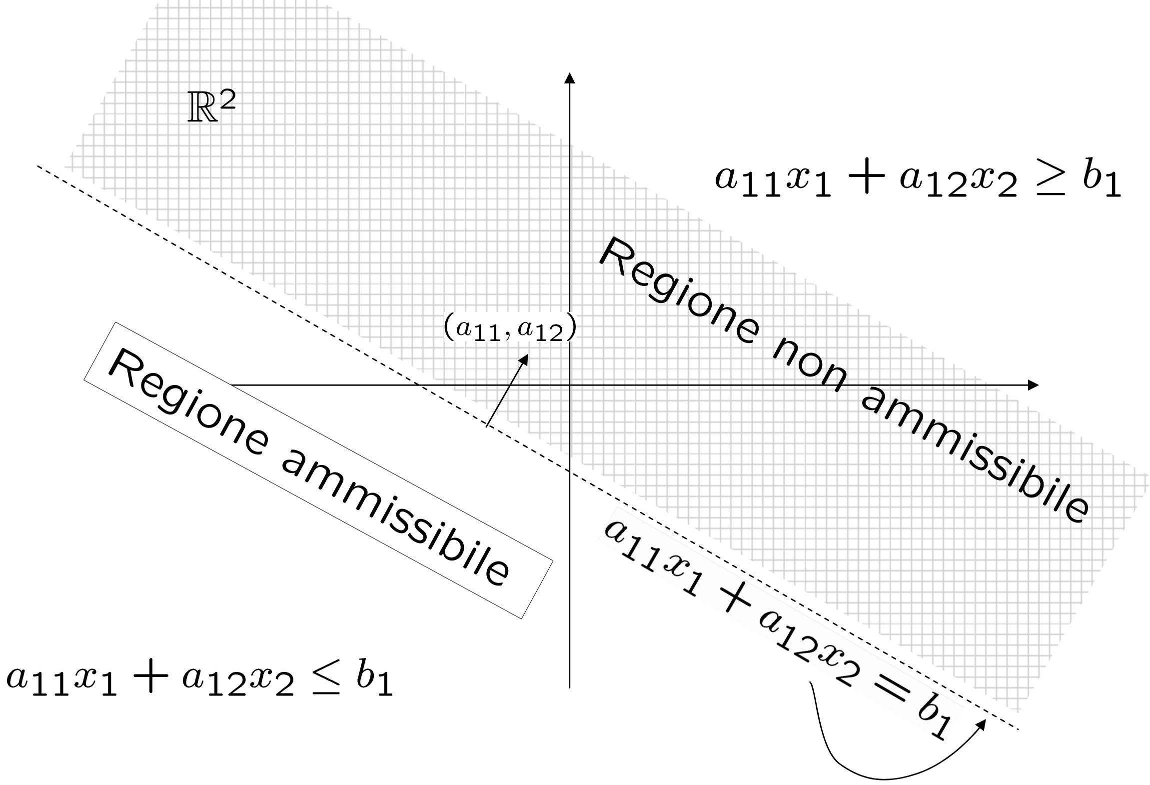
Definiscono la regione ammissibile per le variabili
(= dove occorre "cercare")

$$\boxed{\sum_{j=1}^{N} a_{ij} x_j \le b_i} \quad \text{per } i = 1 \ldots M_1$$

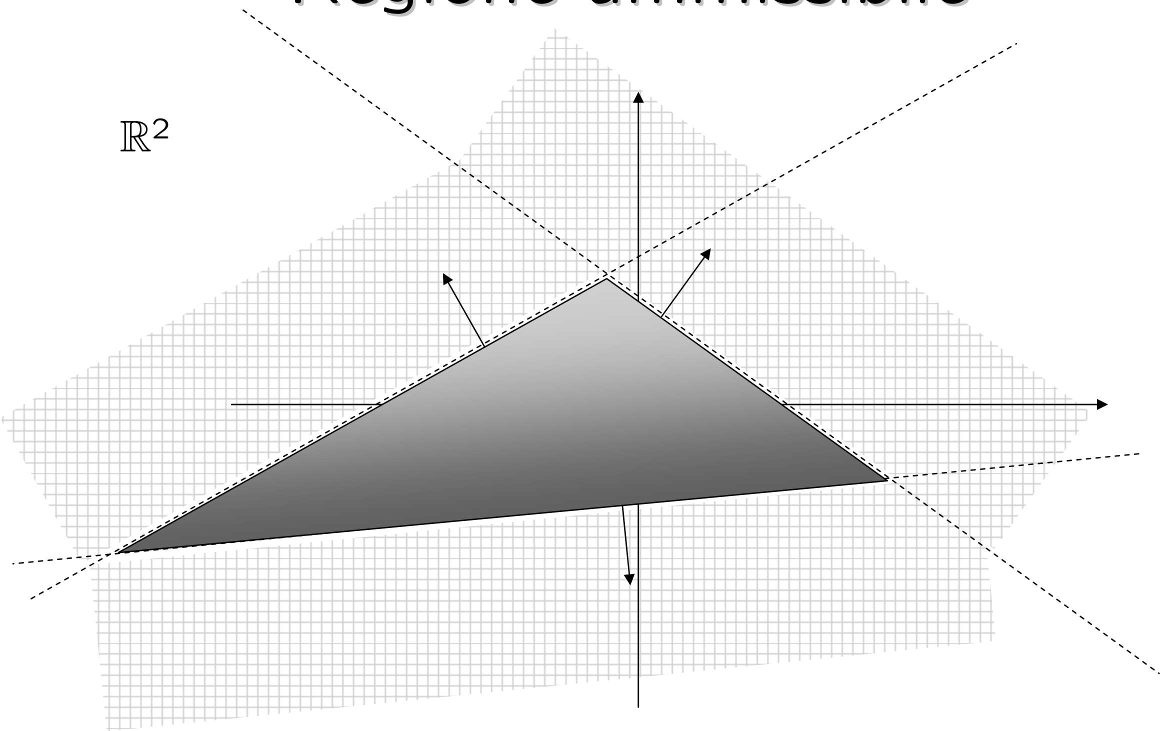
Ciascun vincolo lineare
definisce un semispazio in $\mathbb{R}^N$

Regione ammissibile = intersezione di semispazi
(= poliedro)

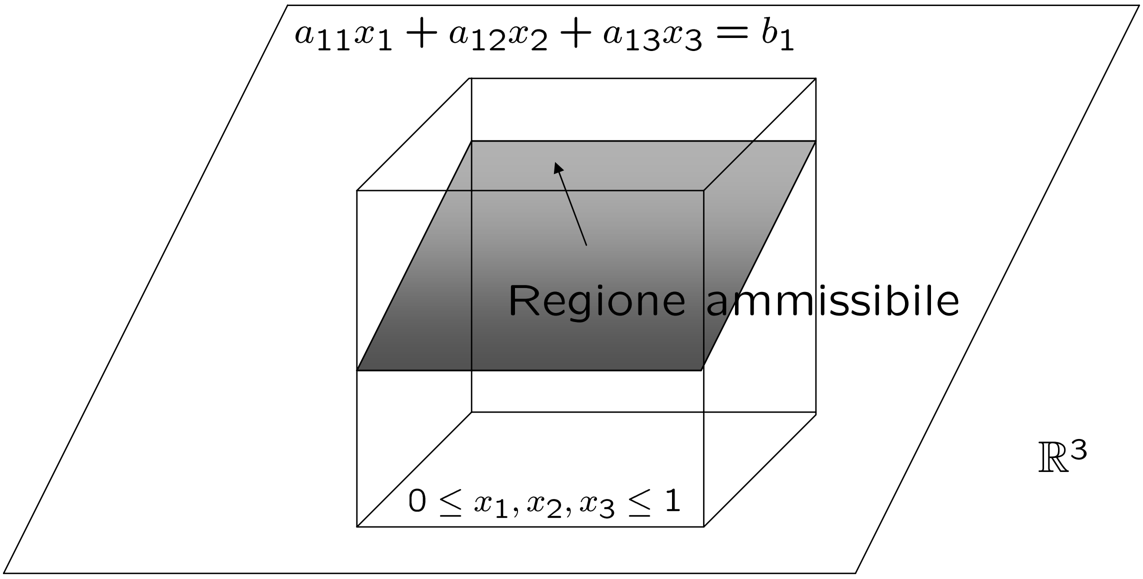
Vincolo lineare



Regione ammissibile



Vincolo di uguaglianza



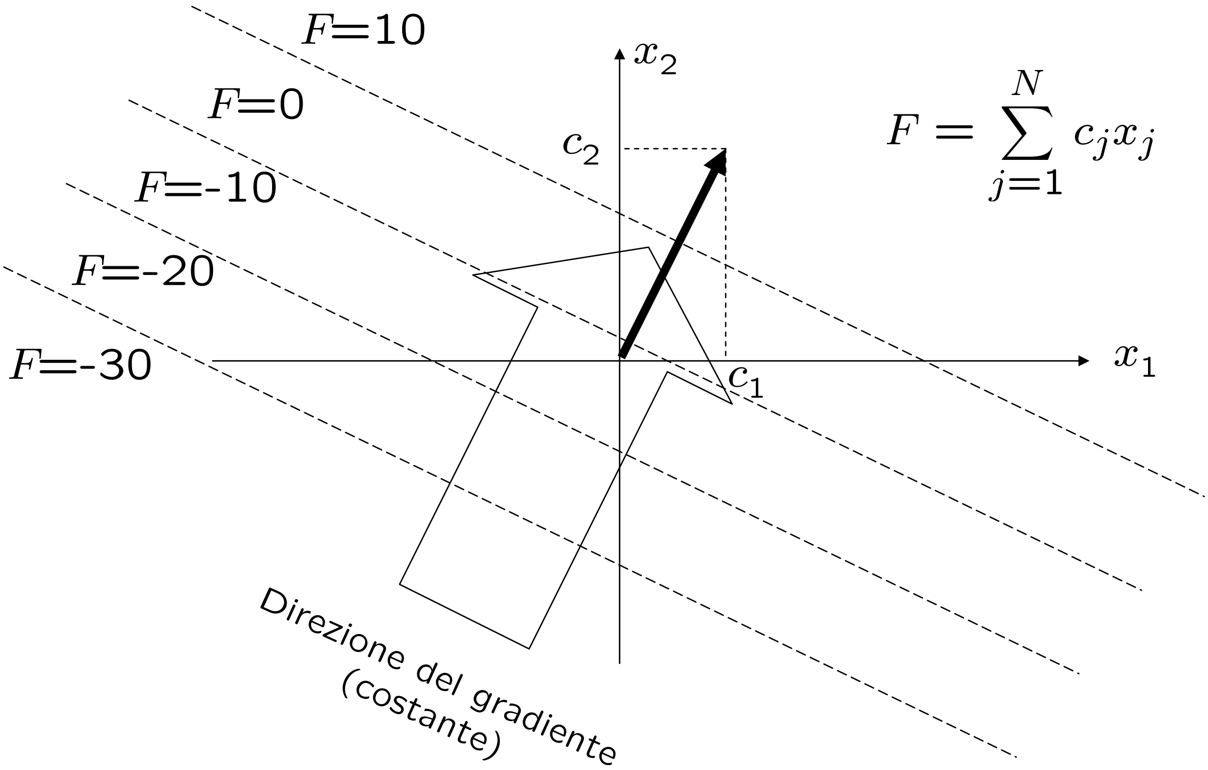
Vincoli di uguaglianza

Eliminazione di variabile:

$$\sum_{j=1}^{N} a_j x_j = b \quad \xRightarrow[a_1 \neq 0]{} \quad x_1 = \frac{b}{a_1} - \sum_{j=2}^{N} \frac{a_j}{a_1} x_j$$

<u>NOTA</u>:
Il risolutore effettua automaticamente la sostituzione !

Funzione obiettivo



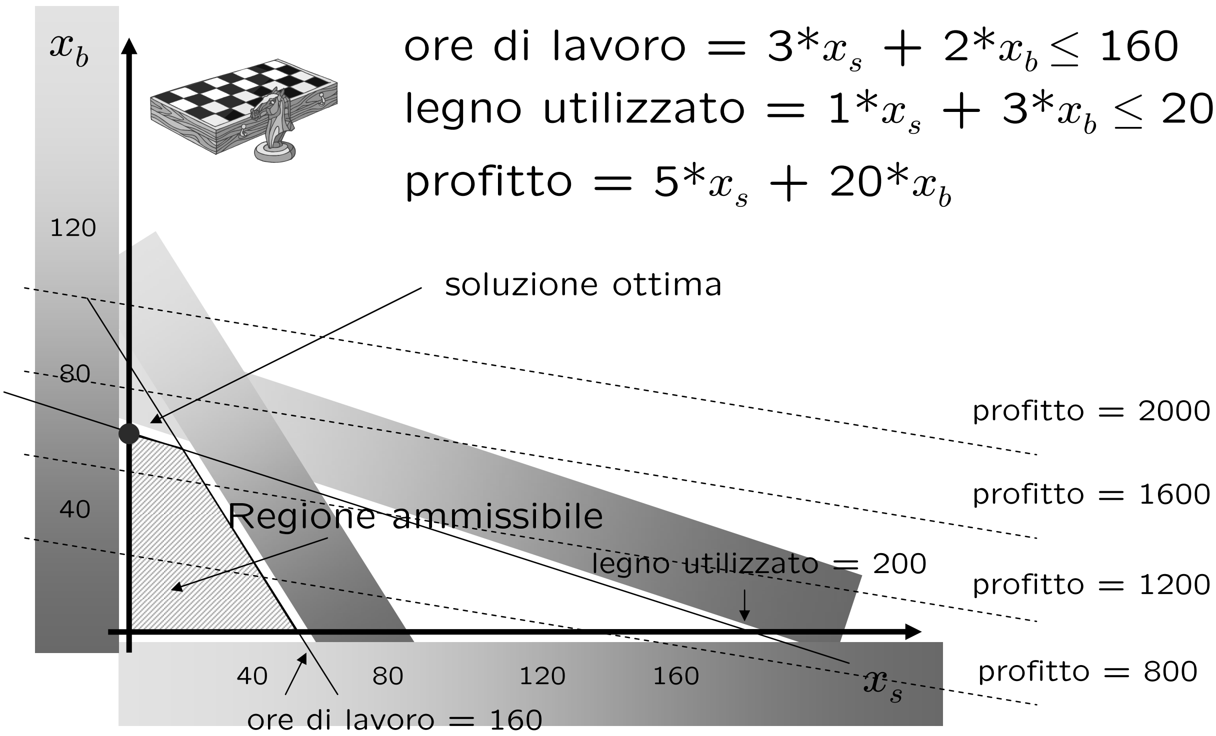
$$F = \sum_{j=1}^{N} c_j x_j$$

Interpretazione geometrica

ore di lavoro = $3^*x_s + 2^*x_b \leq 160$

legno utilizzato = $1^*x_s + 3^*x_b \leq 200$

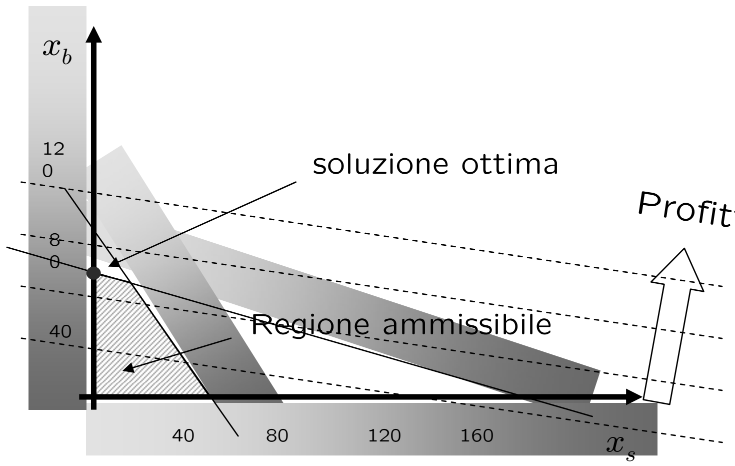
profitto = $5^*x_s + 20^*x_b$



Soluzione ottima

È la soluzione "migliore" fra tutte quelle compatibili con i vincoli:

Per qualsiasi altro y tale che $Ay \le b$ e $Hy = h$ si ha che $c^T y \ge c^T x$



$$
\begin{array}{ll}
\min\limits_{x} & c^T x \\
\text{s.t.} & Ax \le b \\
 & Hx = h
\end{array}
$$

Modellistica e risoluzione mediante il linguaggio MOSEL

Linguaggi di modellistica

- **MOSEL**, associato al pacchetto commerciale Xpress-MP
- **OPL** (Optimization Programming Language), associato al pacchetto commerciale Ilog-CPLEX
- **AMPL** (A Modeling Language for Mathematical Programming) è il più diffuso linguaggio di modellistica, supporta diversi risolutori
- **GAMS** (General Algebraic Modeling System) è uno dei primi linguaggio di modellistica
- **LINGO**, linguaggio di modellistica della Lindo Systems Inc.
- **GNU MathProg**, un subset di AMPL associato al pacchetto *free* GLPK (GNU Linear Programming Kit)
- **FLOPC++** linguaggio di modellistica *open source* implementato come C++ class library

---

- **MPS** (Mathematical Programming System), è invece un formato di file standard per rappresentare problemi di ottimizzazione

MOSEL

Il linguaggio di modellistica serve unicamente ad agevolare la stesura e la manutenzione del modello di ottimizzazione, interponendosi fra chi formula il problema e il risolutore numerico

MOSEL è un linguaggio per la modellistica e la risoluzione di problemi di ottimizzazione

- È un facile linguaggio di “programmazione”
- Ha una buona interfaccia grafica

Nota bene:

LA MODELLISTICA NON DIPENDE DAL RISOLUTORE !

Produzione di scacchiere - MOSEL

```
model Chess
uses "mmxprs"

declarations
       xs,xb: mpvar
end-declarations

Profit := 5*xs + 20*xb
Wood := 1*xs + 3*xb <= 200
Work := 3*xs + 2*xb <= 160

maximize(Profit)

writeln("LP Solution:")
writeln(" Objective: ", getobjval)
writeln(" Make: ",getsol(xs), " small sets")
writeln(" Make: ",getsol(xb), " big sets")

end-model
```

Xpress-IVE

(Apri il file `\examples\mosel\book\Intro\chess.mos`)

Formulazione in MOSEL

```
model Chess
uses "mmxprs"

declarations
       xs,xb: mpvar
end-declarations

Profit := 5*xs + 20*xb
Wood := 1*xs + 3*xb <= 200
Work := 3*xs + 2*xb <= 160

maximize(Profit)

writeln("LP Solution:")
writeln(" Objective: ", getobjval)
writeln(" Make: ",getsol(xs), " small sets")
writeln(" Make: ",getsol(xb), " big sets")

end-model
```

Dichiarazione delle variabili

Funzione obiettivo

Vincoli

Risoluzione del problema

Output

Matlab Optimization Toolbox - LP

linprog

Purpose Solve a linear programming problem

$$\min_x f^T x \quad \text{such that} \quad \begin{aligned} A \cdot x &\le b \\ Aeq \cdot x &= beq \\ lb \le x &\le ub \end{aligned}$$

where f, x, b, beq, lb, and ub are vectors and A and Aeq are matrices.

Syntax

```
x = linprog(f,A,b,Aeq,beq)
x = linprog(f,A,b,Aeq,beq,lb,ub)
x = linprog(f,A,b,Aeq,beq,lb,ub,x0)
x = linprog(f,A,b,Aeq,beq,lb,ub,x0,options)
[x,fval] = linprog(...)
[x,fval,exitflag] = linprog(...)
[x,fval,exitflag,output] = linprog(...)
[x,fval,exitflag,output,lambda] = linprog(...)
```

Description `linprog` solves linear programming problems.

Esistono risolutori LP molto più efficienti !

Produzione di scacchiere - Matlab

max profitto $= 5 * x_s + 20 * x_b$

legno utilizzato $= 1 * x_s + 3 * x_b \le 200$

ore di lavoro $= 3 * x_s + 2 * x_b \le 160$

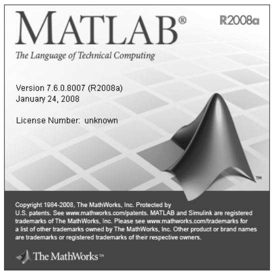
```
>> A=[1 3;3 2];
>> b=[200;160];
>> c=[5 20];
>> [x,fval]=linprog(-c,A,b,[],[],[0;0])
Optimization terminated successfully.

x =

    0.0000
   66.6667


fval =

 -1.3333e+003

>>
```

Hybrid Toolbox per Matlab

(Bemporad, 2003-2008)

Features:

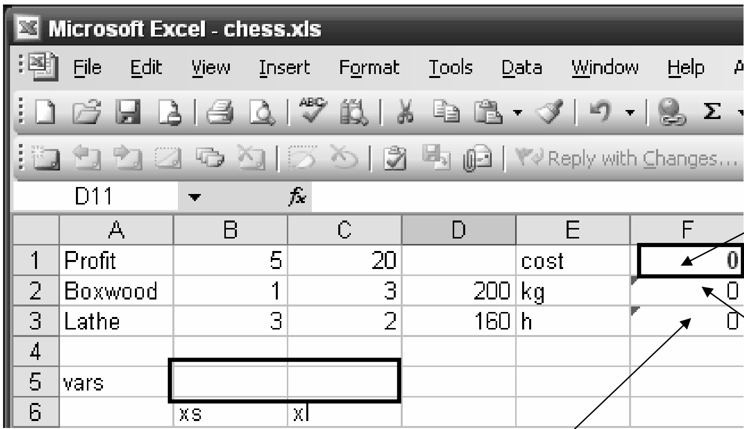
- Hybrid model (MLD and PWA) design, simulation, verification
- Control design for linear systems w/ constraints and hybrid systems (on-line optimization via QP/MILP/MIQP)
- Explicit control (via multiparametric programming)
- C-code generation
- Simulink
- *Matlab interfaces to various solvers for LP, QP, MILP, MIQP*



http://www.dii.unisi.it/hybrid/toolbox

1700+ downloads since Oct 2004

Produzione di scacchiere - Excel

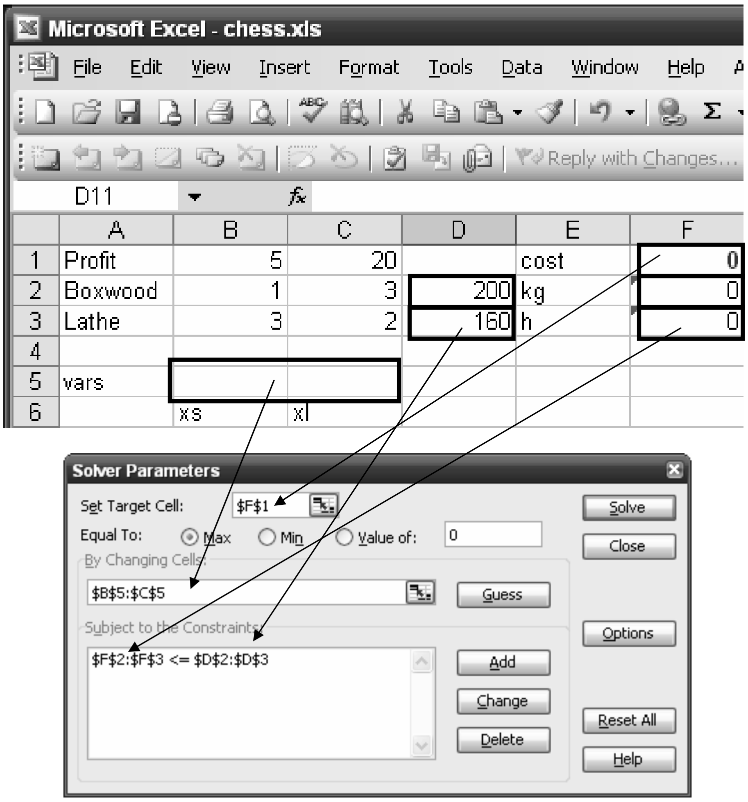


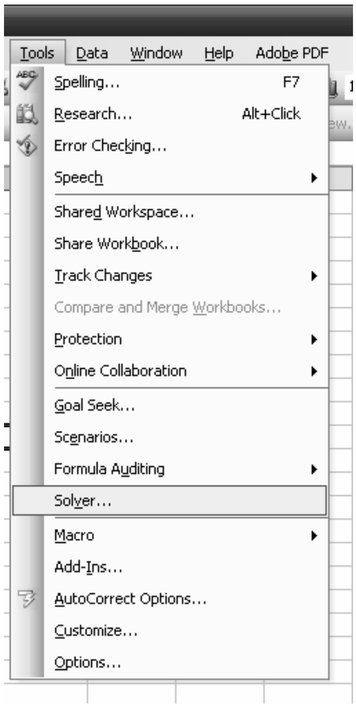
=sumproduct(B1:C1,B5:C5)

=sumproduct(B2:C2,B5:C5)

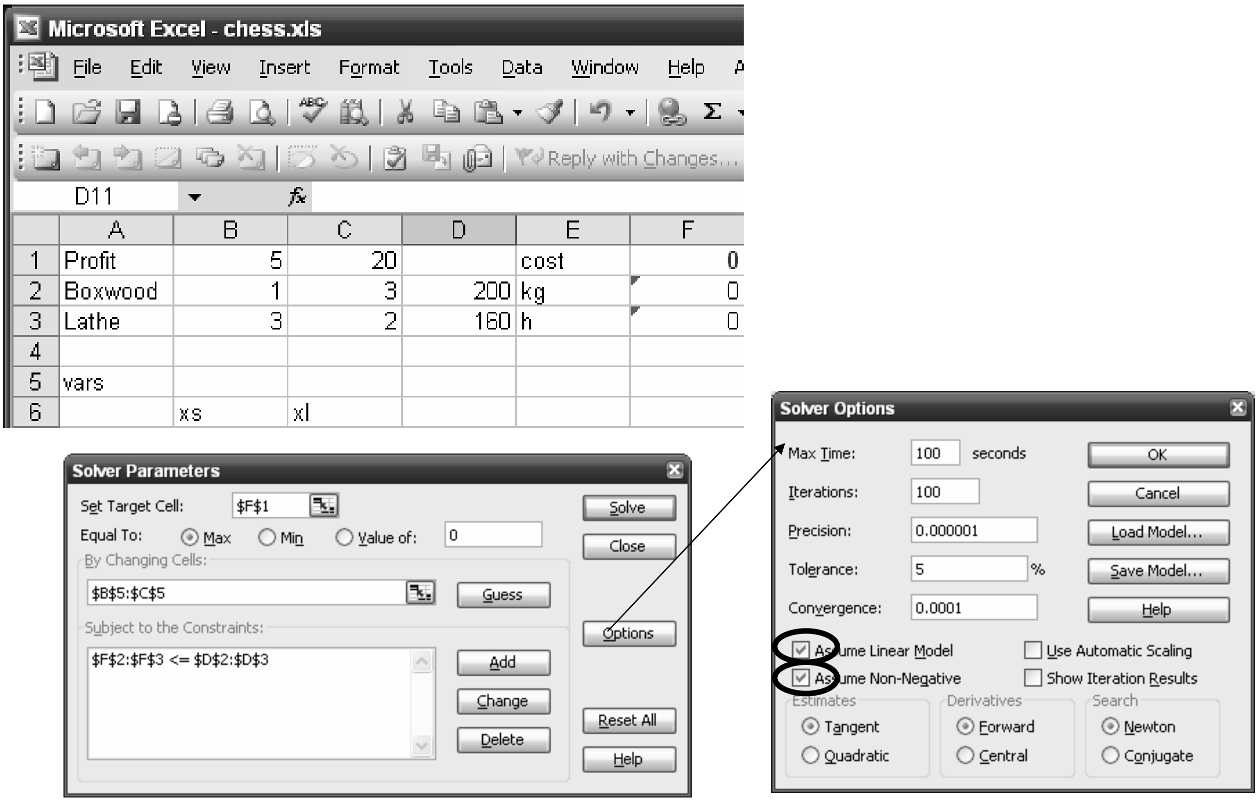
=sumproduct(B3:C3,B5:C5)

Produzione di scacchiere - Excel

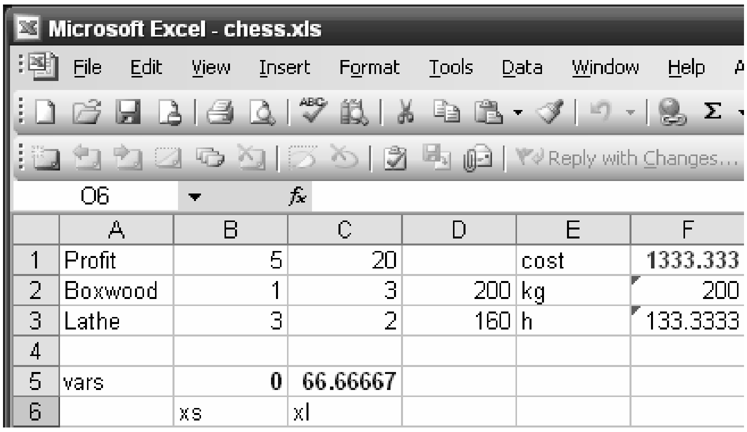


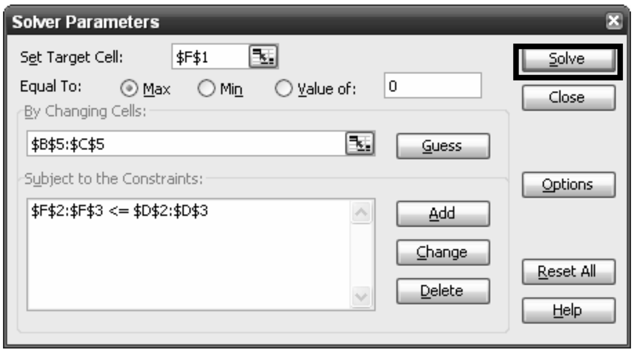


Produzione di scacchiere - Excel



Produzione di scacchiere - Excel





Risolutori di problemi LP

BPMPD - linear programming.
CLP – open source linear programming (www.coin-or.org)
CPLEX - linear and integer programming.
C-WHIZ - linear programming models.
Excel and Quattro Pro Solvers - spreadsheet-based linear, integer and nonlinear programming.
FortMP - linear and mixed integer quadratic programming.
GAUSS - matrix programming language.
GLPK – linear and integer programming
HS/LP Linear Optimizer - linear programming with OMNI language.
KORBX - linear programming.
LAMPS - linear and mixed-integer programming.
LINDO Callable Library - linear, mixed-integer and quadratic programming.
LINGO - linear, integer, nonlinear programming with modeling language.
LOQO - Linear programming, unconstrained and constrained nonlinear optimization.
LP88 and BLP88 - linear programming.
MINOS - linear programming and nonlinear optimization.
MOSEK - linear programming and convex optimization.
MPSIII - linear and mixed integer programming (includes OML, WHIZARD, and DATAFORM).
OML - linear and mixed-integer programming.
OSL - linear, quadratic and mixed-integer programming.
PCx - linear programming with a primal-dual interior-point method.
PORT 3 - minimization, least squares, etc.
PROC LP - linear and integer programming.
QPOPT - linear and quadratic problems.
SQOPT - large-scale linear and convex quadratic programming problems.
What'sBest - linear and mixed integer programming.
WHIZARD - linear and mixed-integer programming.
XPRESS-MP from Dash Associates - linear and integer programming.
....

Considerazioni generali sulla modellistica e l'ottimizzazione

Modellistica e ottimizzazione

Motivazione: apportare dei benefici ad un'organizzazione (banca, industria, finanziaria, ...)

Modellistica = trasformare un problema reale in forma matematica

Perché ? Per risolvere il problema in maniera razionale !

Ottimizzazione = trova una soluzione al problema (ammissibile e ottimale)

Questo corso verte principalmente sulla MODELLISTICA
(come ottenere modelli chiari, accurati e mantenibili)

Non ci occuperemo degli ALGORITMI DI OTTIMIZZAZIONE,
cioè dello strumento necessario per risolvere il modello all'ottimo
(es: metodo del simplesso, metodo punto interno, ecc.)

Modellistica

I modelli non rappresentano la realtà in maniera esatta

Si fanno delle **ipotesi di lavoro**

Es.:

- Sempre 160 ore di lavoro
- Sempre esattamente 200 kg
- Vendiamo tutte le scacchiere
- I prezzi sono costanti

Esistono infiniti modelli per un dato problema !

La modellistica è un'arte

(non esiste una "teoria" univoca)

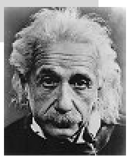
Modellistica

Esiste un **compromesso** fra rappresentatività e semplicità del modello:

- Il modello deve essere **rappresentativo**, deve cioè catturare le caratteristiche principali del problema

- Il modello deve essere **semplice** abbastanza per poter essere risolto in maniera efficiente

"Make everything as simple as possible, but not simpler."
— Albert Einstein



Dati e modelli

I modelli dipendono dai dati:

Prezzi
Quantità
Necessità
...

Tipo di dato $\longrightarrow$ Che cosa descrive ?

Qualità del dato $\longrightarrow$ Quanto è esatto ?

TIPI E QUALITÀ DIPENDONO DAL PROBLEMA

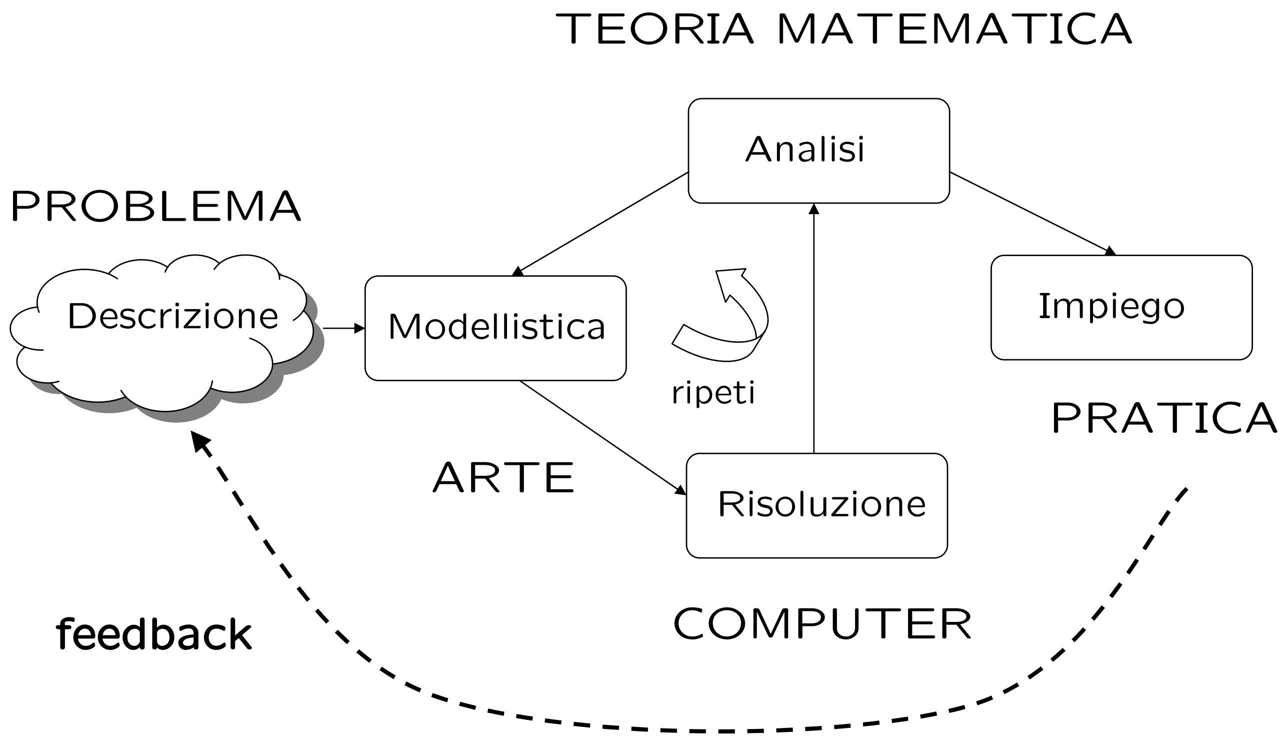
Dati e modelli

I modelli unificano parti diverse di una stessa organizzazione.

Richiedono pertanto dati da diversi reparti:

- Reparto vendite (**sales department**):
  Costi di vendita ?
- Reparto produzione (**production department**):
  Costi di produzione ?
- Reparto marketing:
  Predizioni di vendita ?

Progetto di ottimizzazione



Benefici

Benefici di un progetto di modellistica e ottimizzazione:

- Decisioni razionali basati sull'informazione
- Anche se non applicate, forniscono un buon "consiglio"
- Capire il problema tramite il modello (es: dimostrare che alcune parti di una organizzazione funzionano bene/non funzionano)
- Robustezza delle decisioni al variare dei dati
- ...

INCREMENTA PROFITTI E EFFICIENZA

Modellistica di Problemi LP

Programmazione lineare

minimizza o massimizza $\sum_{j=1}^{N} c_j x_j$ (funzione obiettivo)

soggetto a $\sum_{j=1}^{N} a_{ij} x_j \leq b_i$ per $i = 1 \ldots M_1$

$\sum_{j=1}^{N} a_{ij} x_j \geq b_i$ per $i = M_1 + 1 \ldots M_2$

$\sum_{j=1}^{N} a_{ij} x_j = b_i$ per $i = M_2 + 1 \ldots M_3$

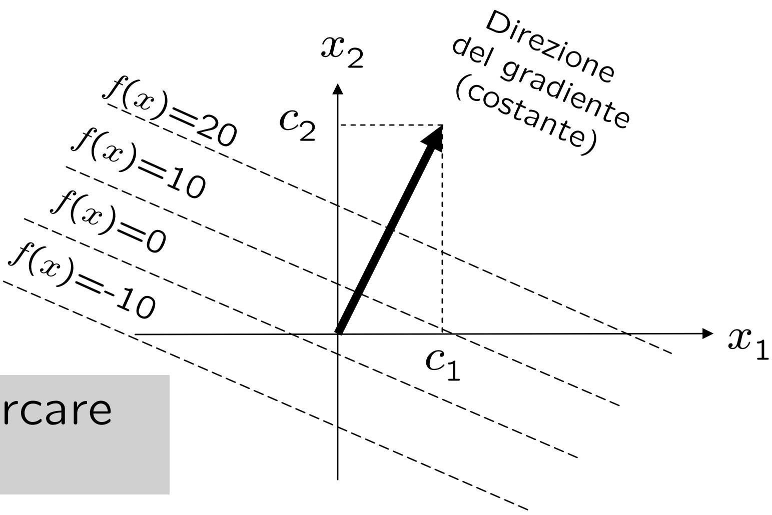
$x_j \geq 0$ per $j = 1 \ldots N$

Funzione obiettivo

La funzione obiettivo è una funzione lineare

$$f(x) = \sum_{j=1}^{N} c_j x_j$$

Il vettore $[c_1\ c_2\ \ldots\ c_N]$ è il **gradiente** in $\mathbb{R}^N$ della funzione



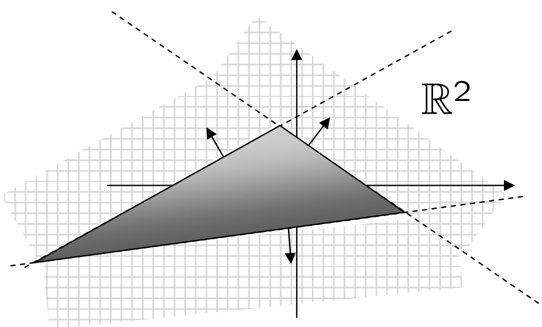
Ci dice **"come"** cercare la soluzione ottima

Vincoli (*Constraints*)

Definiscono la regione ammissibile per le variabili

$$\sum_{j=1}^{N} a_{ij} x_j \leq b_i \quad \text{per } i = 1 \ldots M_1$$

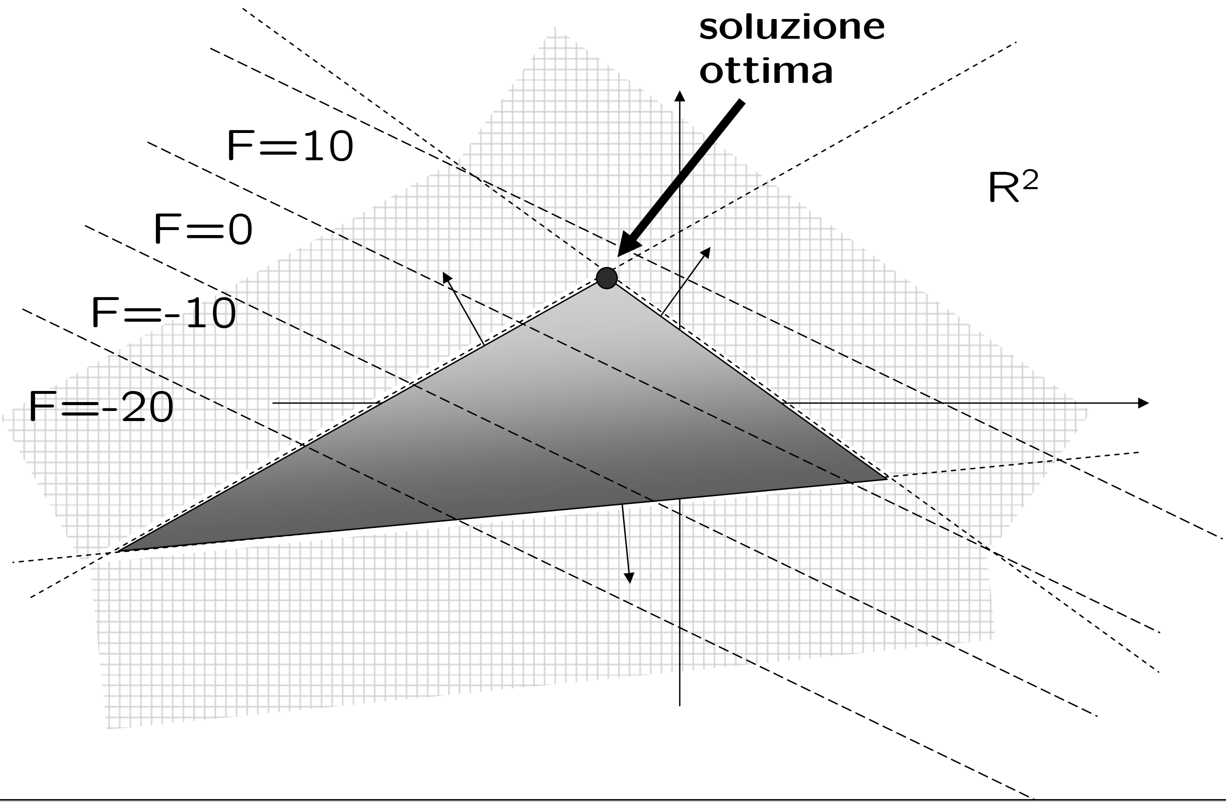
Ciascun vincolo lineare definisce un semispazio in $\mathbb{R}^N$



La regione ammissibile è un poliedro

Ci dicono **"dove"** cercare la soluzione ottima

Programmazione Lineare (LP)



Modellistica

"Formulare in termini matematici un problema reale"

Occorre definire:

- Chi sono le **variabili** da decidere
- Qual'è la **funzione obiettivo** che ci interessa minimizzare o massimizzare
- Quali **vincoli** si hanno nella scelta

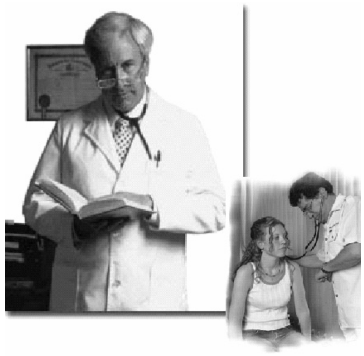
Non esiste un modo univoco di modellare un dato problema. Non esiste una "teoria sistematica" della modellistica!

Vincoli (*constraints*)

Definiscono l'insieme ammissibile (**feasible set**), cioè dove cercare la soluzione

I vincoli definiscono **relazioni** fra le variabili decisionali

Metodo di modellistica suggerito:



Disaggregare le relazioni e restrizioni presenti nel problema reale in sottinsiemi di vincoli che sappiamo come modellare

Si conoscono infatti diversi tipi di vincoli ...

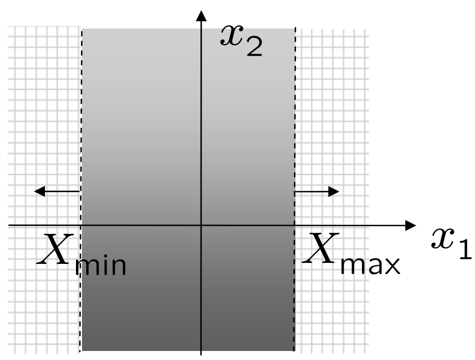
1 - Upper & Lower Bounds

Sono i vincoli più semplici ...

Rappresentano limiti naturali sui valori che certe variabili possono assumere

Esempi: Dobbiamo vendere almeno 100 unità di prodotto i
Non possiamo comprare più di 50 automobili

$$x_i \leq X_{\text{max}}$$
$$x_i \geq X_{\text{min}}$$



1 - Upper & Lower Bounds

Attenzione ai vincoli di **non negatività** !!!
Diversi risolutori assumono tacitamente che tutte le variabili sono positive o nulle.

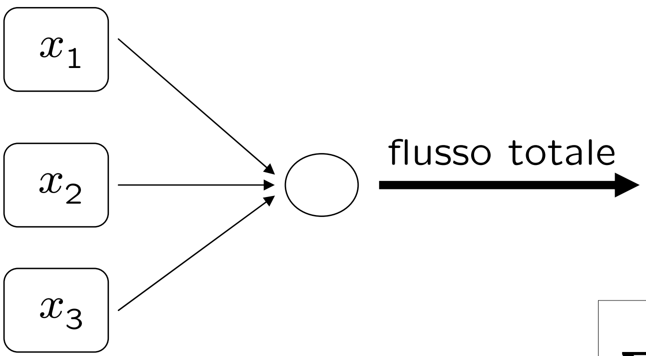
<u>Esempio</u>: In MOSEL tutte le mpvar sono di default non negative

```
declarations
     x: mpvar
end-declarations

x is_free
```

Le variabili "libere" devono essere esplicitamente specificate

2 – Flusso (*flow constraints*)

Il flusso di un certo bene è composto da vari flussi



x_1, x_2, x_3 = variabili di ottimizzazione

Vincolo sul flusso totale:

$$\sum x_i \leq F_{\text{max}}$$

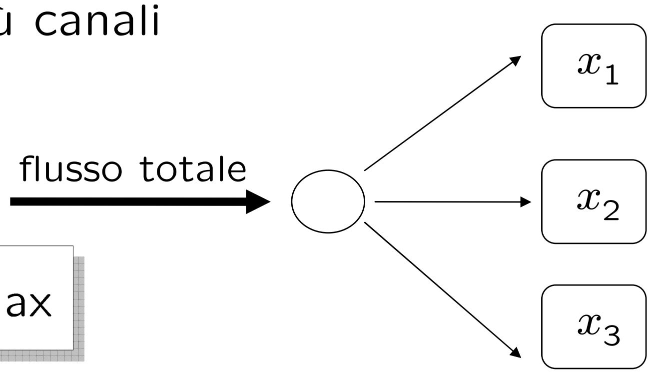
Esempio:

Abbiamo N stabilimenti a disposizione e dobbiamo produrre un minimo di 44 autovetture alla settimana

$$\sum_{i=1}^{N} \text{prod}_i \geq 44$$

2 – Flusso (*flow constraints*)

In alternativa, il flusso di un certo bene può essere suddiviso in più canali



$$\boxed{\sum x_i \leq F_{\max}}$$

Esempio:

Abbiamo un deposito di 10 M€ con il quale liquidare le spese sostenute da ciascuno di 3 stabilimenti

$$\sum_{i=1}^{3} \text{spese}_i \leq 1000$$

3 – Risorse

Le risorse a disposizione non sono infinite ma vanno suddivise fra varie $\boxed{\text{attività}}$

= variabili di ottimizzazione

In genere i vincoli sulle risorse vengono definiti tramite una tabella di coefficienti R_{ji}

$$\sum_{i=1}^{N} R_{ji} x_i \leq \max \text{ risorsa}_j$$

R_{ji} = quanta risorsa di tipo j viene consumata dall'attività i-esima

<u>Esempio</u>: quanto legno serve per costruire una scacchiera piccola ?

4 – Qualità

Esempio:

Dobbiamo fare un investimento in azioni di tipo Generali, Tiscali e Bulgari. Ogni azione ha un suo fattore di rischio. Vogliamo mantenere il rischio medio dell'investimento sotto il 50%.

	GENERALI	TISCALI	BVLGARI
Rischio	5%	70%	40%

rG : Rischio azioni Generali
rT : Rischio azioni Tiscali
rB : Rischio azioni Bulgari

4 – Qualità

Variabili:

sG = k€ investiti in Generali
sT = k€ investiti in Tiscali
sB = k€ investiti in Bulgari

Importo totale dell'investimento = sG + sT + sB

Rischio medio:

$$\frac{sG*rG + sT*rT + sB*rB}{sG + sT + sB} \leq 0.5$$

Problema: non è un vincolo lineare !

4 – Qualità

Con una semplice manipolazione, possiamo trasformare il vincolo in un vincolo lineare:

```
sG*rG + sT*rT + sB*rB ≤ 0.5 *(sG + sT + sB)
```

Nota: la trasformazione funziona solo se il denominatore è positivo

Funzione obiettivo e vincoli possono essere spesso trasformati, mediante operazioni matematiche, in forme lineari equivalenti senza che la soluzione ottima venga alterata

5 – Miscelazione (*Blending*)

Sono utili quando dobbiamo esprimere entità costituite da percentuali fissate delle variabili di ottimizzazione

Esempio:

Preparazione di un cocktail. Dobbiamo preparare del Negroni, costituito da: 30% Campari, 40% Martini, 30% Gin

Le quantità Campari, Martini e Gin sono vincolate fra loro per il fatto che devono comparire nel cocktail in proporzioni preassegnate

5 – Miscelazione (*Blending*)

Il vincolo di miscelazione risulta essere:

```
0.3 = Campari / (Campari+Martini+Gin)
0.4 = Martini / (Campari+Martini+Gin)
0.3 = Gin / (Campari+Martini+Gin)
```

Assumendo che `Campari`, `Martini` e `Gin` siano quantità positive, il vincolo può essere riscritto come vincolo lineare:

```
Campari = 0.3*(Campari+Martini+Gin)
Martini   = 0.4*(Campari+Martini+Gin)
Gin = 0.3*(Campari+Martini+Gin)
```

(È simile al vincolo sulla qualità)

5 – Miscelazione (*Blending*)

Nota: potremmo eliminare una variabile (ad es: Gin) per sostituzione (addirittura due variabili):

`Gin = 0.3/0.7*(Campari+Martini)`

```
Campari = 0.3/0.7*(Campari+Martini)
Martini   = 0.4/0.7*(Campari+Martini)
```

Non conviene eliminare variabili !

Ci penserà il risolutore ad eliminarle. Lasciandole nel modello, rendiamo quest'ultimo molto più leggibile (e quindi più facilmente mantenibile).

Variabili di ottimizzazione

Rappresentano le incognite del nostro problema

Esempio: **Quante azioni dobbiamo comprare ?**

Spesso nel modello vengono introdotte variabili ridondanti per esprimere certi vincoli sul modello stesso in maniera più chiara

Esempio: per esprimere il vincolo di flusso $x_1+x_2\leq 1$ possiamo introdurre una nuova variabile $x_3=x_1+x_2$ e porre $x_3\leq 1$

Es: Produzione di leghe metalliche

La ditta Metalli S.p.A. ha ricevuto un ordine di 500 tonnellate di acciaio da utilizzare per la costruzione di una nave. Il metallo deve avere determinate caratteristiche di composizione.

La ditta ha a disposizione sette tipi diversi di materiale grezzo che possono essere utilizzati per la produzione dell'acciaio richiesto.

L'obiettivo è quello di determinare la composizione che minimizza il costo complessivo, nel rispetto dei vincoli imposti sulla composizione dell'acciaio prodotto.

Dati a disposizione

Vincoli sulla qualità dell'acciaio prodotto:

Elemento chimico	Composizione minima %	Composizione massima %
Carbonio (C)	2	3
Rame (Cu)	0.4	0.6
Manganese (Mn)	1.2	1.65

Dati a disposizione

Composizioni %, quantità disponibili, costi per tonnellata

Mat. grezzo	C%	Cu%	Mn%	t	€
Iron 1	2.5	0	1.3	400	200
Iron 2	3	0	0.8	300	250
Iron 3	0	0.3	0	600	150
Copper 1	0	90	0	500	220
Copper 2	0	96	1.2	200	240
Aluminium 1	0	0.4	1.2	300	200
Aluminium 2	0	0.6	0	250	165

Modellistica

"La ditta Metalli S.p.A. ha ricevuto un ordine di 500 tonnellate di acciaio..."

Lower Bound

"Il metallo deve avere determinate caratteristiche di composizione...."

Vincoli sulla qualità

"...che minimizza il costo ..."

Funzione obiettivo

"La ditta ha a disposizione sette tipi diversi di materiale grezzo ..."

Variabili di ottimizzazione

Upper bounds

Mat. grezzo	C%	Cu%	Mn%	t	€
Iron 1	2.5	0	1.3	400	200
Iron 2	3	0	0.8	300	250
Iron 3	0	0.3	0	600	150
Copper 1	0	90	0	500	220
Cooper 2	0	96	1.2	200	240
Aluminium 1	0	0.4	1.2	300	200
Aluminium 2	0	0.6	0	250	165

Variabili: use_j
j=1,2,...7

R_j=quantità max disponibile

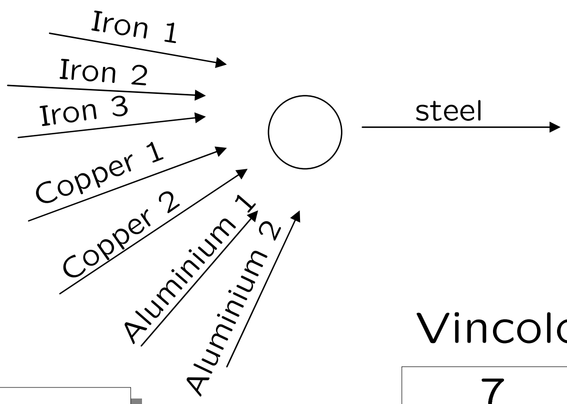
$$\text{use}_j \leq R_j$$

$$\text{use}_j \geq 0$$

Non negatività

Lower bound e flusso

"La ditta Metalli S.p.A. ha ricevuto un ordine per 500 tonnellate di acciaio..."



Lower bound:

$$\text{produce} \geq 500$$

$$\text{use}_j \geq 0$$

Vincolo di flusso:

$$\sum_{j=1}^{7} \text{use}_j = \text{produce}$$

Funzione obiettivo

"...che minimizza il costo..."

Mat. grezzo	C%	Cu%	Mn%	t	€
Iron 1	2.5	0	1.3	400	200
Iron 2	3	0	0.8	300	250
Iron 3	0	0.3	0	600	150
Copper 1	0	90	0	500	220
Cooper 2	0	96	1.2	200	240
Aluminium 1	0	0.4	1.2	300	200
Aluminium 2	0	0.6	0	250	165

C_j

costo per tonnellata di materiale grezzo

Costo totale:

$$\text{Cost} = \sum_{j=1}^{7} C_j \text{use}_j$$

Dati a disposizione

Elemento Chimico	Composizione minima %	Composizione massima %
Carbon (C)	2	3
Copper (Cu)	0.4	0.6
Manganese (Mn)	1.2	1.65

Pmin_i Pmax_i

Vincoli sulla qualità del prodotto

Vincoli sulla qualità del prodotto

	Mat. Grezzo	C% (i=1)	Cu% (i=2)	Mn% (i=3)	t	€
j=1	Iron 1	2.5	0	1.3	...	...
...	...	...	...	...	...	...

P_{ji} = % di metallo i contenuta nel materiale grezzo j (es: 1 t di Iron 1 contiene 25 kg di C)

Quantità di metallo di tipo i

$$\frac{\boxed{\sum_{j=1}^{7} P_{ji}\text{use}_j}}{\boxed{\sum_{j=1}^{7} \text{use}_j}} \geq \boxed{\text{Pmin}_i}$$

Quantità totale di prodotto

% minima di metallo di tipo i contenuta nel prodotto finale

Vincoli sulla qualità del prodotto

Per ogni tipo di metallo ...

$$\boxed{\text{for } i = 1 \ldots 3}$$
$$\sum_{j=1}^{7} P_{ji}\text{use}_j \geq \text{Pmin}_i * \text{produce}$$
$$\sum_{j=1}^{7} P_{ji}\text{use}_j \leq \text{Pmax}_i * \text{produce}$$
end

È un insieme di diseguaglianze di tipo lineare

Modello completo del problema

minimizza

$$\text{Cost} = \sum_{j=1}^{7} C_j\text{use}_j$$

soggetto a

$$\sum_{j=1}^{7} \text{use}_j = \text{produce}$$
$$\text{produce} \geq 500$$
for $j = 1 \ldots 7$
$$\text{use}_j \leq R_j$$
end
for $i = 1 \ldots 3$
$$\sum_{j=1}^{7} P_{ji}\text{use}_j \geq Pmin_i * \text{produce}$$
$$\sum_{j=1}^{7} P_{ji}\text{use}_j \leq Pmax_i * \text{produce}$$
end

Modello MOSEL - Dati

```
declarations
      !Index sets
      COMP = 1..3
      RAW = 1..7
      !Data arrays
      P : array(RAW,COMP) of real
      PMIN,PMAX : array(COMP) of real
      AVAIL : array(RAW) of real
      COST : array(RAW) of real
      DEM : real
end-declarations
```

DATI

I dati devono essere inizializzati

Inizializzazione dei dati

```
!Data initialization
!By hand
P :: [2.5, 0, 1.3,
      3, 0, 0.8,
      0, 0.3, 0,
      0, 90, 0,
      0, 96, 4,
      0, 0.4, 1.2,
      0, 0.6, 0]

PMIN := [2, 0.4, 1.2]
PMAX := [3, 0.6, 1.65]

AVAIL := [400,300,600,500,200,300,250]
COST := [200,250,150,220,240,200,165]
DEM := 500
```

Mat. grezzo	C%	Cu%	Mn%	t	€
Iron 1	2.5	0	1.3	400	200
Iron 2	3	0	0.8	300	250
Iron 3	0	0.3	0	600	150
Copper 1	0	90	0	500	220
Cooper 2	0	96	1.2	200	240
Aluminium 1	0	0.4	1.2	300	200
Aluminium 2	0	0.6	0	250	165

Variabili

```
declarations
      !Decision variables
      use : array(RAW) of mpvar
      produce : mpvar
end-declarations
```

Lower bound

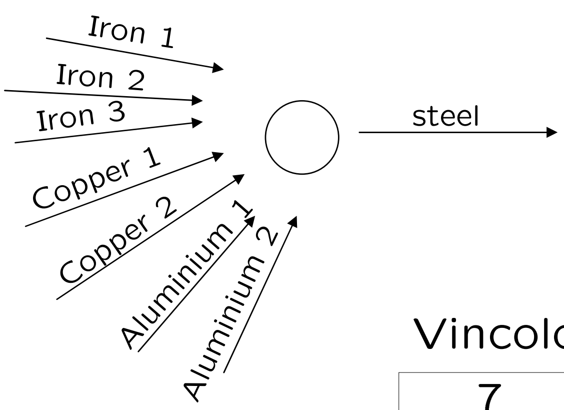
$$\text{produce} \geq 500$$

Non negatività

$$\text{use}_j \geq 0$$

Vincolo di flusso

```
produce >= DEM;
produce =  sum(i in RAW) use(i)
```



Lower bound:

$$\text{produce} \geq 500$$

Vincolo di flusso:

$$\sum_{j=1}^{7} \text{use}_j = \text{produce}$$

Funzione obiettivo

```
!Objective function
Cost := sum(i in RAW) COST(i)*use(i)

minimize(Cost)
```

Costo totale

$$\text{Cost} = \sum_{j=1}^{7} C_j \text{use}_j$$

Risorse

```
!Constraints
use(1) <= AVAIL(1)
use(2) <= AVAIL(2)
use(3) <= AVAIL(3)
use(4) <= AVAIL(4)
use(5) <= AVAIL(5)
use(6) <= AVAIL(6)
use(7) <= AVAIL(7)
```

$$\sum_{j=1}^{7} \text{use}_j \leq R_j$$

Raw	C%	Cu%	Mn%	Tons	€
Iron 1	2.5	0	1.3	400	200
Iron 2	3	0	0.8	300	250
Iron 3	0	0.3	0	600	150
Copper 1	0	90	0	500	220
Cooper 2	0	96	1.2	200	240
Aluminium 1	0	0.4	1.2	300	200
Aluminium 2	0	0.6	0	250	165

Vincoli sulla qualità

for $i = 1 \dots 3$

$$\sum_{j=1}^{7} P_{ji}\text{use}_j \geq Pmin_i * \text{produce}$$

$$\sum_{j=1}^{7} P_{ji}\text{use}_j \leq Pmax_i * \text{produce}$$

end

```
sum(i in RAW) P(i,1)*use(i) >= PMIN(1)*produce
sum(i in RAW) P(i,2)*use(i) >= PMIN(2)*produce
sum(i in RAW) P(i,3)*use(i) >= PMIN(3)*produce

sum(i in RAW) P(i,1)*use(i) <= PMAX(1)*produce
sum(i in RAW) P(i,2)*use(i) <= PMAX(2)*produce
sum(i in RAW) P(i,3)*use(i) <= PMAX(3)*produce
```


Risoluzione con Xpress-MP

File MOSEL: **metalli.mos**

Xpress-IVE

Risultato:

```
Total cost: 98121.6
Amount of steel produced: 500
Alloys used:
400, 0, 39.7763, 0, 2.76127, 57.4624, 0
```

Programmazione lineare mista-intera (MILP)

Programmazione lineare

I modelli di programmazione lineare suppongono che le variabili di ottimizzazione possano assumere qualsiasi valore **reale**

$$\boxed{x_i \in \mathbb{R}}$$

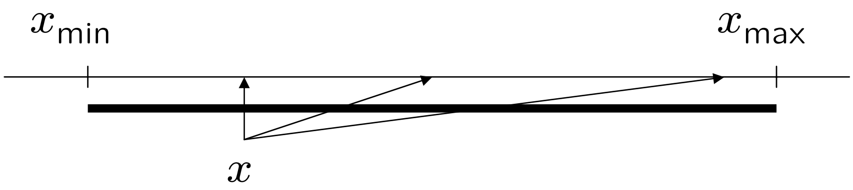
Purtroppo questa assunzione non sempre è realistica.

Esempio: Costruisci 11.7 scacchiere
Utilizza 3.6 operai
Installa 5.75 impianti di produzione

Variabili reali

Le variabili possono assumere qualsiasi valore reale.
Tale ipotesi è accettabile in molti casi.

Proprietà principale: la **continuità**

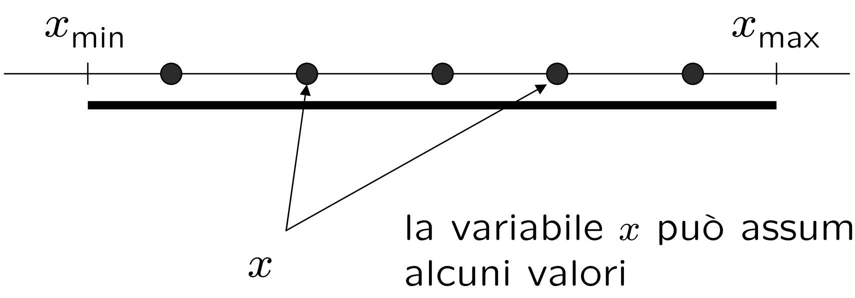


la variabile x può assumere un qualsiasi valore all'interno dell'intervallo

Vantaggio di questa ipotesi: semplicità computazionale

Variabili intere

Possono assumere valori solo su un insieme discreto.



la variabile x può assumere solo alcuni valori

Le variabili intere risultano utili in moltissime applicazioni

Modelli di programmazione intera

Sono modelli utilizzati quando:

1- Si hanno a disposizione solo un insieme finito di scelte

2- Si hanno vincoli di tipo logico

vincolo naturale

vincolo decisionale

Il modello risultante è identico ad un programma lineare, ma ha un vincolo aggiuntivo: le variabili possono assumere soltanto valori interi

Programmazione lineare mista intera (*mixed integer linear programming*, MILP)

minimizza o massimizza $\sum_{j=1}^{N} c_j x_j$ (funzione obiettivo)

soggetto a

$$\sum_{j=1}^{N} a_{ij} x_j \leq b_i \quad \text{per } i = 1 \ldots M_1$$

$$\sum_{j=1}^{N} a_{ij} x_j \geq b_i \quad \text{per } i = M_1 + 1 \ldots M_2$$

$$\sum_{j=1}^{N} a_{ij} x_j = b_i \quad \text{per } i = M_2 + 1 \ldots M_3$$

$$x_j \geq 0 \quad \text{per } j = 1 \ldots N$$

$$\boxed{x_j \in N \quad \text{for } j = 1 \ldots N_I}$$

N_I variabili intere

Variabili binarie

Possono assumere soltanto due valori: 0 o 1

$$\boxed{x \in \{0,1\}}$$

È il minimo livello di informazione che una variabile può descrivere:

SI / NO   VERO / FALSO

Sono uno strumento di modellistica molto potente, essendo in grado di esprimere decisioni, scelte, condizioni logiche, ...

Risolutori di programmi misti-interi

In conclusione:

- La programmazione mista-intera (**mixed-integer programming**) è un problema difficile da risolvere ...

MA

- Esistono risolutori di tipo Branch & Bound oppure Branch & Cut per problemi misti-interi lineari MILP e quadratici MIQP che hanno ottime prestazioni (CPLEX, Xpress-MP, GLPK, BARON, ...)

H.D. Mittelmann and P. Spellucci, "Decision Tree for Optimization Software", World Wide Web, http://plato.asu.edu/guide.html (2005)

Modellistica MILP

La complessità della risoluzione di un problema di programmazione mista intera lineare (MILP) dipende in larga misura dal numero di variabili intere coinvolte.

Quando modelliamo dobbiamo pertanto:

Utilizzare le variabili intere con parsimonia !

Dobbiamo quindi individuare quali variabili sono necessariamente da modellare come intere, ed eventualmente se possiamo ridurne il numero.

Vincoli logici

Le variabili binarie permettono di prendere delle **decisioni** di tipo:

- Fare/non fare una qualche operazione
- Scegliere fra più opzioni
- Implicazioni
- Vincoli di tipo "o questo o quello"

Arricchiscono moltissimo il linguaggio modellistico !

Fare/non fare

Sono i vincoli logici più comuni ...

Vengono modellati semplicemente da una variabile binaria $b \in \{0,1\}$: fare se b=0, non fare se b=1

Esempio:

Possiamo affittare tre aule diverse per fare lezione, A,B e C. L'aula A ha 15 posti e costa 300€, l'aula B ha 20 e costa 380€, l'aula C ha 25 posti e costa 470€. Ogni anno dobbiamo decidere quali aule affittare.

$$\text{Posti} = 15*\text{prendi}_A+20*\text{prendi}_B+25*\text{prendi}_C$$
$$\text{Costo} = 300*\text{prendi}_A+380*\text{prendi}_B+470*\text{prendi}_C$$

prendi_A, prendi_B, $\text{prendi}_C \in \{0,1\}$

Vincoli di scelta

Esprimono la scelta fra un numero finito di possibilità.

$$\text{scegli}_1 + \text{scegli}_2 + \text{scegli}_3 \leq 1$$

esprime il fatto che possiamo scegliere al più una fra le opzioni 1,2,3. Infatti:

scegli_1	0	1	0	1	0	1	0	1
scegli_2	0	0	1	1	0	0	1	1
scegli_3	0	0	0	0	1	1	1	1
	ok	ok	ok	no	ok	no	no	no

Vincoli di scelta

Altre tipologie di scelta:

$\boxed{\text{scegli}_1 + \text{scegli}_2 + \text{scegli}_3 \leq 2}$ = scegline al più 2 fra 3

$\boxed{\sum_{i=1}^{N} \text{scegli}_i \leq m}$ = scegline al più m fra N

Vincolo di "or esclusivo"

$\boxed{\text{scegli}_1 + \text{scegli}_2 + \text{scegli}_3 = 1}$ = devo scegliere necessariamente una e una sola opzione

$\boxed{\sum_{i=1}^{N} \text{scegli}_i = m}$ = scegline esattamente m fra N

Implicazioni

Come modellare relazioni che intercorrono fra scelte diverse ?

Esempio:

Possiamo ordinare i PC o dalla IBM, o dalla DELL o dalla ASUS, e i monitor o dalla PHILIPS, o dalla SONY o dalla NEC. Se compriamo i PC dalla IBM, allora dobbiamo comprare i monitor dalla SONY

$$\text{pc}_{\text{IBM}}+\text{pc}_{\text{DELL}}+\text{pc}_{\text{ASUS}}=1$$
$$\text{monitor}_{\text{PHILIPS}}+\text{monitor}_{\text{SONY}}+\text{monitor}_{\text{NEC}}=1$$
$$\boxed{\text{pc}_{\text{IBM}} \leq \text{monitor}_{\text{SONY}}}$$

equivale al vincolo logico $[\text{pc}_{\text{IBM}}=1] \rightarrow [\text{monitor}_{\text{SONY}}=1]$

Implicazioni base

$[pc_{IBM}=1] \rightarrow [monitor_{SONY}=1]$

pc_{IBM}	0	1	0	1
$monitor_{SONY}$	0	0	1	1
	ok	no	ok	ok

$pc_{IBM} \leq monitor_{SONY}$

Si vede che non è possibile ordinare i PC dalla IBM e non ordinare i monitor dalla SONY

Altre implicazioni

"Se facciamo il progetto A, allora dobbiamo fare anche i progetti B e C" → $b \geq a$, $c \geq a$

"Se facciamo il progetto A, allora dobbiamo fare il progetto B ma **non** il progetto C" → $b \geq a$, $1-c \geq a$

"Se facciamo i progetti B e C, allora dobbiamo fare anche il progetto A" → $a \geq b+c-1$

$a,b,c \in \{0,1\}$

Rendono la modellistica molto versatile !

Condizioni logiche

Nota l'equivalenza come diseguaglianza lineare di una relazione logica, è immediato sostituire "A" con "non A" semplicemente sostituendo A con (1-A)

"Se **non** facciamo il progetto A, allora dobbiamo fare il progetto B ma non il progetto C"

$b \geq a$
$1\text{-}c \geq a$

$b \geq 1\text{-}a$
$1\text{-}c \geq 1\text{-}a$

Per verificare che la traduzione in diseguaglianza è corretta, è sufficiente scrivere la tabella di verità e controllare tutti i casi.

Condizioni logiche

Alcune condizioni ricorrenti:

Al più una fra A,B,C,D	$a+b+c+d \leq 1$
Esattamente due fra A,B,C,D	$a+b+c+d = 2$
Se A allora B	$b \geq a$
Se A allora non B	$a+b \leq 1$
Se non A allora B	$a+b \geq 1$
A se e solo se B	$a = b$
Se A allora B e C	$b \geq a$ and $c \geq a$
Se A allora B oppure C	$b+c \geq a$
Se B oppure C allora A	$a \geq b$ and $a \geq c$
Se B e C allora A	$a \geq b+c-1$
Non B	$1-b$

$a,b,c,d \in \{0,1\}$

Linearizzazione di funzioni obiettivo

Funzioni obiettivo MinMax

Esempio: acquisto di azioni

Vogliamo comprare alcune azioni di tipo Tecnologico, Costruzioni, Trasporti e Moda. Vogliamo minimizzare il rischio massimo sul breve, medio e lungo termine

	breve	medio	lungo
Tecnologico	3.3	2	1
Costruzioni	2.1	1.2	3
Trasporti	2	2.2	2
Moda	1.5	2	3

misure di rischio

Funzioni obiettivo MinMax

Abbiamo tre funzioni di rischio!

$$\text{Breve} = \sum_{i \in \text{azioni}} \text{compra}(i) * \text{Rischio}(i, \text{breve})$$

$$\text{Medio} = \sum_{i \in \text{azioni}} \text{compra}(i) * \text{Rischio}(i, \text{medio})$$

$$\text{Lungo} = \sum_{i \in \text{azioni}} \text{compra}(i) * \text{Rischio}(i, \text{lungo})$$

"minimizzare il rischio massimo"

$$\min \max\{\text{Breve}, \text{Medio}, \text{Lungo}\}$$

Funzioni obiettivo MinMax

Come rendere il problema lineare ?

$$\begin{array}{rl} \min & s \\ \text{sogg.a} & s \geq \text{Breve} \\ & s \geq \text{Medio} \\ & s \geq \text{Lungo} \end{array}$$

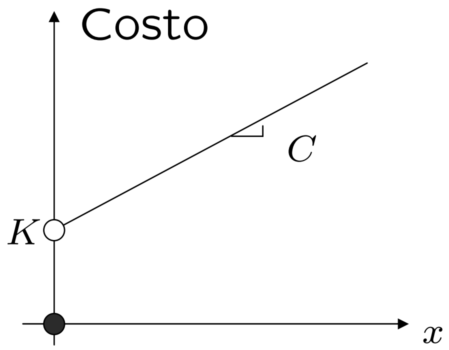
s rappresenta un upper-bound del massimo fra Breve, Medio e Lungo

$$s \geq \max\{\text{Breve}, \text{Medio}, \text{Lungo}\}$$

È facile dimostrare (per assurdo) che all'ottimo si ha:

$$s = \max\{\text{Breve}, \text{Medio}, \text{Lungo}\}$$

Costi fissi



$$\text{Costo} = \begin{cases} 0 & \text{se} \quad x = 0 \\ K + C * x & \text{se} \quad x > 0 \end{cases}$$

$K > 0$

È un tipo molto comune di funzione di costo, utile ogni volta che si abbia un costo base solo per il fatto di utilizzare una certa risorsa

Costi fissi

Introduciamo una variabile binaria:

$$\text{Costo} = b * K + C * x$$

Condizioni:

se $x \geq 0$ allora b=1
se $x = 0$ allora b=0

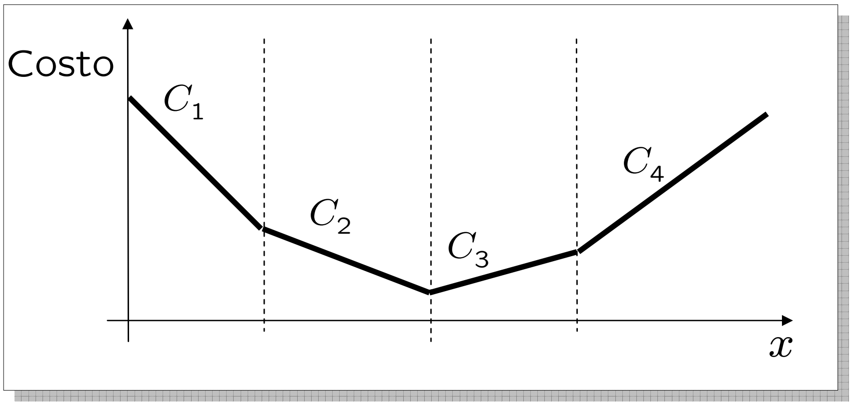
$$\longrightarrow \quad x \leq X_{max} * b$$

Verifichiamo:

1) se $x > 0$ allora non può che essere b=1

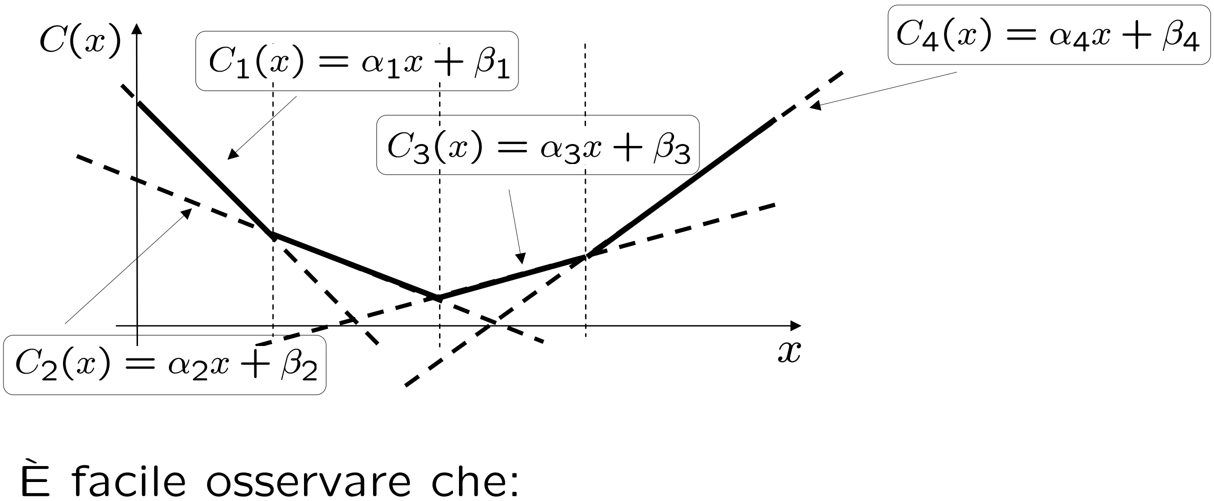
2) se $x = 0$, allora b è libera. Quando però minimizziamo il costo, essendo $K > 0$, all'ottimo avremo b=0

Costi convessi lineari a tratti



I costi lineari a tratti e **convessi** possono essere rappresentati **senza** introdurre variabili binarie !

Costi convessi lineari a tratti

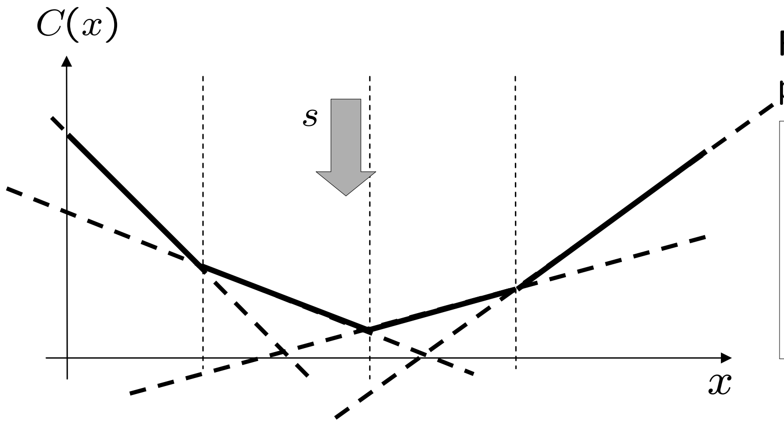


È facile osservare che:

$$C(x) = \max\{\alpha_1 x + \beta_1, \alpha_2 x + \beta_2, \alpha_3 x + \beta_3, \alpha_4 x + \beta_4\}$$

(in generale: ogni funzione lineare a tratti convessa può essere rappresentata come max di funzioni affini, e viceversa)

Costi convessi lineari a tratti



Riformulazione come programma lineare:

$$
\begin{array}{ll}
\min & s \\
\text{sogg.a} & \left\{ \begin{array}{lcl} s & \geq & \alpha_1 x + \beta_1 \\ s & \geq & \alpha_2 x + \beta_2 \\ s & \geq & \alpha_3 x + \beta_3 \\ s & \geq & \alpha_4 x + \beta_4 \end{array} \right.
\end{array}
$$

La variabile s rappresenta un upper-bound del massimo

$$s \geq \max\{\alpha_1 x + \beta_1, \alpha_2 x + \beta_2, \alpha_3 x + \beta_3, \alpha_4 x + \beta_4\}$$

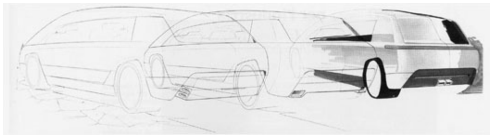
Come per le funzioni obiettivo minmax, è facile dimostrare (per assurdo) che all'ottimo si ha:

$$s = \max\{\alpha_1 x + \beta_1, \alpha_2 x + \beta_2, \alpha_3 x + \beta_3, \alpha_4 x + \beta_4\}$$

Altri esempi di modellistica MILP

Scelta di un progetto di espansione

Una casa automobilistica decide di espandere la propria capacità produttiva al fine di penetrare il mercato asiatico. Sono stati studiati alcuni progetti di espansione e cinque risultano i più favorevoli. I vari progetti hanno diversi benefici attesi dopo cinque anni, e certi costi di investimento ogni anno. Quali progetti dobbiamo attuare, noti quali saranno i finanziamenti disponibili nei prossimi anni ?



Progetti di espansione

Progetto	Descrizione	Beneficio atteso
1	Potenziare la catena di montaggio	10.8
2	Potenziare il sistema di verniciatura	4.8
3	Investire sul Diesel	3.2
4	Rinnovare il modello A	4.44
5	Nuova city-car	12.25

Costi per anno e capitale disponibile

Progetto	Anno 1	Anno 2	Anno3	Anno 4	Anno 5
1	1.8	2.4	2.4	1.8	1.5
2	1.2	1.8	2.4	0.6	0.5
3	1.2	1.0	0	0.48	0
4	1.4	1.4	1.2	1.2	1.2
5	1.6	2.1	2.5	2.0	1.8
Capitale	4.8	6	4.8	4.2	3.5

Problema di decisione

Le variabili di ottimizzazione definiscono la decisione da prendere:

$$\boxed{\text{choose}_i \in \{0,1\},\ \forall i \in PROJ}$$

$PROJ = \{1,2,3,4,5\}$

$\text{choose}_i = 1$ viene scelto il progetto i
$\text{choose}_i = 0$ non viene scelto il progetto i

“... Quale progetto scegliere ...”

Vincoli

Vincolo di risorsa:

"... noti quali saranno i finanziamenti ..."

$$\sum_{i \in PROJ} \text{choose}_i * COST_{ij} \leq \text{FUNDS}_j \qquad \forall j \in T$$

T={1,2,3,4,5}

Somma spesa nel generico anno j

Il vincolo deve essere imposto per ogni anno

Beneficio totale atteso

È facile calcolare il beneficio atteso complessivo:

$$\text{Benefit} = \sum_{i \in PROJ} \text{choose}_i * BEN_i$$

Modello di ottimizzazione

$$\min \quad \text{Benefit} = \sum_{i \in PROJ} \text{choose}_i * BEN_i$$

$$\text{sogg. a} \quad \sum_{i \in PROJ} \text{choose}_i * COST_{ij} \leq \text{FUNDS}_j, \quad \forall j \in T$$

$$\text{choose}_i \in \{0, 1\}, \quad \forall i \in \{PROJ\}$$

Modello in MOSEL

```
declarations
  PROJECTS = 1..5
  TIME = 1..5
  COST: array(PROJECTS,TIME) of real
  CAP: array(TIME) of real
  RET: array(PROJECTS) of real
  DESCR: array(PROJECTS) of string
  choose: array(PROJECTS) of mpvar
end-declarations
initializations from 'espansione.dat'
  COST CAP RET DESCR
end-initializations
Profit:= sum(p in PROJECTS) RET(p)*choose(p)
forall(t in TIME)
    sum(p  in PROJECTS) COST(p,t)*choose(p) <= CAP(t)
forall(p in PROJECTS)
    choose(p) is_binary

maximize(Profit)
```

espansione.mos

```
Soluzione MIP:
Beneficio complessivo: 19.89
 Investire sul Diesel (1)
 Rinnovare il modello A (1)
 Nuova city-car (1)
```

Soluzione del problema rilassato

Supponiamo di rilassare i vincoli di interezza:

$$\boxed{\text{choose}_i \in [0, 1]}$$

Adesso il problema diventa un semplice LP, e quindi molto più semplice da risolvere.

In MOSEL:

```
maximize(XPRS_LIN, Profit)
```


Risultati

```
Soluzione MIP:
Beneficio complessivo: 19.89
 Investire sul Diesel (1)
 Rinnovare il modello A (1)
 Nuova city-car (1)
```

MILP

```
Soluzione LP:
Beneficio complessivo: 25.775
 Potenziare la catena di montaggio (0.954167)
 Potenziare il sistema di verniciatura (0.00416667)
 Investire sul Diesel (1)
 Nuova city-car (1)
```

LP

MILP

$\text{choose}_i = 1$

```
Imponi Progetto 1 (Potenziare la catena di
montaggio):
Beneficio complessivo: 18.8
 Potenziare la catena di montaggio (1)
 Potenziare il sistema di verniciatura (1)
 Investire sul Diesel (1)
```

Commenti finali

Ruolo dell’ottimizzazione nei processi decisionali

La ricerca operativa (**operations research** e **management science**) è un insieme di tecniche basate sulla **matematica** e altri approcci di tipo **scientifico** in grado di determinare soluzioni a problemi decisionali

Ruolo dell'ottimizzazione nei processi decisionali

- È un valido aiuto nell'effettuazione di processi decisionali particolarmente complessi:
  - Fornendo un'analisi **quantitativa** del contesto in cui deve essere presa la decisione, restituendo una maggiore comprensione del contesto stesso
  - Fornendo opzioni che hanno un senso e suggerendo possibili **atti decisionali**
- I modelli di ottimizzazione sono in grado quindi di dare contributi significativi ad un certo progetto

Campi di applicazione

- Pianificazione strategica (*strategic planning*)
- Catene di distribuzione (*supply chain management*)
- Gestione dei prezzi e dei redditi (*pricing and revenue management*)
- Logistica e locazione di siti produttivi (*logistics and site location*)
- Ricerca di mercato (*marketing research*)
- …

Materiale integrativo

- **Programmazione lineare**: ulteriori tipologie di vincolo, inammisibilità, vincoli soft, ulteriori esempi applicativi
- **Programmazione mista-intera**: ulteriori tipologie di variabili, formule generali di conversione di vincoli logici in disuguaglianze miste intere, ulteriori esempi applicativi
- **Problemi di trasporto**
- **Funzioni obiettivo**: ulteriori tipologie
- **Problemi di assegnamento**
- **Programmazione quadratica**
- **Ottimizzazione di sistemi dinamici**: controllo ad orizzonte recessivo (MPC), esempi applicativi (supply-chain management)
- **Programmazione stocastica**

Fine