Programmazione Quadratica

Programmazione lineare

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

soggetto a

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

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

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

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

Programmazione quadratica

minimize or maximize

$$\boxed{\sum_{j=1}^{N}\sum_{i=1}^{N} H_{ji}x_jx_i + \sum_{j=1}^{N} c_jx_j}$$

funzione obiettivo

soggetto a

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

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

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

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

STESSI VINCOLI DELL'LP

Funzione obiettivo

Si hanno dei termini quadratici:

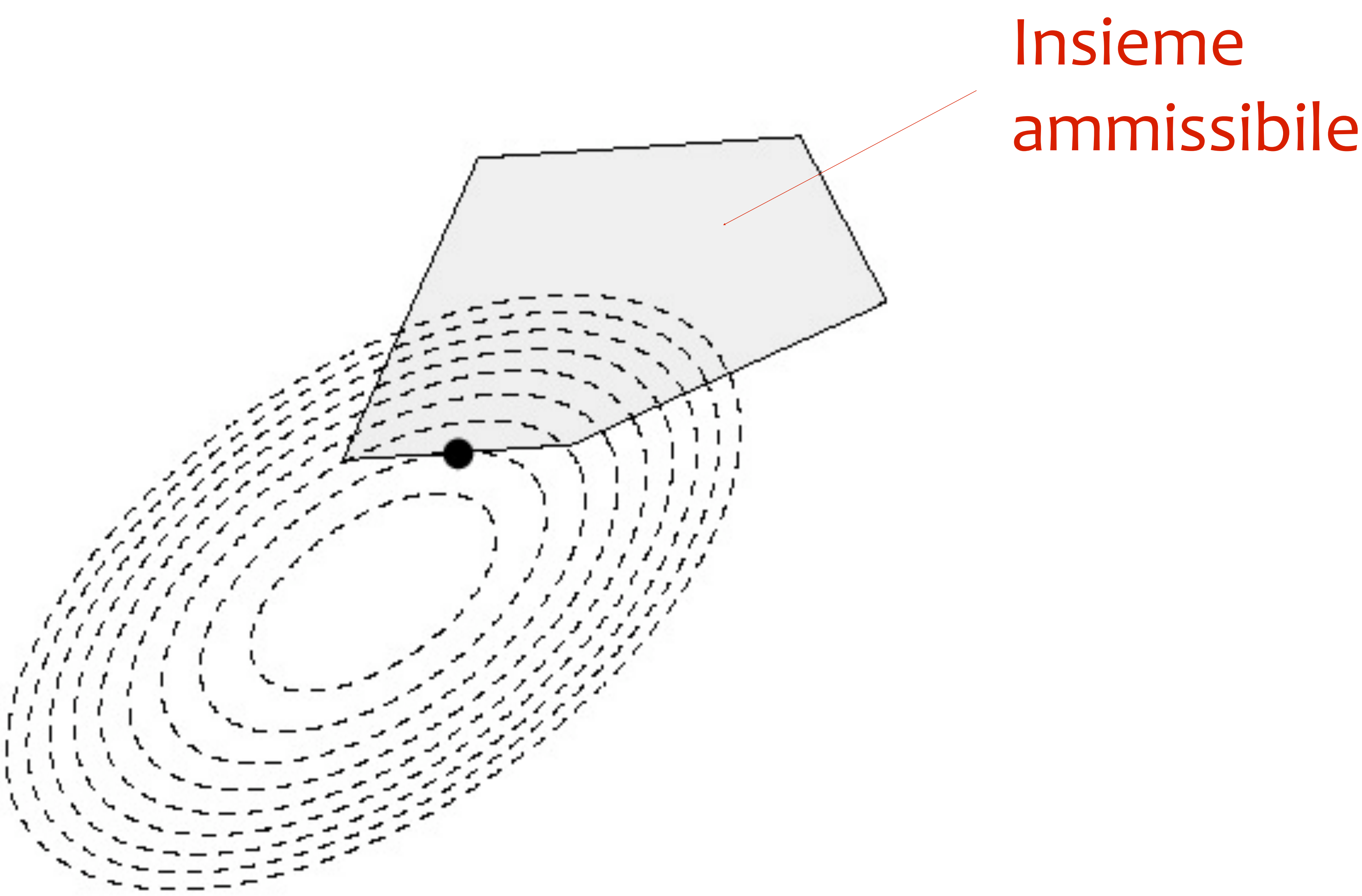
$$\sum_{j=1}^{N}\sum_{i=1}^{N}\boxed{H_{ji}x_jx_i}+\sum_{j=1}^{N}c_jx_j$$

Solitamente, per programmazione quadratica (QP) si intende quella convessa, ovvero quella per cui

$\sum_{j=1}^{N}\sum_{i=1}^{N}H_{ji}x_jx_i > 0$ per ogni $[x_1 \dots x_n] \neq [0 \dots 0]$

Programmazione quadratica

$$\begin{array}{ll} \min & \frac{1}{2}x'Hx + c'x \\ \text{s.t.} & Ax \leq b, \;\; x \in \mathbb{R}^n \end{array}$$



Programmazione quadratica

Anche la programmazione quadratica è molto diffusa nelle applicazioni e vari risolutori QP sono disponibili in molti pacchetti software per l'ottimizzazione

Xpress-MP, Cplex, Matlab, OOQP, LOQO, NAG, ...

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

Applicazioni:

- Economia (es: selezione di portafoglio titoli)
- Ingegneria (es: automazione di processo)
- ...

Matlab Optimization Toolbox QP

quadprog

Purpose

Solve the quadratic programming problem

$$\min_x \frac{1}{2}x^T H x + f^T x \quad \text{such that} \quad \begin{aligned} A \cdot x &\le b \\ Aeq \cdot x &= beq \\ lb \le x &\le ub \end{aligned}$$

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

Syntax

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

Description

`x = quadprog(H,f,A,b)` returns a vector `x` that minimizes `1/2*x'*H*x + f'*x` subject to `A*x <= b`.

Esistono risolutori QP molto più efficienti !

Scelta portafoglio titoli

(Mean Variance Portfolio Selection)

x_i = frazione di portafoglio da investire nell' i-esimo titolo

$$\sum_{i=1}^{N} x_i = 1$$

p_i = ritorno di investimento dell' i-esimo titolo (stocastico)

C = capitale totale investito

Valore del portafoglio dopo il periodo di investimento:

$$P = \sum_{i=1}^{N} C x_i (1 + p_i)$$

Valore atteso del portafoglio dopo il periodo di investimento:

$$E[P] = \sum_{i=1}^{N} C x_i (1 + \bar{p}_i) \qquad \bar{p}_i = E[p_i]$$

Scelta portafoglio titoli

Matrice di covarianza del ritorno di investimento:

$$E[(p-\bar{p})'(p-\bar{p})] = \Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1N} \\ \sigma_{12} & \sigma_{22} & \dots & \sigma_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1N} & \sigma_{2N} & \dots & \sigma_{NN} \end{bmatrix}$$

Varianza del ritorno di investimento complessivo:

$$E[(P-E[P])^2] = E[(C\sum_{i=1}^{N} x_i(p_i-\bar{p}_i))^2] = C^2 \sum_{i,j=1}^{n} \sigma_{ij}x_ix_j = C^2x'\Sigma x$$

Obiettivo: minimizzare il rischio di investimento, cioè la varianza del ritorno di investimento

Modello matematico

$$
\begin{array}{ll}
\min & x'\Sigma x \\
\text{s.t.} & \displaystyle\sum_{i=1}^{N} \bar{p}_i x_i \geq P_{\min} \\
 & \displaystyle\sum_{i=1}^{N} x_i = 1 \\
 & x_i \geq 0
\end{array}
$$

(modello di Markowitz)

(Harry Max Markowitz 1927-)

Programmazione lineare mista-intera (MILP)

Programmazione lineare

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

soggetto a

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

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

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

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

Programmazione lineare

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

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

Purtroppo questa assunzione non sempre è realistica.

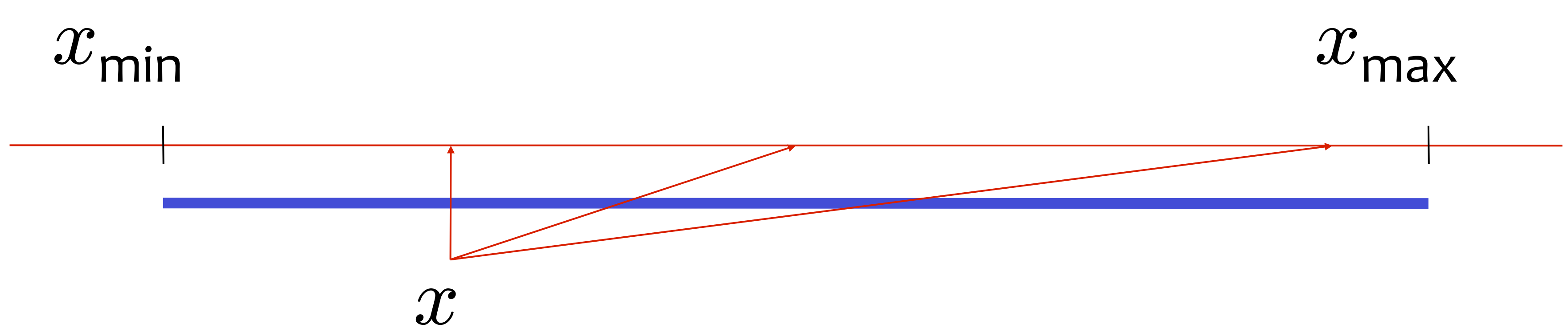
Esempio:

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**

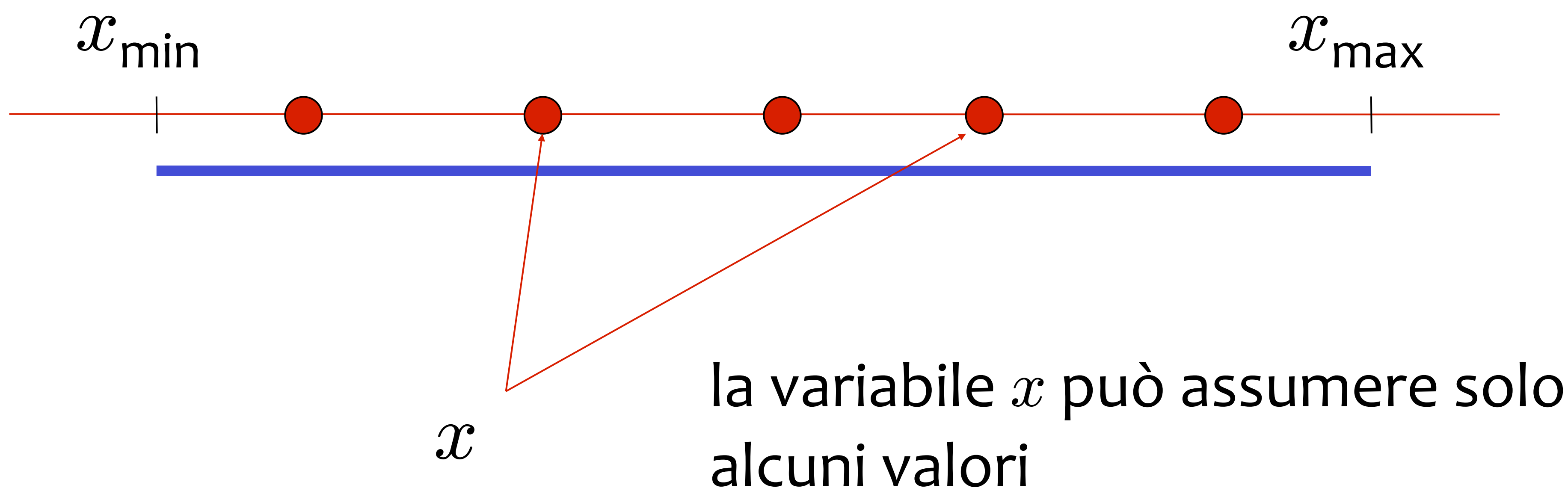


la variabile x può assumere un qualsiasi valore all'interno dell'intervallo

Vantaggio di questa ipotesi: semplicità computazionale

Variabili intere

Possono assumere valori solo su un insieme discreto.



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

Modellistica MILP

Oltre alle variabili reali continue, si hanno a disposizione nuovi tipi di variabili:

intere
binarie
semicontinue
SOS (special ordered sets)
...

Si possono modellare una gamma molto estesa di nuovi vincoli ! In particolare:

vincoli logici

Variabili intere

Le variabili intere possono assumere valori soltanto nell'insieme dei numeri naturali $\mathbb{N} = \{0,1,2,\ldots\}$

$$\boxed{x \in \mathbb{N}}$$

È un vincolo molto comune e del tutto naturale.

Esempi:

Numero di persone assegnate ad un progetto
Numero di impianti di produzione impiegati
Numero di lotti prodotti
...

Variabili binarie

Possono assumere soltanto due valori: 0 o 1

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

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

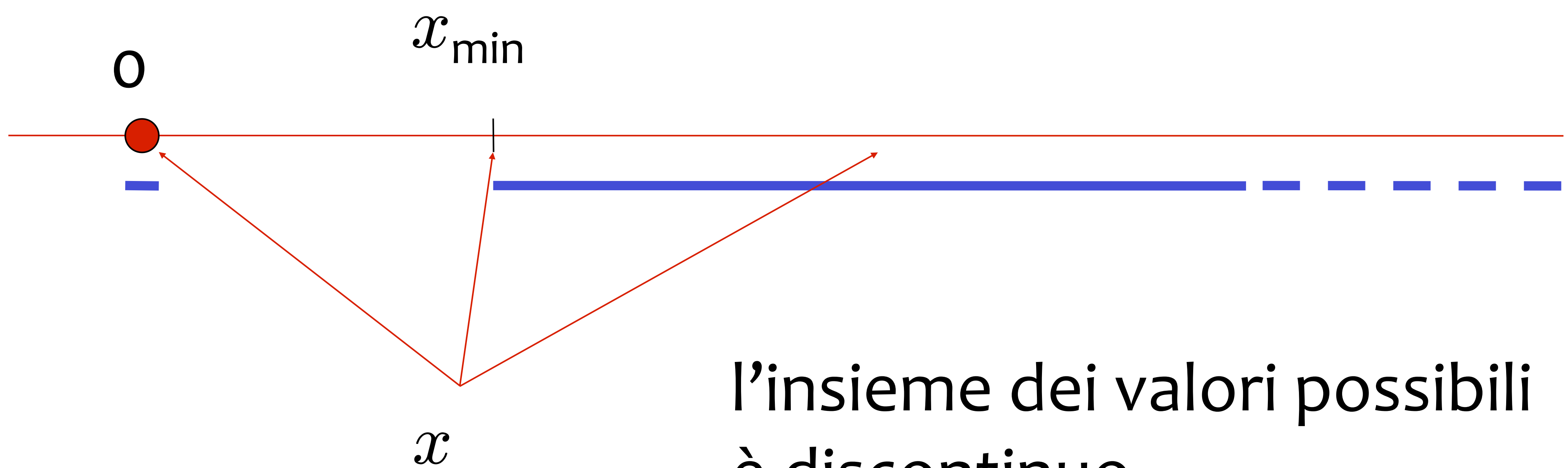
SI / NO

VERO / FALSO

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

Variabili semicontinue

Una variabile semicontinua o è nulla, oppure assume un valore maggiore di una certa quantità x_{min}



l'insieme dei valori possibili è discontinuo

Permettono di esprimere delle situazioni reali molto frequenti. Esempi:

Se vendo, devo vendere almeno una certa quantità
Livelli di produzione minima per garantire un profitto

Altri tipi di variabile

Esistono altri tipi di variabile non reale. Ad esempio:

Special Ordered Sets SOS1 e SOS2

Riconoscerle è importante per facilitare la risoluzione del problema di ottimizzazione.

Dal punto di vista della sola modellistica, possiamo focalizzare l'attenzione soltanto sulle variabili continue e intere.

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.

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

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

$\text{prendi}_A, \text{prendi}_B, \text{prendi}_C \in \{0,1\}$

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

$$pc_{IBM}+pc_{DELL}+pc_{ASUS}=1$$

$$monitor_{PHILIPS}+monitor_{SONY}+monitor_{NEC}=1$$

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

equivale al vincolo logico $[pc_{IBM}=1] \rightarrow [monitor_{SONY}=1]$

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" → $a \geq b+c-1$

Rendono la modellistica molto versatile !

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

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)

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

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

$\Rightarrow$

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

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:

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

Funzioni obiettivo razionali

Abbiamo già visto vincoli sulla qualità e di miscela-zione, che sono vincoli sul **rapporto di variabili**

Esempio: vincolo sulla qualità

$$\frac{\sum_{i=1}^{N} c_i x_i}{\sum_{i=1}^{N} x_i} \leq c_{\text{media}}$$

Supponendo $\sum_{i=1}^{N} x_i > 0$, è possibile riscrivere il vincolo come vincolo lineare:

$$\sum_{i=1}^{N} c_i x_i \leq c_{\text{media}} \sum_{i=1}^{N} x_i$$

Vediamo adesso come trattare rapporti di variabili nella funzione obiettivo

Funzioni obiettivo razionali

Esempio:

"La ditta Metalli S.p.A. ha ricevuto un ordine di 500 tonnellate di acciaio da usare per la costruzione di una nave. L'acciaio deve avere determinate caratteristiche di composizione.

La ditta dispone di sette tipi diversi di materiale grezzo da utilizzare per la produzione di questo acciaio.

L'obiettivo è determinare la composizione che minimizza la percentuale di zolfo, soddisfacendo i vincoli sulle caratteristiche di composizione."

Dati a disposizione

Mat. grezzo	C%	Cu%	Mn%	Tons	S%
Iron 1	2.5	0	1.3	400	0.5
Iron 2	3	0	0.8	300	0.76
Iron 3	0	0.3	0	600	0.3
Copper 1	0	90	0	500	1.2
Copper 2	0	96	1.2	200	1.1
Aluminium 1	0	0.4	1.2	300	0.3
Aluminium 2	0	0.6	0	250	0.45

Composizioni %, quantità disponibili, % di zolfo

Funzione obiettivo razionale

$$\min \text{Zolfo\%} = \frac{\sum S_j * \text{use}_j}{\sum \text{use}_j}$$

funzione non-lineare delle variabili !

sogg. a:

$$\sum A_{ij}\text{use}_j \leq b_i$$

rappresentano i vincoli originali del problema

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

$\text{produce} \geq 500$

for $j = 1 \ldots 7$

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

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

Cambio di variabili

$$\begin{aligned} d &= \frac{1}{\sum \mathrm{use}_j} \\ x_j &= \mathrm{use}_j * d \end{aligned}$$

1) abbiamo aggiunto una nuova variabile: d

2) abbiamo abbandonato le vecchie variabili use_j
per sostituirle con le nuove variabili x_j

Cambio di variabili

Considerando la variabile d come libera, otteniamo un vincolo lineare sulle variabili x_j:

$$\begin{aligned} d &= \frac{1}{\sum \mathrm{use}_j} \\ x_j &= \mathrm{use}_j * d \end{aligned}$$

$$\rightarrow \quad 1 = \sum \mathrm{use}_j * d = \sum x_j$$

$$\rightarrow \quad \boxed{\sum x_j = 1}$$

nuovo vincolo lineare
da aggiungere al problema

Moltiplicando i vincoli originali a destra e a sinistra per la quantità d (d>0), otteniamo dei vincoli equivalenti in x,d:

$$\sum A_{ij}\mathrm{use}_j \leq b_i \quad \rightarrow \quad \sum A_{ij}x_j \leq b_i * d$$

Cambio di variabili

Funzione obiettivo:

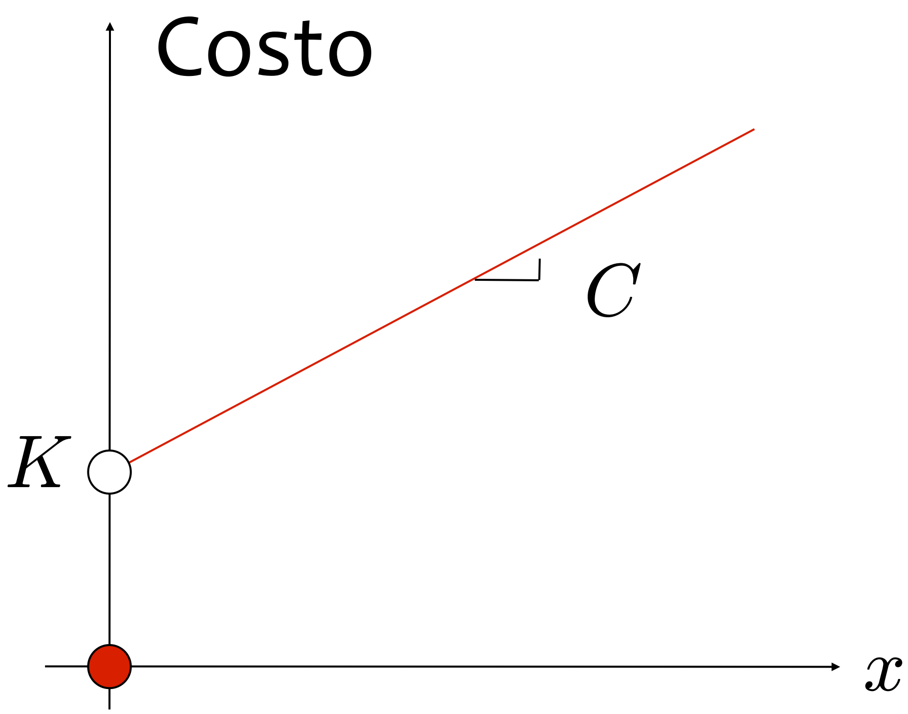
$$\min \text{Zolfo\%} = \frac{\sum S_j * \text{use}_j}{\sum \text{use}_j} = \sum S_j * x_j$$

Tutti i vincoli (=quelli originali + il nuovo vincolo) possono essere espressi come un insieme di disuguaglianze lineari sulle variabili x_j, d.

$$\sum A_{ij} x_j \leq b_i * d$$

Programma Lineare

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

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

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 variabili 1 (*economies of scale*)

Considera il seguente esempio: i prezzi del legno sono fissati come segue:

0-100 kg	0.99 €/kg
100-200 kg	0.95 €/kg
200-400 kg	0.90 €/kg
+400 kg	0.80 €/kg

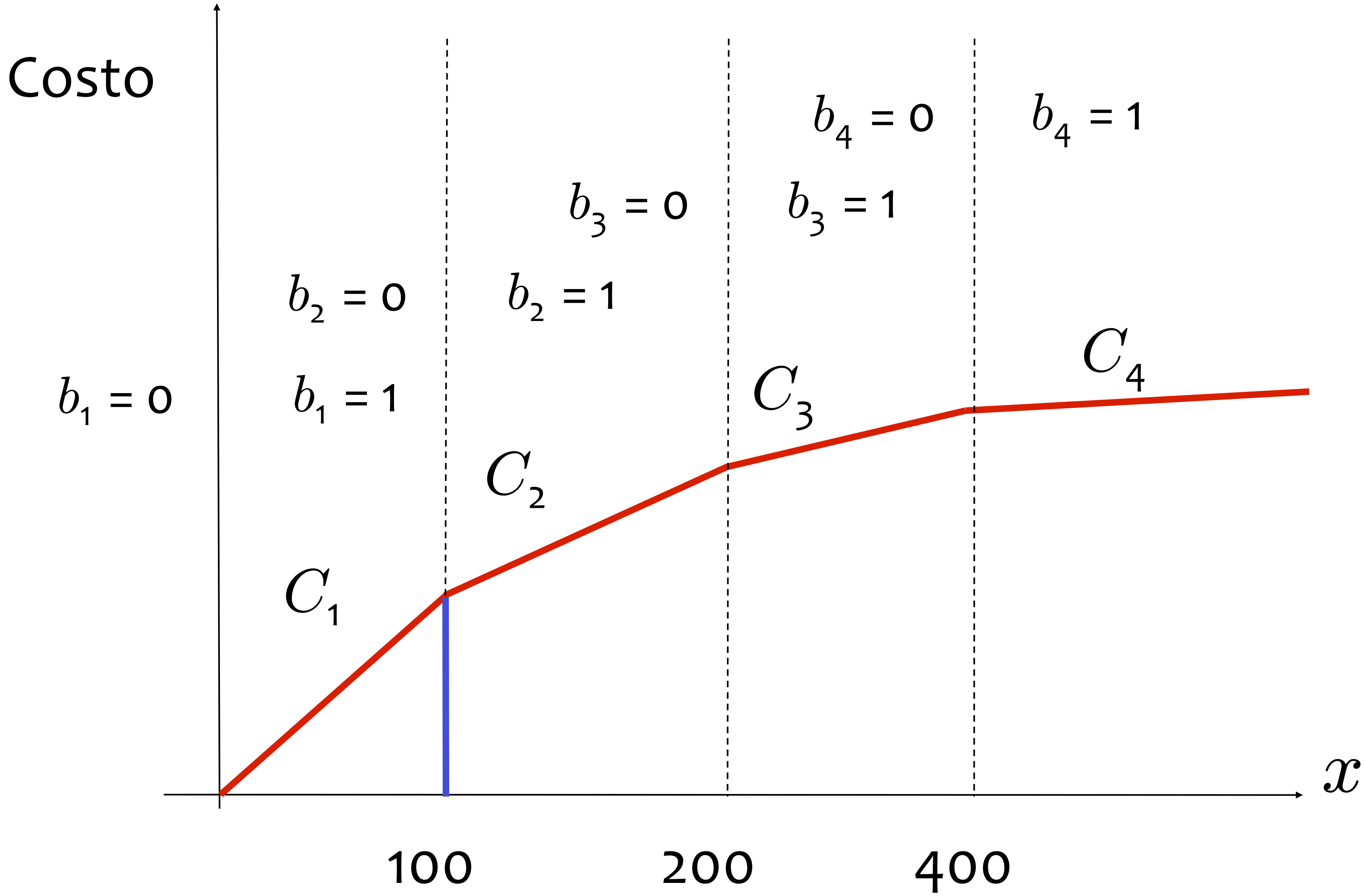
scala dei prezzi

Nota: non vengono effettuati sconti in blocco! I primi 100 kg sono sempre pagati 99 €, indipendentemente dal fatto che se ne comprino quantità maggiori

Come esprimere la funzione di costo ?

Costi variabili 1

Occorre introdurre delle variabili continue x_1, x_2, x_3, x_4 e binarie b_1, b_2, b_3, b_4



$x = x_1 + x_2 + x_3 + x_4$

x_i = quantità acquistata al prezzo C_i

Costi variabili 1

$$\text{Costo} = C_1x_1 + C_2x_2 + C_3x_3 + C_4x_4$$

$$\begin{gathered}
100 * b_2 \leq x_1 \leq 100 * b_1 \\
(200 - 100) * b_3 \leq x_2 \leq (200 - 100) * b_2 \\
(400 - 200) * b_4 \leq x_3 \leq (400 - 200) * b_3 \\
x_4 \leq X_{max} * b_4 \\
b_1 \geq b_2 \geq b_3 \geq b4 \\
x_1, x_2, x_3, x_4 \geq 0
\end{gathered}$$

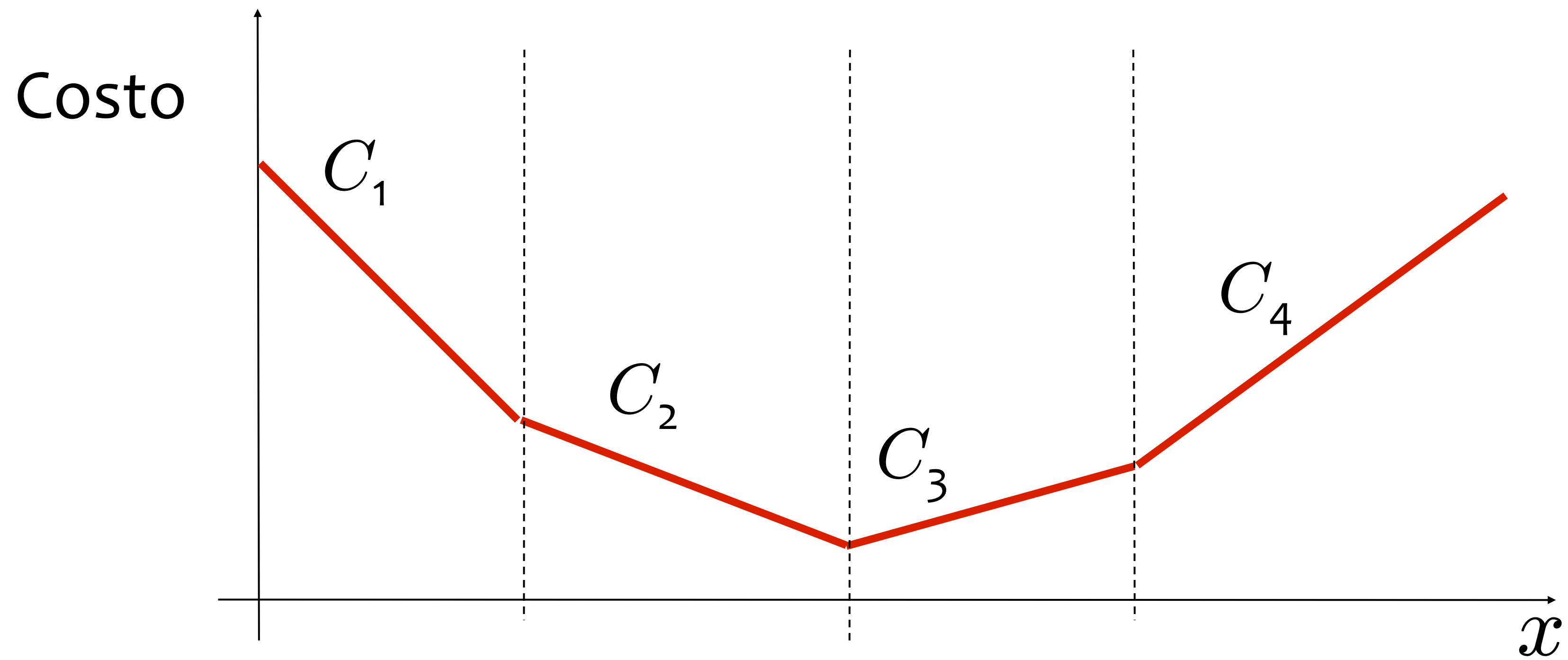
Esempio:

$$100 * b_2 \leq x_1 \leq 100 * b_1$$
$$(200-100) * b_3 \leq x_2 \leq (200-100) * b_2$$
$$(400-200) * b_4 \leq x_3 \leq (400-200) * b_3$$
$$x_4 \leq X_{max} * b_4$$
$$b_1 \geq b_2 \geq b_3 \geq b4$$

[1000]
[1100]
[1110]
[1111]

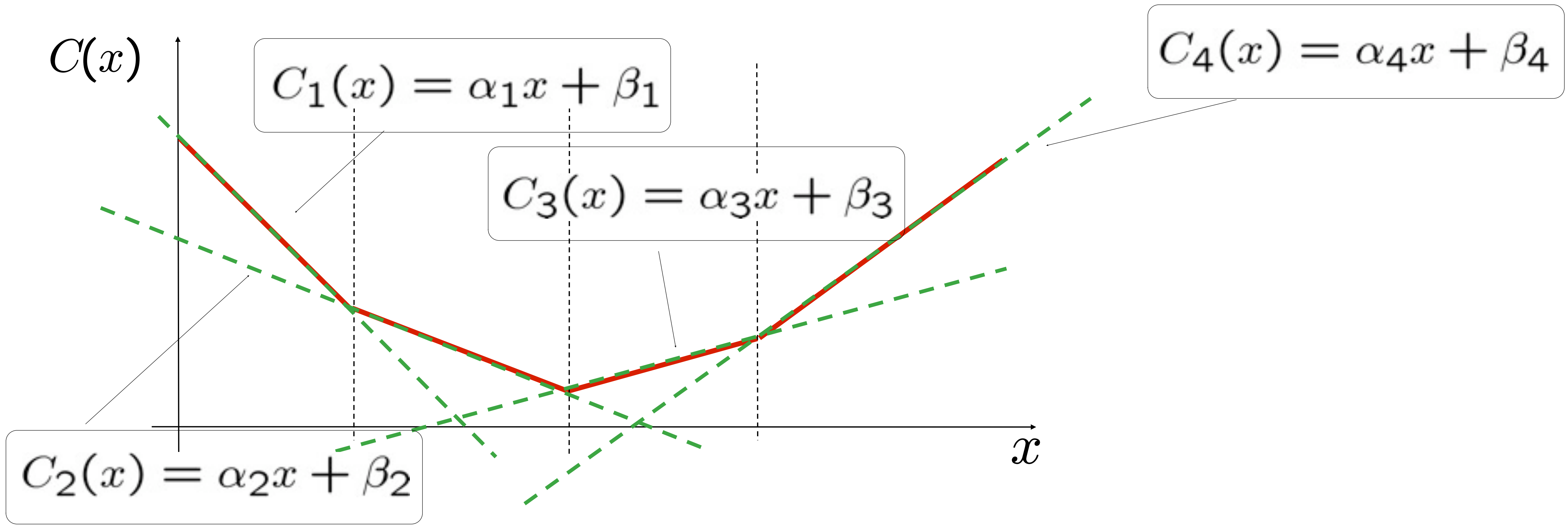
x_1=100, x_2=100, x_3 minore di 200

Costi convessi lineari a tratti



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

Costi convessi lineari a tratti

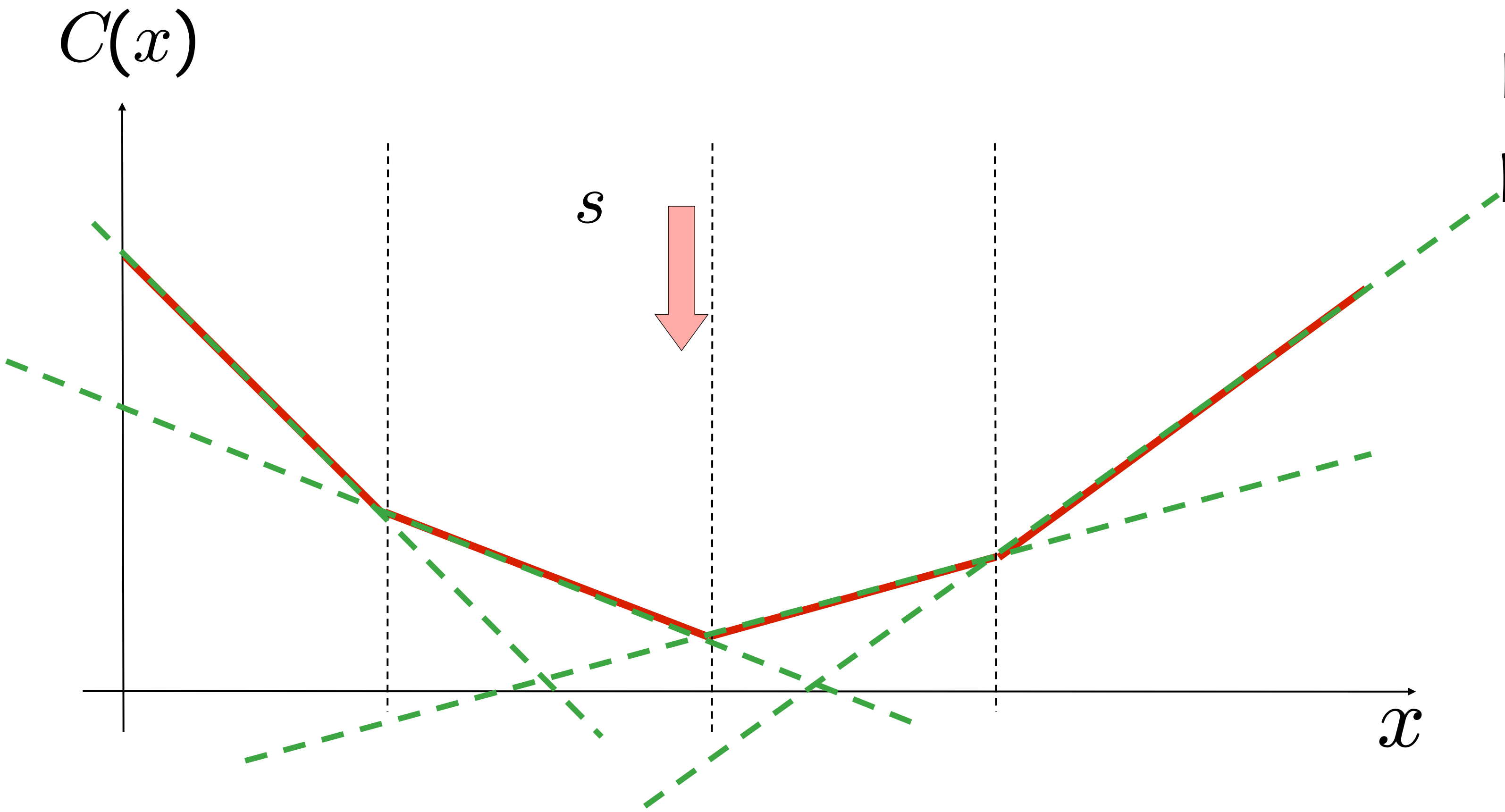


È facile osservare che:

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

(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:

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

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:

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

Costi variabili 2

Considera il seguente esempio: i prezzi del legno sono fissati come segue:

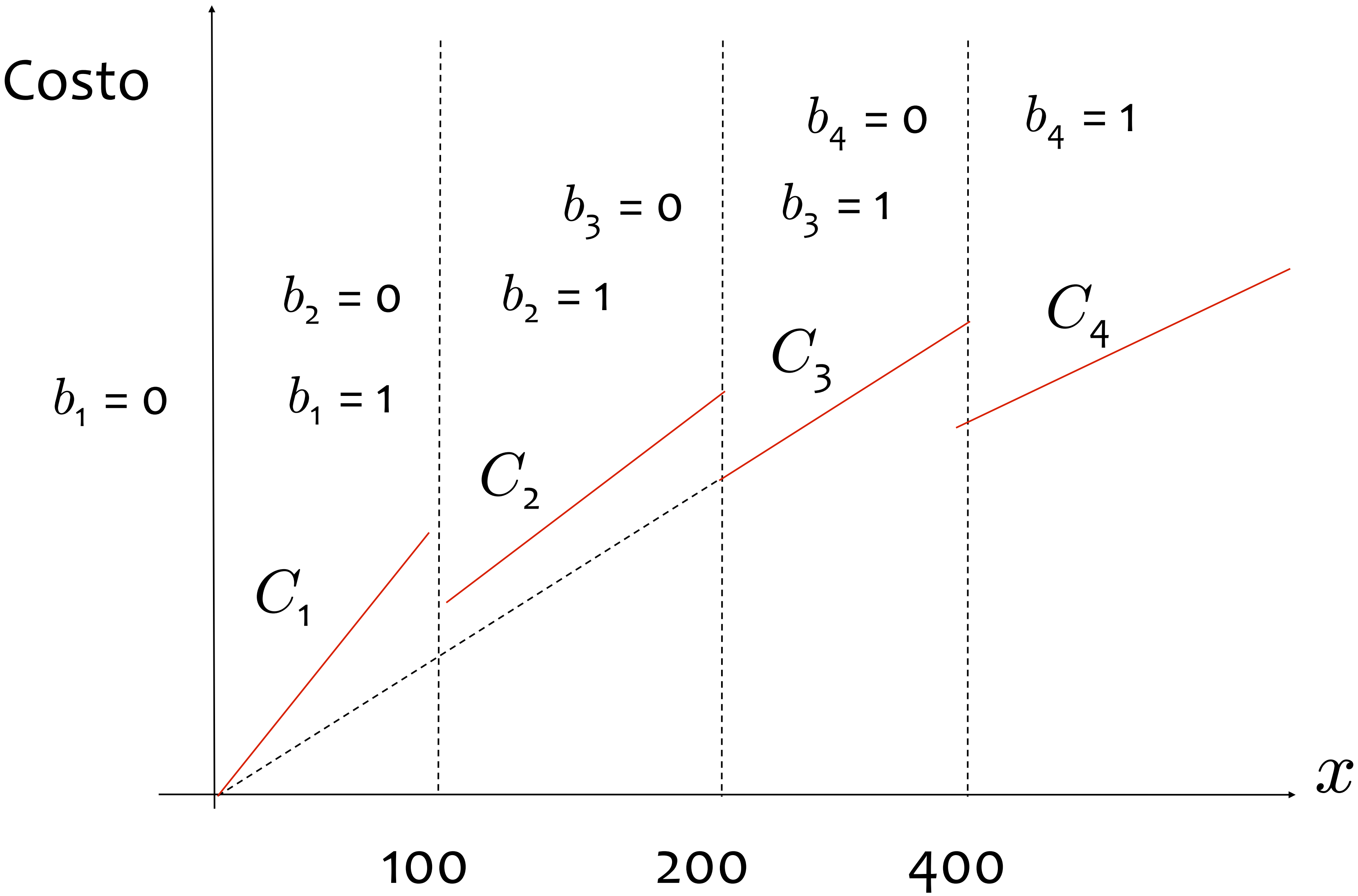
0-100 kg	0.99 €/kg
100-200 kg	0.95 €/kg
200-400 kg	0.90 €/kg
+400 kg	0.80 €/kg

scala dei prezzi

Nota: vengono effettuati sconti in blocco! Maggiore è la quantità acquistata, minore è il costo per kg

Come esprimere la funzione di costo ?

Costi variabili 2



$x = x_1+x_2+x_3+x_4$

x_i = quantità acquistata al prezzo C_i

Costi variabili 2

Definiamo le variabili b e x esattamente come prima:

$$
\begin{gathered}
100 * b_2 \leq x_1 \leq 100 * b_1 \\
(200 - 100) * b_3 \leq x_2 \leq (200 - 100) * b_2 \\
(400 - 200) * b_4 \leq x_3 \leq (400 - 200) * b_3 \\
x_4 \leq X_{max} * b_4 \\
b_1 \geq b_2 \geq b_3 \geq b4 \\
x_1, x_2, x_3, x_4 \geq 0
\end{gathered}
$$

e riscriviamo il costo come:

$$
\begin{aligned}
\text{Costo} = \quad & C_1 x * (b_1 - b_2) + C_2 x * (b_2 - b_3) \\
& + C_3 x * (b_3 - b_4) + C_4 x * b_4
\end{aligned}
$$

Costi variabili 2

$$\text{Costo} = \quad C_1 \boxed{x * (b_1 - b_2)} + C_2 x * (b_2 - b_3)$$
$$+ C_3 x * (b_3 - b_4) + C_4 x * b_4$$

Il vincolo $b_1 \lambda b_2 \lambda b_3 \lambda b_4$ impone che le uniche combinazioni possibili siano [1000],[1100],[1110],[1111]

$\Rightarrow$ uno soltanto dei quattro termini sarà non nullo

OK, ma il costo non è lineare !

Abbiamo il **prodotto fra variabili continue e binarie**

Prodotto variabili continue-binarie

Tecnica del "big-M":

$$z = x * b$$

$x \geq 0$

$z \geq 0$

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

può essere modellato come

$$\begin{aligned} z &\leq x \\ z &\geq x - M * (1 - b) \\ z &\leq M * b \end{aligned}$$

sotto l'ipotesi che M sia scelto sufficientemente grande, in modo che sia sempre $x \leq M$

Prodotto variabili continue-binarie

$$
\begin{aligned}
z &\leq x \\
z &\geq x - M * (1 - b) \\
z &\leq M * b
\end{aligned}
$$

$x \geq 0$

$z \geq 0$

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

b=0

$$
\begin{aligned}
z &\leq x \\
z &\geq x - M \\
z &\leq 0
\end{aligned}
$$

z=0

b=1

$$
\begin{aligned}
z &\leq x \\
z &\geq x \\
z &\leq M
\end{aligned}
$$

z=x

Vincoli di cardinalità (*counting*)

È un vincolo (molto ricorrente) che limita il **numero** di scelte discrete che possiamo decidere.

Associamo ad ogni opzione i, i=1,2,...,N, una variabile binaria b_i, $b_i \in \{0,1\}$

$$\sum_{i=1}^{N} b_i \leq M$$ Possiamo scegliere al più M opzioni

$$\sum_{i=1}^{N} b_i = M$$ Dobbiamo scegliere esattamente M opzioni

$$\sum_{i=1}^{N} b_i \geq M$$ Dobbiamo scegliere almeno M opzioni

Vincoli di “or” logico

Anche questi sono vincoli molto ricorrenti. Hanno bisogno di variabili binarie per essere modellati.

Esempio:

Siamo interessati ad investire del denaro per scopi pubblicitari, al fine di ottenere un buon impatto commerciale in TV o in radio...

Basta che sia soddisfatto almeno uno dei due vincoli

Vincoli di “or” logico

Impatto minimo = 1.5

	TV	Radio	Costo
Mediaset	0.5	0.3	250
RAI	0.4	0.35	200
Radio Capital	0.0	0.5	50
MTV	0.3	0.2	100

Vincoli di "or" logico

Vincoli sulle variabili $buy(i)$:

$$\sum_{i \in MEDIA} buy(i) * \text{Impact}(i, TV) \geq \text{Impact}_{\text{min}}$$

oppure

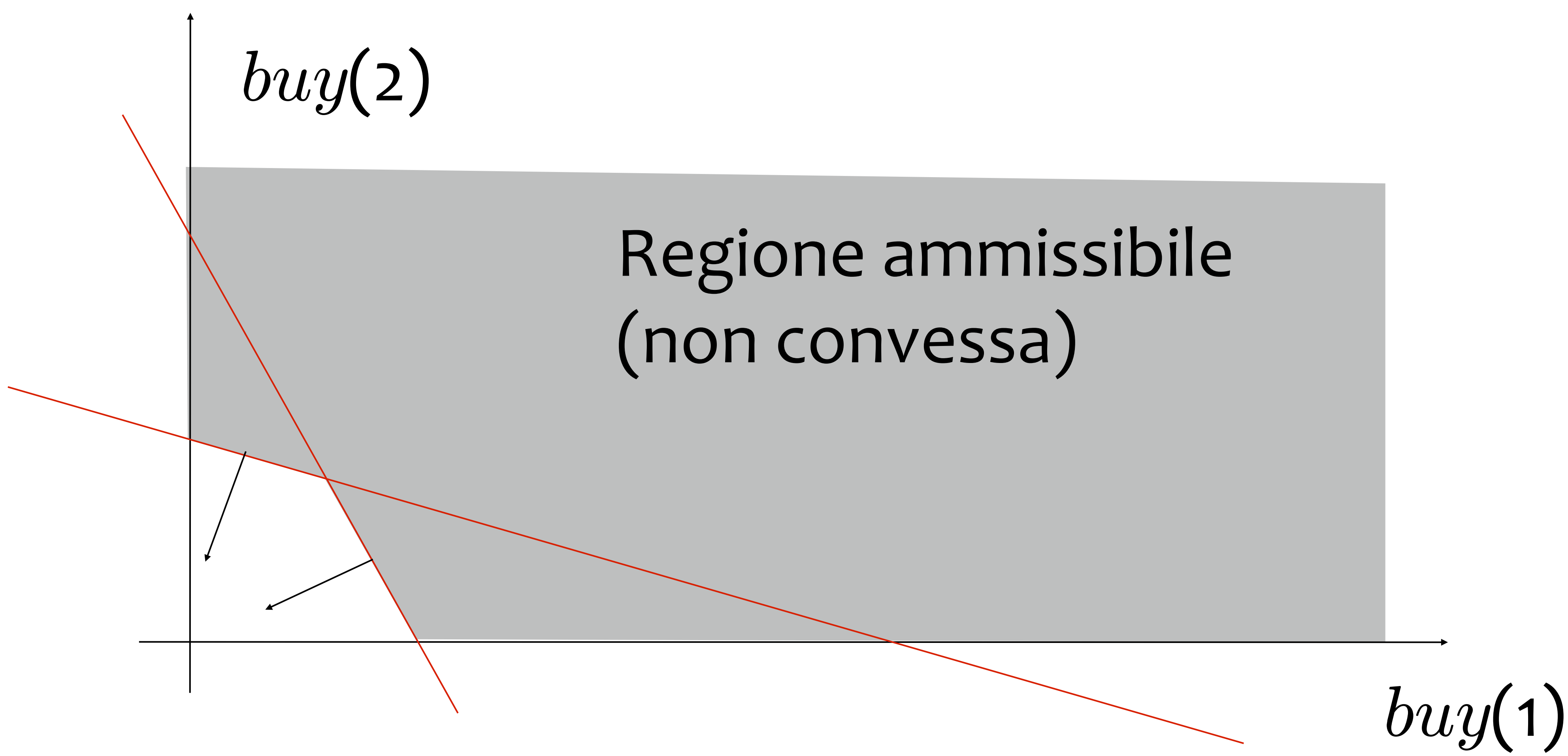
$$\sum_{i \in MEDIA} buy(i) * \text{Impact}(i, RADIO) \geq \text{Impact}_{\text{min}}$$

$$\sum_{i \in MEDIA} buy(i) * \text{Impact}(i, TV) \geq b * \text{Impact}_{\text{min}}$$
$$\sum_{i \in MEDIA} buy(i) * \text{Impact}(i, RADIO) \geq (1-b) * \text{Impact}_{\text{min}}$$

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

Vincoli di “or” logico

Se almeno uno dei due vincoli è soddisfatto, allora esiste un valore della variabile binaria b ammissibile. Viceversa, se entrambe i vincoli non possono essere soddisfatti, non esistono valori ammissibili per b.



Programmazione stocastica

Incertezza dei dati del problema

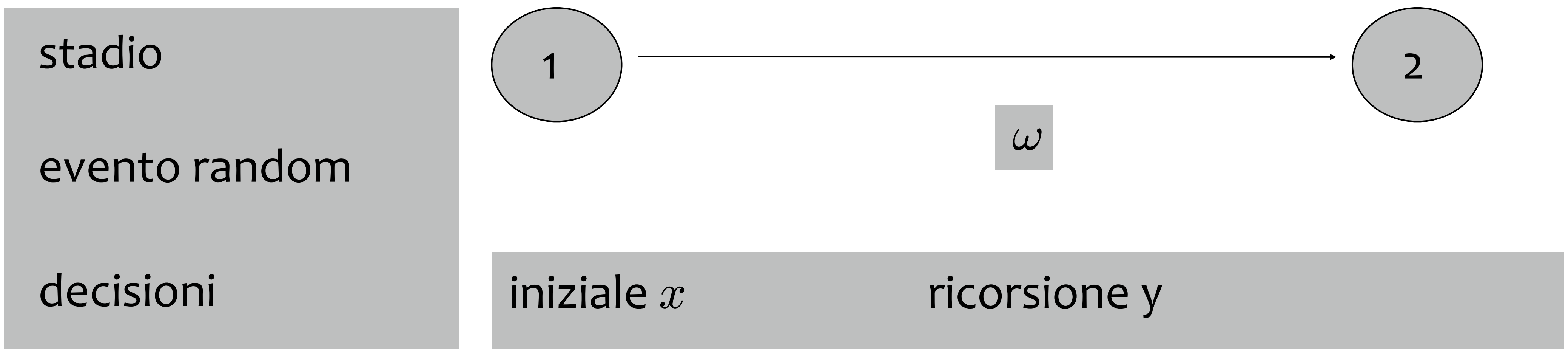
- Finora abbiamo ipotizzato che i dati del problema di ottimizzazione (es. A,b,c di un problema LP) fossero noti con precisione

- Spesso però in pratica si hanno molte situazioni in cui tali dati non sono *deterministici*, ma piuttosto **stocastici**, cioè descritti da funzioni di probabilità

- Questo accade ad esempio quando i dati del problema saranno noti soltanto nel futuro (es: domanda di un cliente, prezzo del materiale, ecc.)

Programmazione stocastica (2 stadi)

- Nella modellazione a 2 stadi, la sequenza delle decisioni può essere rappresentata come segue:



- ω = variabile random
- x = variabile di decisione presa *prima* che l'evento ω si sia manifestato
- y = variabile di decisione presa *dopo* che l'evento ω si sarà manifestato, e quindi da esso dipendente. Permette di effettuare *hedging*.

Programmazione stocastica (2 stadi)

<u>Esempio</u>: Significato delle variabili in problemi gerarchici di pianificazione (*hierarchical planning problems*)

- **Stadio 1 (variabile x)**: decisioni strategiche e globali, con impatto a lungo termine.

  Esempi: investimenti, capacità produttive, ecc.
  In questo caso la domanda del cliente può essere trattata come incerta.

- **Stadio 2 (variabile y)**: variabili operative, decisioni a breve termine in risposta a come le variabili stocastiche ω si sono realizzate (ad esempio la domanda del cliente).

  Esempi: quantitativi di produzione, schedulazione delle macchine, ecc.

Programmazione stocastica (2 stadi)

- Considera il seguente problema di programmazione lineare stocastica a due-stadi con ricorsione

$$
\begin{array}{ll}
\max & E[a'x + c(\omega)'y(\omega)] \\
\text{s.t.} & Ax = b \\
 & B(\omega)x + C(\omega)y(\omega) = d(\omega) \\
 & x \geq 0, \;\; y(\omega) \geq 0
\end{array}
$$

$$
\begin{array}{l}
x \in \mathbb{R}^n \\
y \in \mathbb{R}^m \\
\omega \in \mathbb{R}^p
\end{array}
$$

- ω = variabile random
- x = variabile di decisione presa *prima* che l'evento ω si sia manifestato
- y = variabile di decisione presa *dopo* che l'evento ω si sarà manifestato, e quindi da esso dipendente

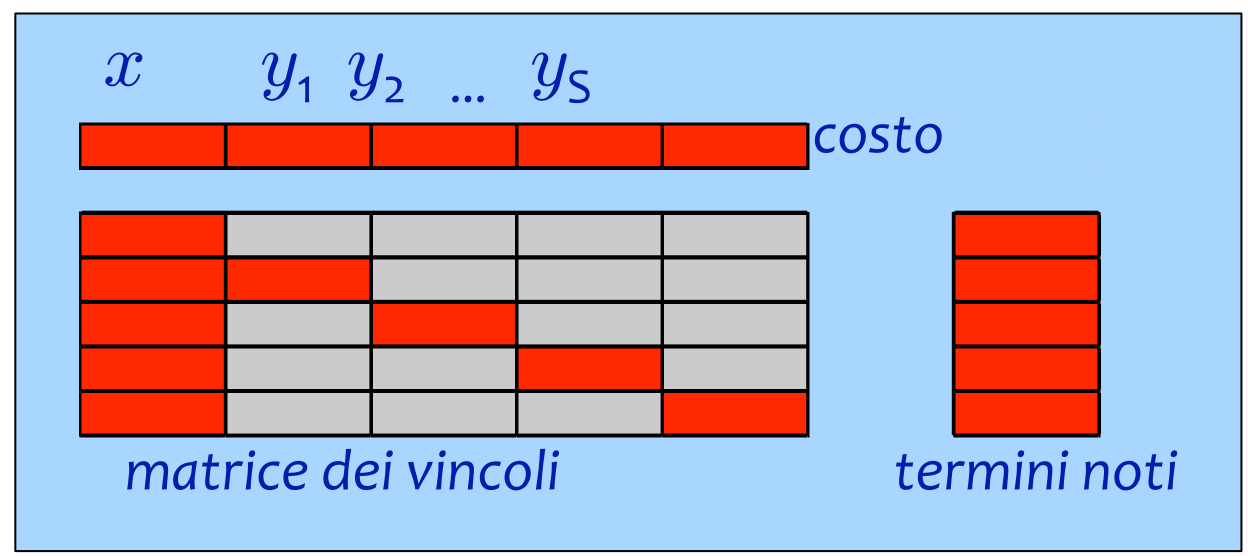
Nota: $$\max E[a'x + c(\omega)'y(\omega)] = \max a'x + E[\max_{y(\omega)} c(\omega)'y(\omega)]$$

Programmazione stocastica (2 stadi)

- Supponiamo che il vettore ω possa assumere solo un numero finito di valori possibili (detti "*scenari*") $\omega \in \{\omega_1, \omega_2, \ldots, \omega_S\}$
- Siano $\{p_1, p_2, \ldots, p_S\}$ le rispettive probabilità (se equiprobabili p_i=1/S)
- Allora, possiamo riscrivere $$E[\max_{y(\omega)} c(\omega)' y(\omega)] = \sum_{j=1}^{S} \max_{y(\omega_j)} c(\omega_j)' y(\omega_j)$$

  e quindi il problema di ottimizzazione stocastica come

$$
\begin{array}{rl}
\max & a'x + \sum_{j=1}^{s} p_j c_j' y_j \\
\text{s.t.} & Ax = b \\
& B_1 x + C_1 y_1 = d_1 \\
& \vdots \\
& B_S x + C_s y_s = d_S \\
& x, y_1, \ldots, y_s \geq 0
\end{array}
$$



La struttura particolare (e sparsa) del problema può essere sfruttata dai risolutori

Esempio

- Un municipalizzata deve gestire gli acquisti di gas per l'anno in corso e il prossimo.
- Il gas può essere comprato, venduto e immagazzinato
- A seconda delle condizioni climatiche, i prezzi di acquisto per quantitativo di gas e la domanda cambiano:

Clima	Probabilità p	Costo gas c	Domanda d
Normale	1/3	5	100
Freddo	1/3	7	120
Molto freddo	1/3	9	130

- Supponiamo che nell'anno in corso ci sia un clima "normale", mentre il clima dell'anno prossimo è incerto
- Quanto gas dobbiamo comprare quest'anno ? Quanto dobbiamo comprarne per immagazzinarlo ? Quanto dovremo comprarne l'anno prossimo ?

`http://stoprog.org/spintroduction.html`

Esempio

- È un problema di programmazione lineare stocastica a due stadi
- Variabili:

\- ω = variabile random "clima" (3 possibili valori)

\- x = variabile di decisione presa *prima* che l'evento ω si sia manifestato

x_1 = acquisto di gas da rivendere subito (anno 1)

x_2 = acquisto di gas da immagazzinare (anno 1)

\- y = variabile di decisione presa *dopo* che l'evento ω si sarà manifestato

y_{1j} = utilizzo di gas immagazzinato (anno 2)

y_{2j} = acquisto di ulteriore gas da rivendere subito (anno 2) j=1,2,3

Esempio

- Funzione costo:

$$\min 5x_1 + (5+1)x_2 + E[c(\omega)y_2(\omega)]$$

$$\min 5x_1 + (5+1)x_2 + \frac{1}{3}5y_{21} + \frac{1}{3}7y_{22} + \frac{1}{3}9y_{23}$$

- Vincoli:

  - Soddisfacimento domanda:

$$x_1 \geq 100$$
$$y_{1j} + y_{2j} \geq d_j, \;\; j = 1,2,3$$

  - Disponibilità di gas immagazzinato:

$$y_{1j} \leq x_2, \;\; j = 1,2,3$$

  - Il gas si può solo comprare o prelevare dallo stoccaggio:

$$x_1, x_2, y_{1j}, y_{2j} \geq 0, \;\; j = 1,2,3$$

Esempio

- Risultato: costo = 1236.67, x_1=100, x_2=100

  y_{11}=100, y_{21}=0, y_{12}=100, y_{22}=20, y_{13}=100, y_{23}=30

- Calcoliamo il costo per ogni realizzazione di scenario:

Clima	Azione stage 2	Costo totale
Normale	nessun acquisto aggiuntivo	1100
Freddo	compra 20 unità al costo di 7 per unità	1240
Molto freddo	compra 30 unità al costo di 9 per unità	1370

- Conferma che il costo medio è (1100+1240+1370)/3=1236.67

Esempio

- Soluzione euristica: rimpiazzare i coefficienti con i rispettivi valori attesi:

$$\begin{array}{ll} \min & 5x_1 + (5+1)x_2 + E[c(\omega)]y_2 \\ \text{s.t.} & x_1 \geq 100 \\ & y_1 + y_2 \geq E[d(\omega)] \end{array} \quad \Rightarrow \quad \begin{array}{ll} \min & 5x_1 + (5+1)x_2 + 7y_2 \\ \text{s.t.} & x_1 \geq 100 \\ & y_1 + y_2 \geq 116.67 \end{array}$$

- Risultato: x_1=100, x_2=116.67, y_1=116.67, y_2=0.

La soluzione però non è ammissibile nel caso "freddo" e "molto freddo". In ogni caso conviene ricalcolare y_2 dopo che l'evento stocastico si è palesato

Clima	Azione stage 2	Costo totale
Normale	nessuna (eccesso di 16.67 in magazzino)	1200
Freddo	compra 3.33 unità al costo di 7 per unità	1223.33
Molto freddo	compra 13.33 unità al costo di 9 per unità	1320

- Il costo medio è 1247.78

Esempio

- Soluzione euristica: ottimizzare ogni scenario, poi prendere come variabili decisionali la media degli ottimizzatori:

$$x^*(j), y^*(j) = \arg\min \; 5x_1 + (5+1)x_2 + c(j)y_2 \quad \text{s.t.} \; x_1 \geq 100, \; y_1 + y_2 \geq d(j) \quad , \; j = 1,2,3$$

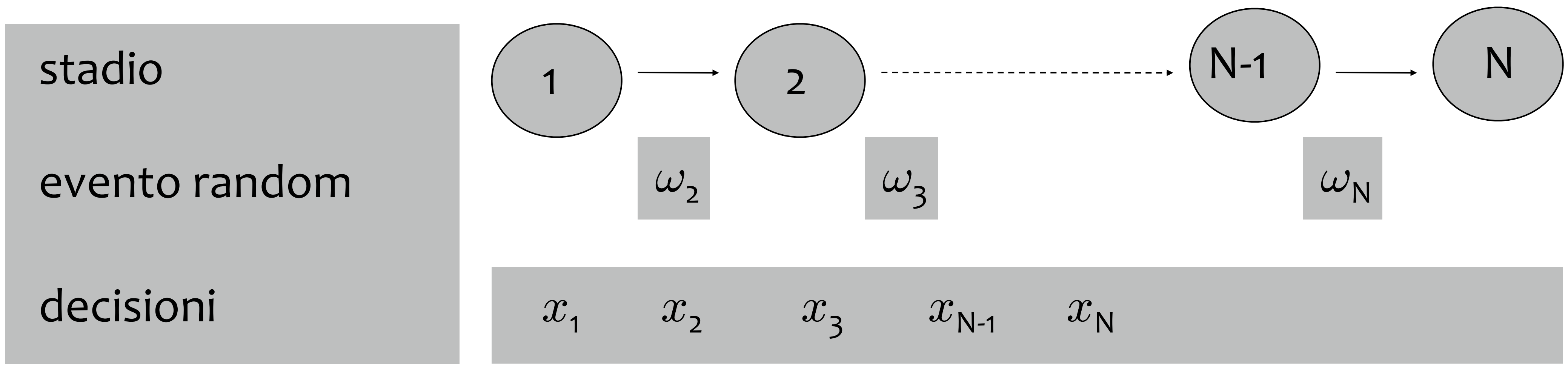
- Risultato: x_1=100, x_2=83.33, y_1=83.33, y_2=33.33.
La soluzione però non è ammissibile nel caso "molto freddo".
In ogni caso conviene ricalcolare y_2 dopo che l'evento stocastico si è palesato

Clima	Azione stage 2	Costo totale
Normale	compra 16.67 unità al costo di 5 per unità	1083.33
Freddo	compra 36.67 unità a 7 per unità	1256.67
Molto freddo	compra 46.67 unità a 9 per unità	1420

- Il costo medio è 1253.33

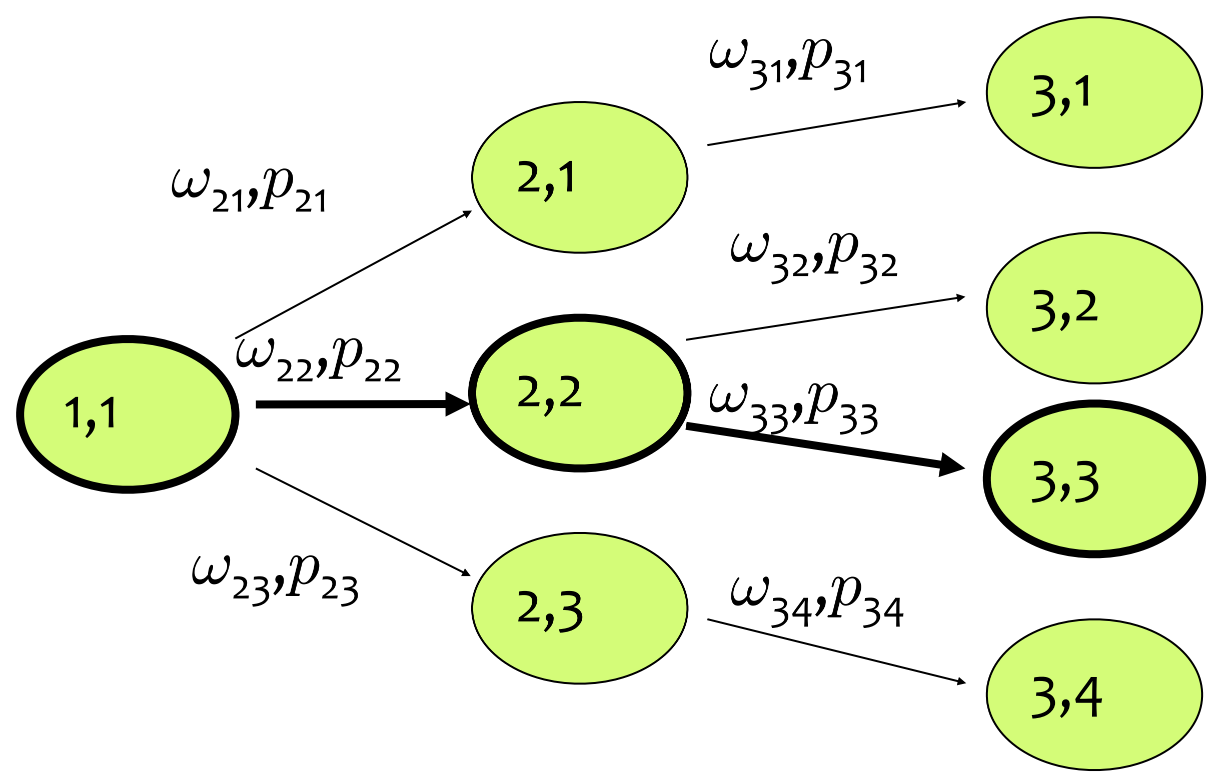
Programmazione stocastica (multi-stadio)

- Nella modellazione multi-stadio con ricorsione, la sequenza delle decisioni può essere rappresentata come segue:



- In questo caso si suppone che i possibili valori assumibili da ω siano organizzati secondo un "*albero*"

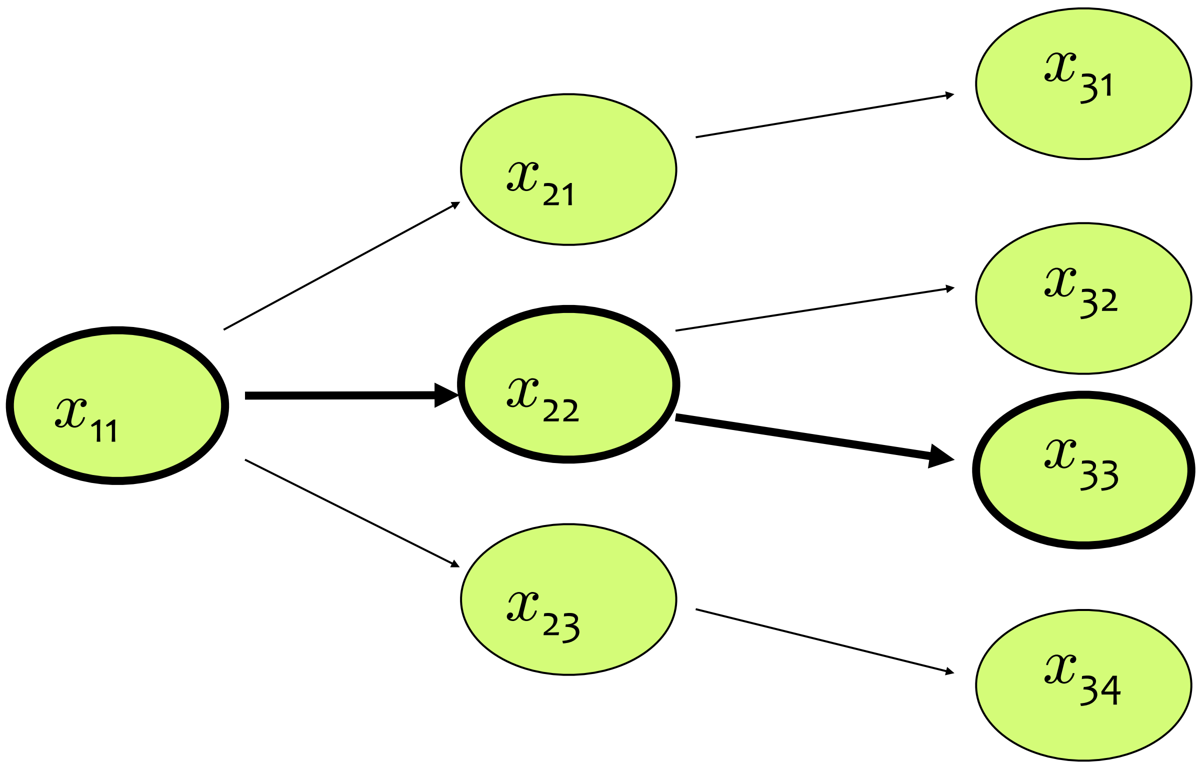
- Gli scenari sono dati da tutti i percorsi possibili sull'albero, ad esempio $[\omega_{22}, \omega_{33}]$ con probabilità $p=p_{22}p_{33}$.



Programmazione stocastica (multi-stadio)

- Ad ogni nodo dell'albero viene assegnata una variabile di decisione x_{ij} che rappresenta la decisione che sarebbe presa se ci venissimo a trovare nel nodo (i,j) dell'albero degli scenari

- x_{11} rappresenta la decisione presa prima che l'evento random si inizi a manifestare



- Con questo costrutto viene imposto un vincolo di **causalità** (o **non-anticipatività**) delle decisioni: la scelta x_{ij} presa allo stadio i, non conoscendo ancora quale sarà la realizzazione futura dell'evento random, è indipendendente da quale percorso verrà seguito negli stadi successivi i+1,...,N.

Linguaggi di modellistica per programmazione stocastica

- **MOSEL + Xpress-SP**, commerciale
- **SMI** (Stochastic Modeling Interface), open-source
- La maggior parte dei linguaggi di modellistica menzionati in precedenza (AMPL, GAMS, ecc.) supporta estensioni per la rappresentazione di problemi di ottimizzazione stocastica
- **SMPS** (Stochastic Mathematical Programming System), è una versione di MPS per rappresentare problemi di ottimizzazione stocastica

Bibliografia:

[1] J.R. Birge and F. Louveaux, "*Introduction to Stochastic Programming*", Springer Verlag, 1997

Fine