Università Politecnica delle Marche

Programmazione matematica

ver 3.0

Fabrizio Marinelli
fabrizio.marinelli@staff.univpm.it
tel. 071 - 2204823

I 3 passi della programmazione matematica

- Tre passi per costruire un modello di programmazione matematica:
  1. determinazione delle variabili decisionali
  2. definizione della funzione obiettivo
  3. definizione dei vincoli, cioè delle relazioni logiche tra le variabili di decisione che caratterizzano il problema

- In genere una scelta corretta delle variabili decisionali permette di esprimere in modo "naturale" funzione obiettivo e vincoli.

variabili decisionali

In alcuni casi (semplici) le variabili decisionali sono legate solo da evidenti vincoli « *tecnologici* ».

- in uno zaino 0-1 i valori assegnati alle variabili devono solo rispettare il vincolo di capacità;
- in un mix produttivo i valori assegnati alle variabili devono solo rispettare i limiti di disponibilità delle risorse.

In alcuni casi (semplici) le variabili decisionali sono legate solo da <u>evidenti</u> vincoli « *tecnologici* ».

In questi casi, ogni possibile assegnamento di valori a variabili ammette un'interpretazione plausibile (anche se non corrisponde a una soluzione ammissibile)

- in uno zaino 0-1 le variabili indicano una selezione di oggetti, quindi ogni assegnamento ha di per sé un significato

In altri casi i vincoli « *tecnologici* » sono meno evidenti.

Quando si associano elementi di un insiemi A a elementi di un insieme B, e l'associazione è di tipo **molti-a-uno**, si può utilizzare la seguente variabile

$x_{ij} = 1$ se l'elemento $i \in$ A è associato all'elemento $j \in$ B

però occorre introdurre il vincolo che garantisca la relazione molti-a-uno

$$\sum_{j \in B} x_{ij} = 1 \qquad \forall i \in A$$

In altri casi ancora, il significato delle variabili presuppone una struttura che deve essere opportunamente codificata.

La codifica deve essere fatta con vincoli che garantiscano la coerenza logica dei valori delle variabili, indipendentemente dagli aspetti « *tecnologici* » del problema.

In altri casi ancora, il significato delle variabili presuppone una struttura che deve essere opportunamente codificata.

In problemi di scheduling, le soluzioni descrivono sequenze di operazioni. Se la variabile usata è

$x_{ij} = 1$ se l'operazione i è la j -esima nella sequenza

occorre introdurre dei vincoli che garantiscano una relazione di tipo uno-a-uno tra operazioni e posizioni nella sequenza:

$$\sum_{j \in pos} x_{ij} = 1 \qquad \forall i \in op$$

$$\sum_{i \in op} x_{ij} = 1 \qquad \forall j \in pos$$

- Variabili binarie
- Tecniche di modellazione matematica
- ~~Problemi multi-obiettivo~~

Variabili binarie

Una **variabile binaria** y è una variabile decisionale che può assumere 2 valori (0 e 1). Una variabile binaria può descrivere:

- l'appartenenza di un elemento a un insieme $X \subseteq Y$ (*variabile indicatrice*)

$$y = \begin{cases} 1 & \text{se } x \in X \\ 0 & \text{se } x \notin X \end{cases}$$

- il valore di verità di una generica *proposizione logica elementare* A (*variabile logica*)

$$y = \begin{cases} 1 & \text{se A} \\ 0 & \text{se non A} \end{cases}$$

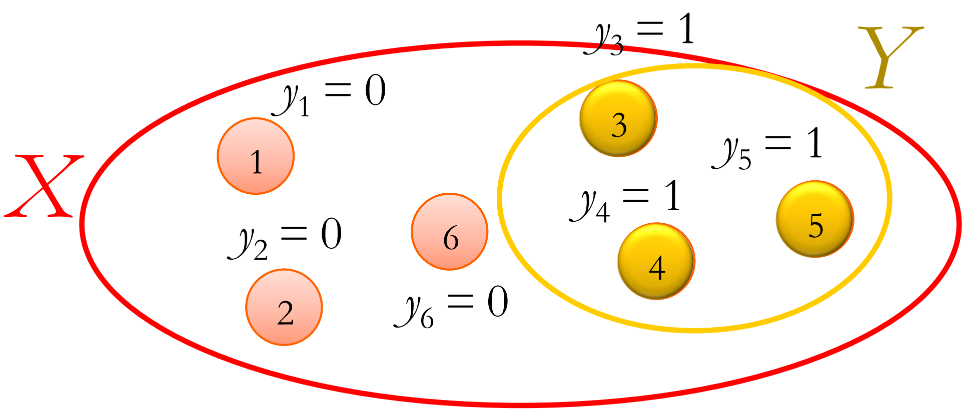
Variabile indicatrice

- Selezione
- Associazione

Variabili indicatrici: selezione

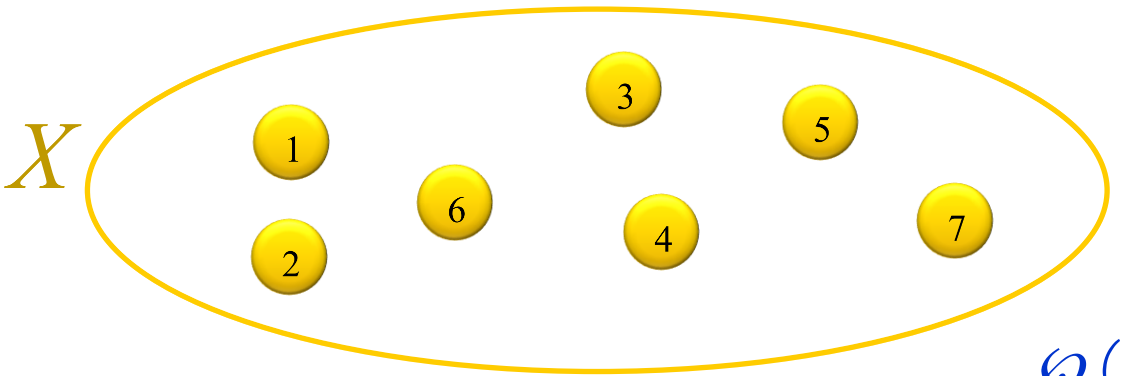
- Dato un insieme X discreto e finito (*insieme base*), la *selezione* di un sottoinsieme Y di elementi di X può essere descritta da un vettore $\mathbf{y} \in \{0,1\}^{|X|}$ di variabili binarie detto *vettore di incidenza* di Y.

Variabile indicatrice: $y_i = \begin{cases} 1 & \text{se } i \in Y \\ 0 & \text{se } i \in X \setminus Y \end{cases}$ $\qquad Y \subseteq X$



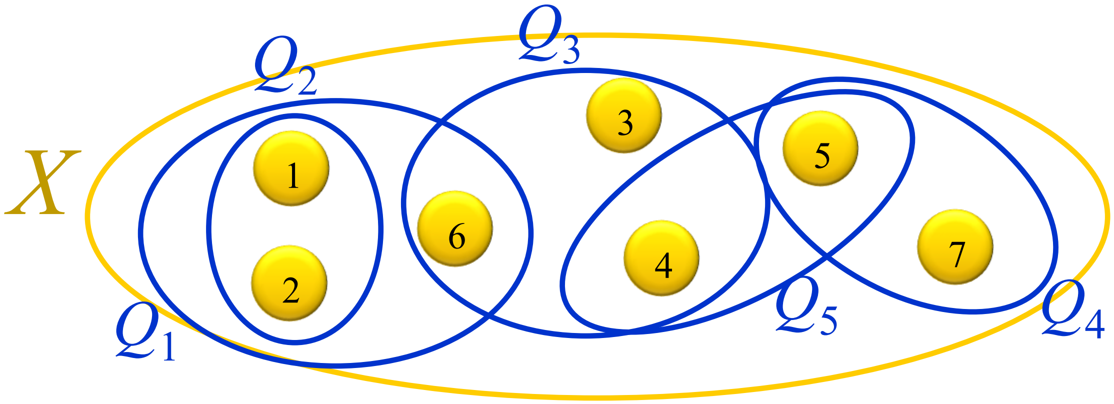
Problemi di ottimizzazione combinatoria

Un *problema di ottimizzazione combinatoria* in genere cerca tra elementi dell'insieme potenza $\wp(X)$ di un insieme finito e discreto X (*insieme base*).



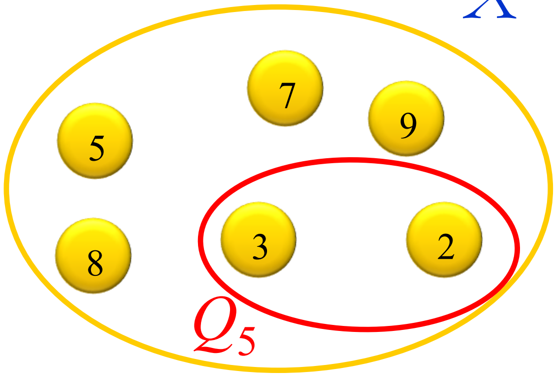
$\wp(X)$ è la famiglia dei 2^7 sottoinsiemi di X

Le soluzioni del problema si ottengono esaminando una particolare famiglia $\mathcal{Q}$ di sottoinsiemi di X ognuno dei quali soddisfa una data proprietà combinatoria.



Esempio: zaino 0-1

X = oggetti



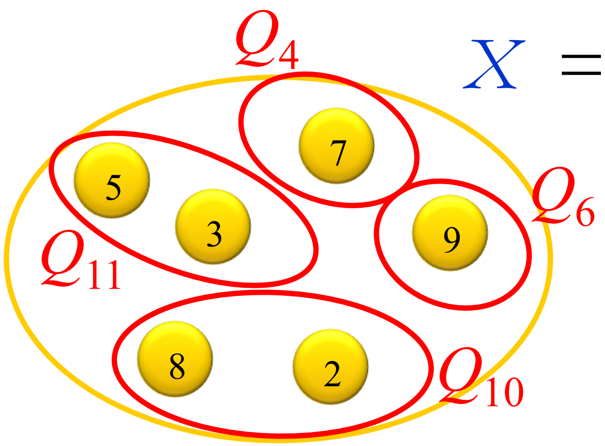
$\wp(X)$ = tutti i possibili 2^n sotto-insiemi di oggetti

Se la capacità dello zaino è 7 allora

$$Q = [\{2\}, \{3\}, \{5\}, \{7\}, \{2, 3\}, \{2, 5\}] \subset \wp(X)$$

La soluzione del problema è un unico elemento di Q che descrive la selezione di oggetti di X

Esempio: container loading



X = pallet

$\wp(X)$ = tutti i possibili 2^n sotto-insiemi di pallet

Se la capacità di un container è 10 allora i possibili carichi sono

$$\mathcal{Q} = [\{2\}, \{3\}, \{5\}, \{7\}, \{8\}, \{9\}, \{2, 3\}, \{2, 5\}, \{2, 7\}, \{2, 8\}, \{3, 5\}, \{3, 7\}, \{2, 3, 5\}] \subset \wp(X)$$

La soluzione del problema è una collezione di elementi di $\mathcal{Q}$ (carichi) che descrive una partizione di X (pallet)

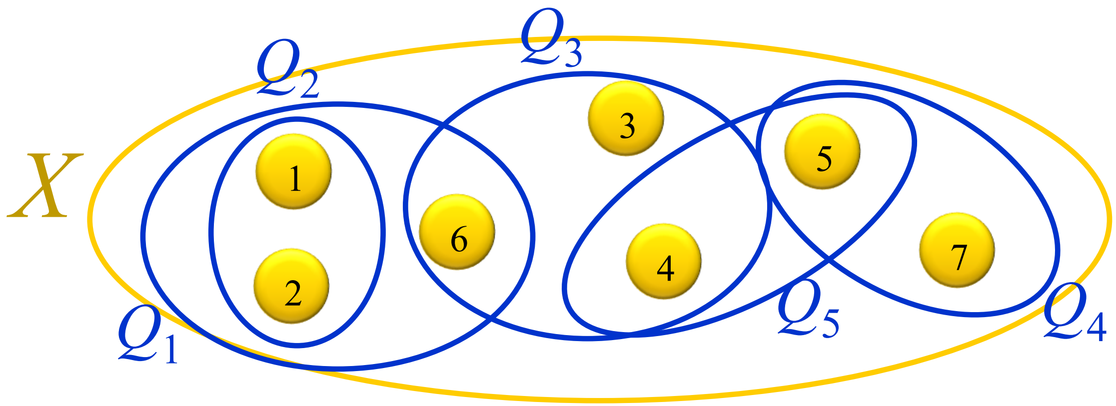
altri esempi

Problema	**Insieme base *X***	**Sottoinsiemi di *X***
Vehicle routing	Destinazioni	Rotte
Crew scheduling	Corse	Turni
Container loading	Pallet	Carichi
Parallel scheduling	Job	Schedule

Covering, packing and partitioning

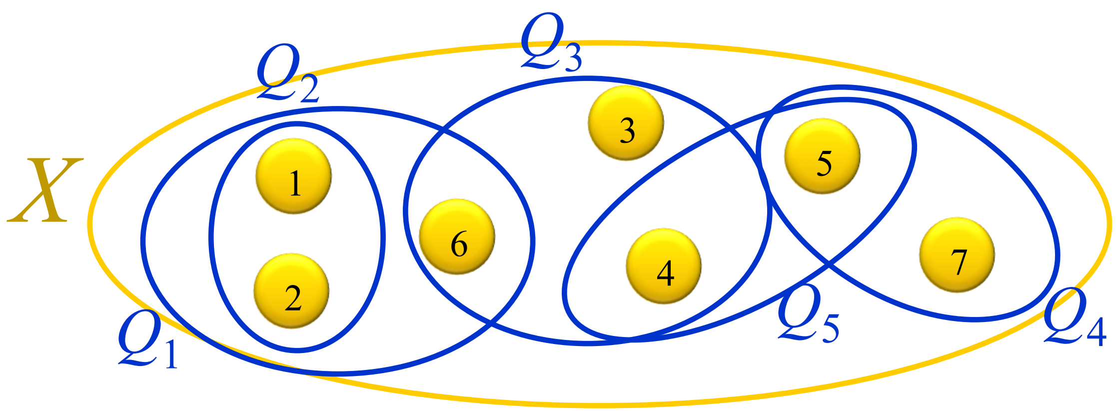
I problemi combinatorici spesso consistono nella selezione di elementi di $\mathcal{Q}$ che formano una *cover*, un *packing* o una *partizione* dell'insieme X.

Matrice di incidenza



X	Q_1	Q_2	Q_3	Q_4	Q_5
1	1	1			
2	2	2			
3			3		
4			4		4
5				5	5
6	6		6		
7				7	

Matrice di incidenza



$$
\begin{array}{c|c}
X & \begin{array}{ccccc} \mathcal{Q}_1 & \mathcal{Q}_2 & \mathcal{Q}_3 & \mathcal{Q}_4 & \mathcal{Q}_5 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} &
\left[\begin{array}{ccccc}
1 & 1 & & & \\
1 & 1 & & & \\
 & & 1 & & \\
 & & 1 & & 1 \\
 & & & 1 & 1 \\
1 & & 1 & & \\
 & & & 1 &
\end{array}\right]
\end{array}
$$

$\mathbf{E}(|X| \times |\mathcal{Q}|)$

matrice di incidenza degli elementi di X sugli elementi di $\mathcal{Q}$

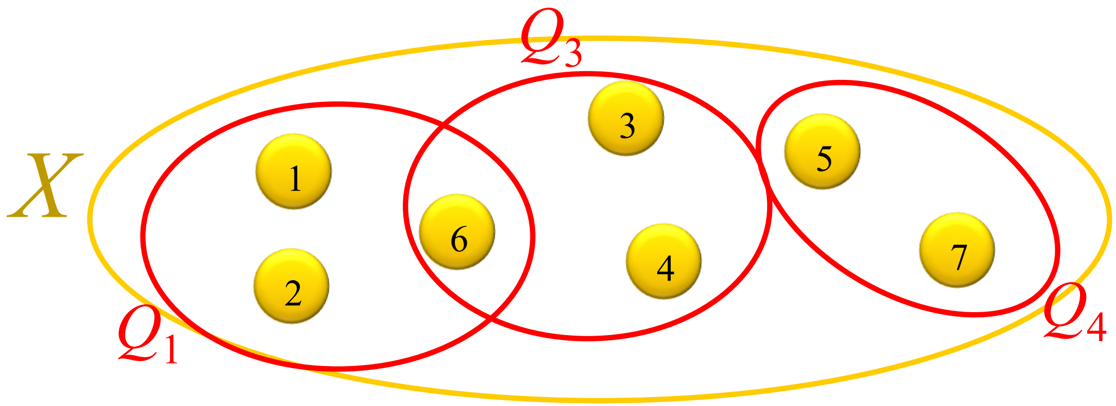
$$e_{ij} = 1 \text{ se e solo se } i \in \mathcal{Q}_j$$

Variabili di selezione

$$
\begin{array}{cc}
X & \begin{array}{ccccc} \mathcal{Q}_1 & \mathcal{Q}_2 & \mathcal{Q}_3 & \mathcal{Q}_4 & \mathcal{Q}_5 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} &
\begin{bmatrix}
1 & 1 & & & \\
1 & 1 & & & \\
 & & 1 & & \\
 & & 1 & & 1 \\
 & & & 1 & 1 \\
1 & & 1 & & \\
 & & & 1 &
\end{bmatrix} \\
 & \begin{array}{ccccc} y_1 & y_2 & y_3 & y_4 & y_5 \end{array}
\end{array}
$$

Variabile di *selezione*: $y_i = \begin{cases} 1 & \text{se l'elemento } \mathcal{Q}_i \text{ di } \mathcal{Q} \text{ è selezionato} \\ 0 & \text{altrimenti} \end{cases}$

Covering



- Un *covering* è una selezione $\{Q_1,\ldots,Q_p\}$ di elementi di Q (cioè di sottoinsiemi di X) tale che

$$\boxed{\bigcup_{j=1}^{p} Q_j = X}$$

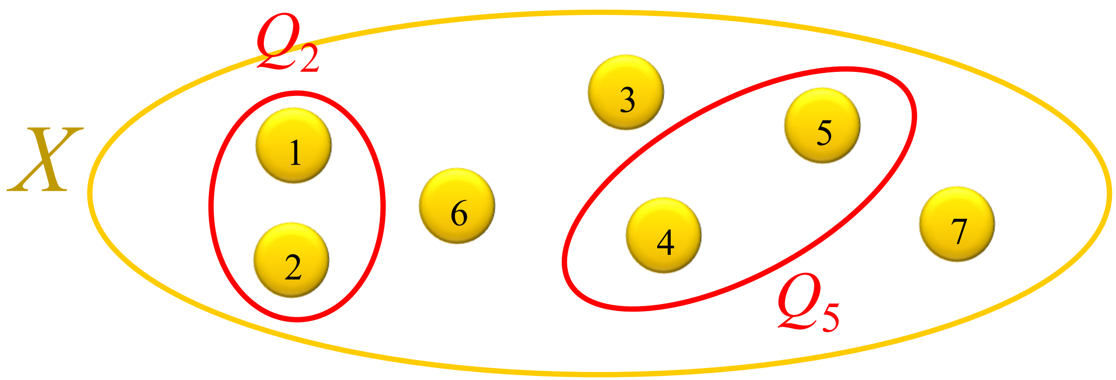
Un *covering minimo* esprime il concetto generale di <u>soddisfacimento</u> di richieste <u>al costo minimo</u>.

Covering

$$
\begin{array}{c|c}
X & \begin{array}{ccccc} Q_1 & Q_2 & Q_3 & Q_4 & Q_5 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} &
\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} \geq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}
\end{array}
$$

- $\mathbf{y} = (1,0,1,1,0)$ è il vettore di incidenza di un covering
- $\mathbf{y} \in \{0,1\}^{|\mathcal{Q}|}$ è un covering se e solo se $\mathbf{Ey} \geq \mathbf{1}$
- L'insieme $\wp(X)$ delle parti di X è banalmente un covering

Packing



- Un *packing* è una selezione $\{Q_1,\ldots,Q_p\}$ di elementi di Q (cioè di sottoinsiemi di X) tale che

$$\boxed{Q_i \cap Q_j = \emptyset \qquad \forall i \neq j}$$

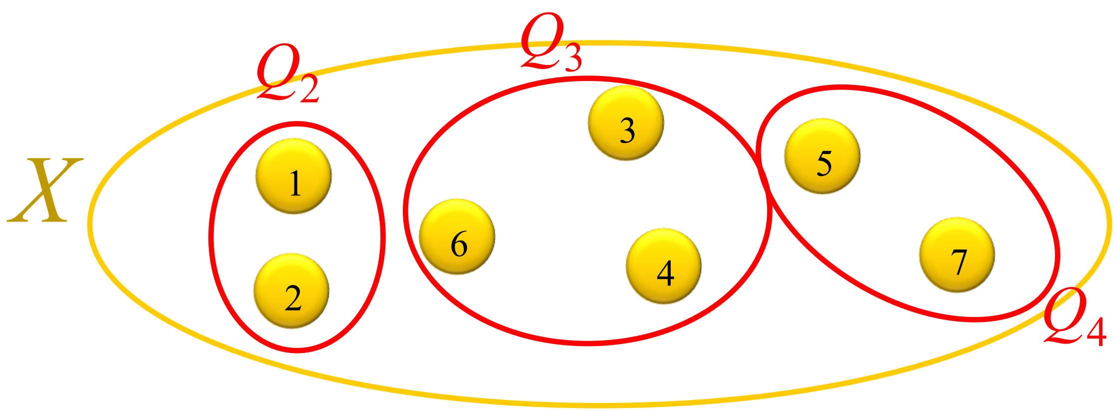
Un *packing massimo* esprime il concetto generale di utilizzo di risorse con massimo profitto.

Packing

$$
\begin{array}{c}
X \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7
\end{array}
\quad
\begin{array}{c}
\begin{array}{ccccc} Q_1 & Q_2 & Q_3 & Q_4 & Q_5 \end{array} \\
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}
\end{array}
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}
\leq
\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}
$$

- $\mathbf{y} = (0,1,0,0,1)$ è il vettore di incidenza di un packing
- $\mathbf{y} \in \{0,1\}^{|\mathcal{Q}|}$ è un packing se e solo se $\mathbf{Ey} \leq \mathbf{1}$
- $\varnothing$ è banalmente un packing

Partitioning



- Un *partitioning* è una selezione $\{\mathcal{Q}_1,\ldots,\mathcal{Q}_p\}$ di elementi di $\mathcal{Q}$ (cioè di sottoinsiemi di X) tale che

$$\begin{cases} \bigcup_{j=1}^{p} Q_j = X & \textit{covering} \\ Q_i \cap Q_j = \emptyset \quad \forall i \neq j & \textit{packing} \end{cases}$$

Partitioning

$$
\begin{array}{c}
 \\
1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7
\end{array}
\begin{array}{c}
\begin{array}{ccccc} X \quad Q_1 & Q_2 & Q_3 & Q_4 & Q_5 \end{array} \\
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}
\end{array}
\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}
=
\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}
$$

- $\mathbf{y} = (0,1,1,1,0,)$ è il vettore di incidenza di un partitioning
- $\mathbf{y} \in \{0,1\}^{|\mathcal{Q}|}$ è un partitioning se e solo se $\mathbf{Ey} = \mathbf{1}$

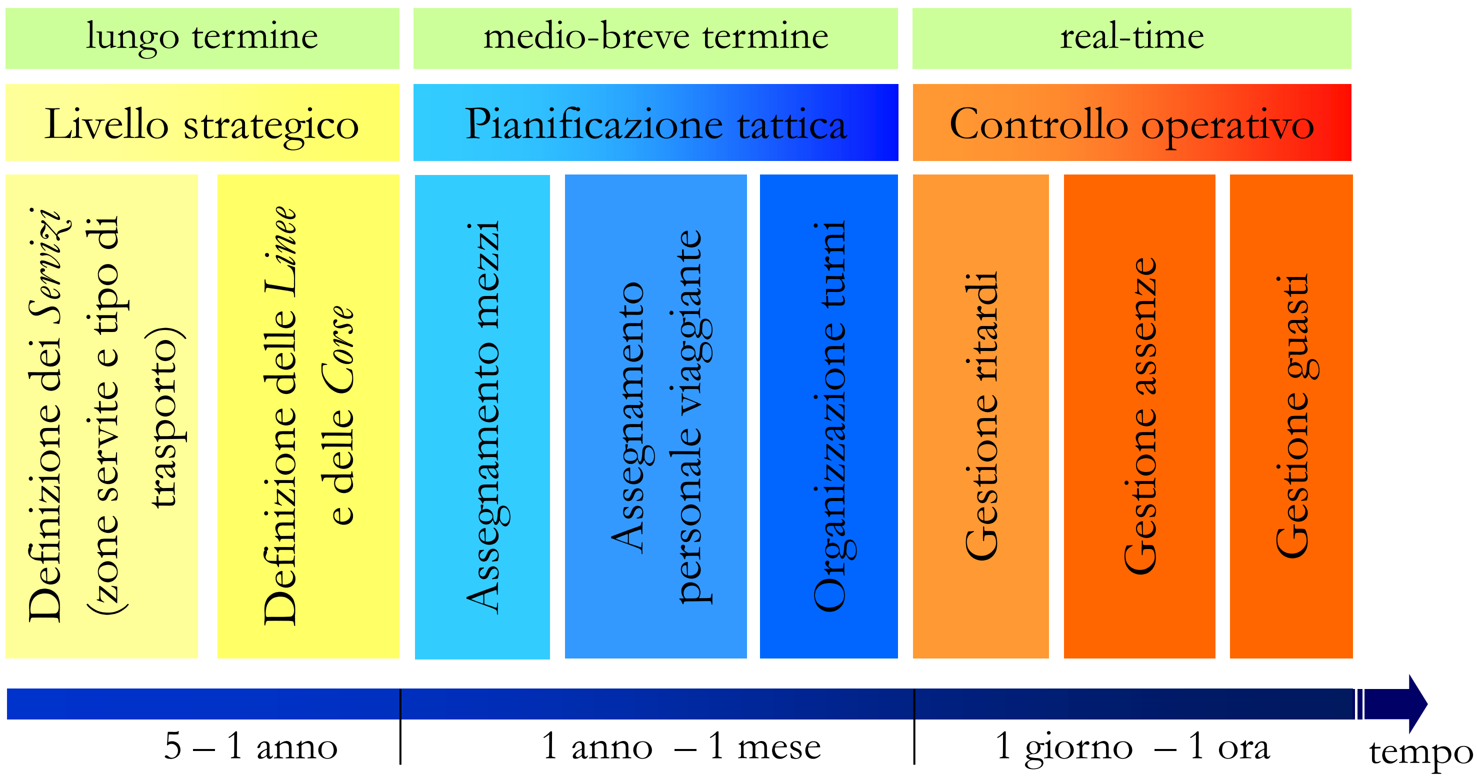
Un esempio:

Il problema di crew scheduling

Trasporto collettivo su gomma

- La gestione delle risorse (in particolare mezzi e personale viaggiante) nell'ambito del trasporto collettivo su gomma (urbano ed extra-urbano) è un'attività critica che richiede l'applicazione di tecniche avanzate di pianificazione.

- I principali problemi di pianificazione sono



Alcune definizioni

- **Linea:** lista di *località* (la prima e l'ultima delle quali sono chiamate *capolinea*) che vengono servite in sequenza da un unico mezzo.
- **Corsa**: segmento lavorativo indivisibile da coprire con un autista che corrisponde al tragitto tra 2 capolinea (di partenza e di arrivo), e **istante temporale in cui inizia**.
- **Turno uomo**: sequenza di corse effettuate da un conducente. Un turno inizia e termina, in genere, in una stessa località che corrisponde a un deposito

Pianificazione tattica

Definizione Tabelle Orarie

- Definizione delle *linee* e delle *corse* per ogni linea.

↓

Vehicle Scheduling

- Definizione dei *turni macchina*. Obiettivo primario: minimizzare il numero/costo dei mezzi utilizzati.

↓

Crew Scheduling

- Definizione dei *turni uomo*. Obiettivo primario: minimizzare il numero/costo del personale

↓

Crew Rostering

- Assegnamento dei turni uomo al personale. Obiettivo primario: soddisfare le esigenze dei dipendenti e bilanciare i carichi di lavoro

Pianificazione tattica: tabelle orarie

Definizione Tabelle Orarie

Vehicle Scheduling

Crew Scheduling

Crew Rostering

Corsa	Linea	Orario Partenza	Capolinea Partenza	Orario Arrivo	Capolinea Arrivo
1	A	7:22	Madonnetta	7:46	P.zza Cavour
2	B	7:51	P.zza Cavour	8:14	Posatora
3	C	7:51	P.zza Cavour	8:18	Tavernelle
4	A	7:54	Madonnetta	8:17	P.zza Cavour
5	C	8:19	Tavernelle	8:40	P.zza Cavour
6	B	8:20	Posatora	8:41	P.zza Cavour
7	C	8:21	P.zza Cavour	8:50	Tavernelle
8	A	8:25	P.zza Cavour	9:00	Madonnetta
9	C	8:30	Tavernelle	9:00	P.zza Cavour

Crew Scheduling Problem (CSP)

- **[Problema]** Data una tabella oraria (cioè un insieme di *corse* S) calcolare l'*insieme ottimale* di *turni* di autisti necessari per eseguire correttamente tutte le corse (cioè senza ritardi) nel rispetto di tutti i vincoli operativi.

- Possibili funzioni obiettivo:
  - costo totale dei turni dei conducenti
  - numero totale dei conducenti e/o di ore di straordinario

- Vincoli operativi tipici:
  - Ogni conducente inizia e finisce il turno al deposito di appartenenza
  - Ogni conducente ha un orario minimo e massimo di lavoro

Esempio

Corsa	Linea	Orario Partenza	Capolinea Partenza	Orario Arrivo	Capolinea Arrivo
1	A	7:22	Madonnetta	8:46	P.zza Cavour
2	B	8:51	P.zza Cavour	10:14	Posatora
3	C	8:54	Tavernelle	10:17	P.zza Cavour
4	A	10:34	Madonnetta	11:20	P.zza Cavour
5	C	11:32	P.zza Cavour	12:00	Tavernelle

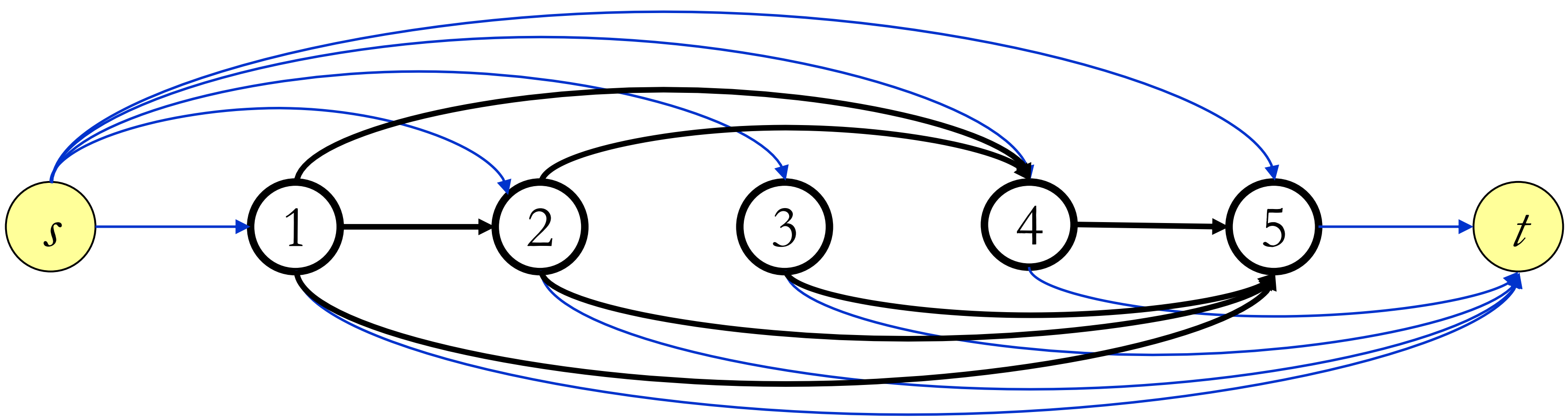
(*) A causa dei tempi di trasferimento, la corsa 3 non può essere preceduta né dalla 1 nè dalla 2 e non può precedere la 4

- soluzione 1 con 3 turni: (1,2), (3) e (4,5)
- soluzione 2 con 2 turni: (1,2,4,5), (3)

Esempio (cont.)

Corsa	Orario Partenza	Orario Arrivo
1	7:22	8:46
2	8:51	10:14
3	8:54	10:17
4	10:34	11:20
5	11:32	12:00

(*) A causa dei tempi di trasferimento, la corsa 3 non può essere preceduta né dalla 1 né dalla 2 e non può precedere la 4



Grafo di compatibilità

- Tutti i possibili turni possono essere calcolati analizzando i cammini del grafo di compatibilità

- **[Definizione]** il grafo di compatibilità $G(N, E)$ di una istanza di CSP è un grafo *diretto aciclico* in cui

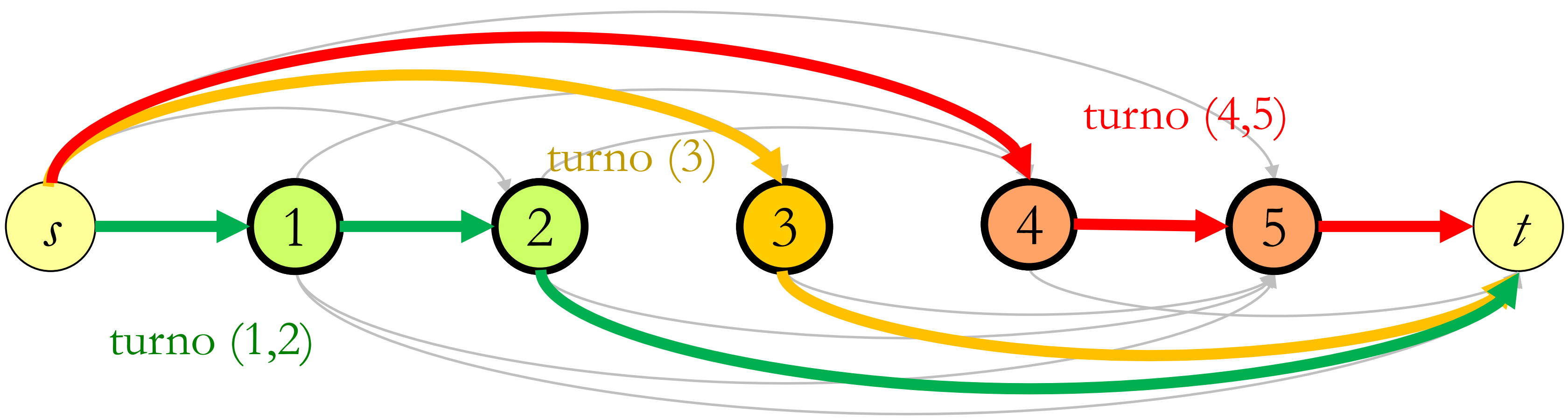
  - $N = S + \{s, t\}$ i nodi sono le corse; i nodi fittizi s e t rappresentano l'inizio e fine turno
  - $(i, j) \in E$ se la corsa j può seguire **immediatamente** la corsa i

- E' facile verificare che un *cammino orientato* da s a t descrive un turno ammissibile

Corsa	Orario Partenza	Orario Arrivo
1	7:22	8:46
2	8:51	10:14
3	8:54	10:17
4	10:34	11:20
5	11:32	12:00

soluzione 1

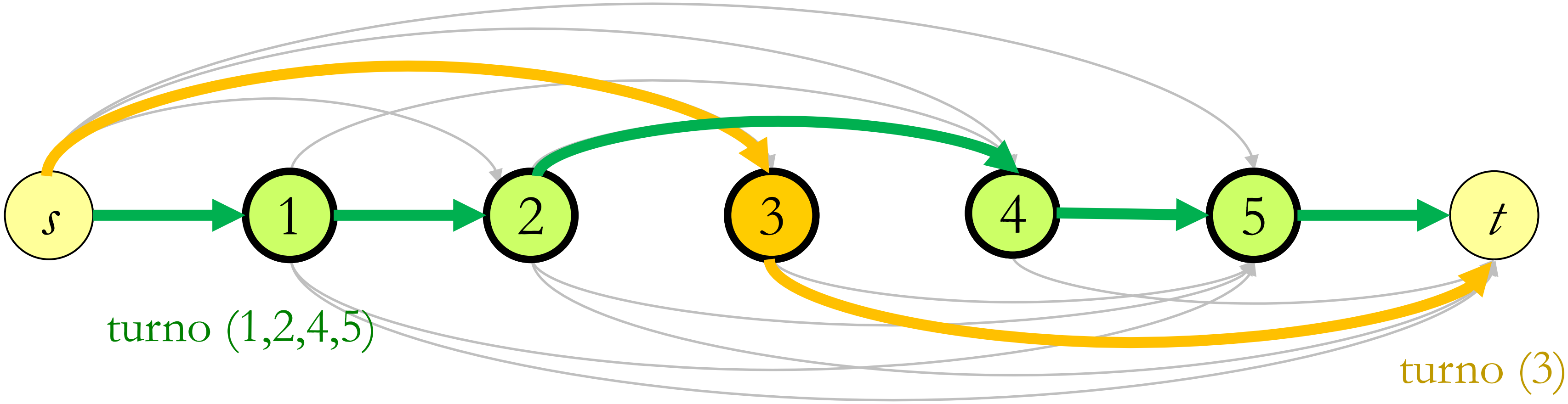
3 turni: (1,2), (3) e (4,5)

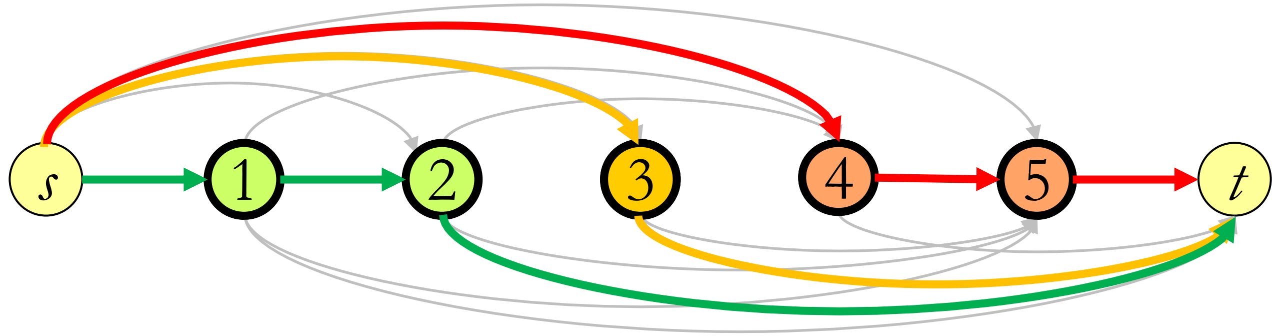


Corsa	Orario Partenza	Orario Arrivo
1	7:22	8:46
2	8:51	10:14
3	8:54	10:17
4	10:34	11:20
5	11:32	12:00

soluzione 2

2 turni: (3) e (1,2,4,5)





1. Un turno è un cammino, cioè un particolare sottoinsieme di nodi (corse)
2. Una soluzione è una collezione di cammini che "copre" tutti i nodi
3. Il problema CSP è un covering di nodi (corse) con cammini (turni).
4. La matrice di incidenza nodi cammini è

$$\mathbf{E} = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Modello matematico

Variabili e parametri

$$x_j = \begin{cases} 1 & \text{se il turno } j \text{ è scelto} \\ 0 & \text{altrimenti} \end{cases}$$

- c_j costo del turno j
- P insieme di tutti i turni possibili
- P_j j-esimo turno

Funzione obiettivo e vincoli

$$\min \sum_{j \in P} c_j x_j$$

$$\sum_{j \in P : i \in P_j} x_j \geq 1 \qquad \forall i \in S$$

$$x_j \in \{0,1\} \qquad \forall j \in P$$

in forma compatta

$$\begin{aligned} &\min \mathbf{c}^{\mathrm{T}} \mathbf{x} \\ &\quad \mathbf{Ex} \geq \mathbf{1} \\ &\quad \mathbf{x} \in \{0,1\}^{|P|} \end{aligned}$$

con $\mathbf{E}(|S| \times |P|)$ matrice di incidenza corse-turni

Esercizio: un problema di allocazione

- Definire un modello di PLI per il seguente problema:

  - Aruba offre risorse di calcolo (cloud computing), per semplicità supponiamo siano slot temporali ognuno pari a T sec.
  - Un calcolo richiede l'esecuzione di n jobs, ognuno senza interruzione. Le durate dei jobs sono $t_1,\ldots,t_n$ sec.

Quanti slot occorre acquistare?

Strategie di allocazione

- **First Fit (FF)**: assegna il prossimo job al primo slot sufficientemente capiente. Se non esiste, alloca un nuovo slot.

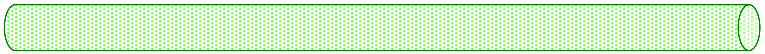
```
for i := 1 to n do
    pos(i) := argmin_{j∈N} { Σ_{pos(h)=j} t_h + t_i ≤ T }
end for
```

$$\textbf{for } i := 1 \textbf{ to } n \textbf{ do} \quad \mathrm{pos}(i) := \arg\min_{j \in N} \left\{ \sum_{pos(h)=j} t_h + t_i \leq T \right\} \quad \textbf{end for}$$

- **First Fit Decreasing (FFD)**: ordina i job per tempi di processamento non crescenti e applica FF.

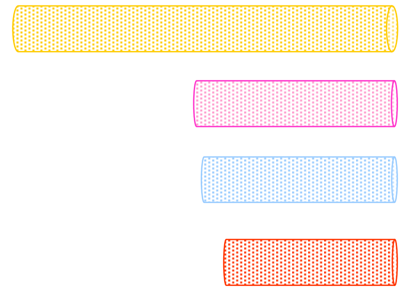
- **Best Fit Decreasing (BFD)**: ordina i job per tempi di processamento non crescenti. Assegna il job con la durata maggiore allo slot con il **tempo residuo più piccolo**, ma comunque sufficiente a eseguire il job. Se non esiste, alloca un nuovo slot.

FFD e BFD non sono strategie ottime



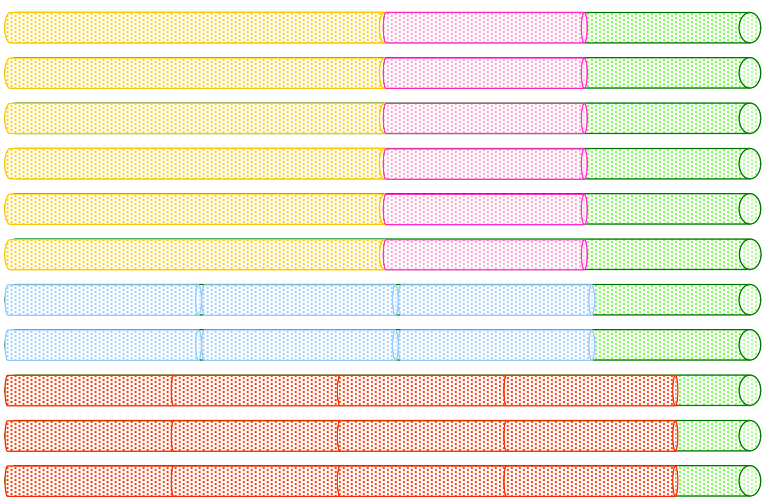
slot size = 100 sec.

jobs

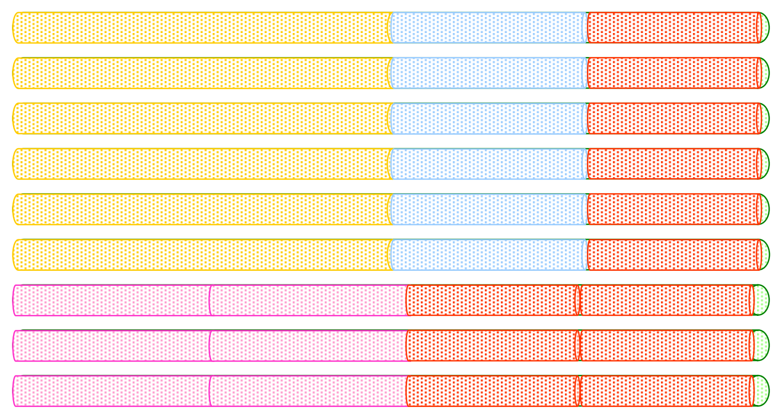


durata (sec.)	# copie
51	6
27	6
26	6
23	12

Soluzione FFD: 11 slots



Soluzione ottima: 9 slots



Modello di programmazione matematica …

… un primo tentativo

- Modello di assegnamento (L. Kantorovich, 1939).

Decisione: assegnamento di job a slot

$$y_j = \begin{cases} 1 \text{ se si utilizza lo slot } j \\ 0 \text{ altrimenti} \end{cases} \qquad x_{ij} = \begin{cases} 1 \text{ se job } i \text{ allocato in slot } j \\ 0 \text{ altrimenti} \end{cases}$$

Nel caso peggiore uso tanti slot quanti sono i job

$$z^* = min \sum_{j=1}^{n} y_j$$ minimizza il numero di slot utilizzati

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1, \dots, n$$ allocazione di tutti i job

$$\sum_{i=1}^{n} t_i x_{ij} \leq T y_j \qquad j = 1, \dots, n$$ ammissibilità dell'allocazione

- L'utilizzo dello slot j non può essere superiore alla sua durata T
- $x_{ij} > 0$ implica $y_j = 1$

Codifica in AMPL …

```
param T;
param n;
param m;

set SLOTS := 1..m;
set JOBS := 1..n;

param t{i in JOBS};

var x {i in JOBS,j in SLOTS} binary;
var y {j in SLOTS} binary;

minimize n_slots: sum {j in SLOTS} y[j];

s.t. demand {i in JOBS}: sum {j in SLOTS}x[i,j] = 1;
s.t. allocation {j in SLOTS}: sum {i in JOBS}(t[i]*x[i,j]) <= T*y[j];
```

…un caso più interessante

slot size: 10000 sec.

jobs

durata (sec.),	#istanze
3834,	10
1506,	22
204,	89
3408,	83
6102,	75
1755,	67
5653,	18
4809,	15
4286,	2
3465,	30
3550,	16
3658,	28
6753,	4
1090,	29
5245,	6

First Fit Decreasing (FFD): 155 slot

Soluzione ottima: 152 slot

Esercizio: un problema di allocazione

- Aruba offre risorse di calcolo (cloud computing), per semplicità supponiamo siano slot temporali ognuno pari a T sec.
- Un calcolo richiede l'esecuzione di n jobs, ognuno senza interruzione. Le durate dei jobs sono $t_1,\ldots,t_n$ sec.

Quanti slot occorre acquistare?

Il problema può essere espresso in termini di covering, packing o partitioning? Come?

Un altro modello di programmazione matematica

- Modello di selezione (P.C. Gilmore, R.E. Gomory,1961)

Decisione: **selezione** di sottoinsiemi di job

X = job

$\wp(X)$ = tutti i possibili sottoinsiemi di job

$\mathcal{Q}$ = famiglia di sottoinsiemi ammissibili di job (cioè con durata totale non superiore alla durata di uno slot)

$$y_j = \begin{cases} 1 & \text{se si seleziona il sottoinsieme } j \in Q \\ 0 & \text{altrimenti} \end{cases}$$

$$z^* = min \sum_{j \in Q} y_j$$

partizione di cardinalità minima.

$$\sum_{j \in Q} a_{ij} y_j = 1 \qquad i \in X$$

Una soluzione è una partizione di X.

$= 1$ se il job i è nel sottoinsieme j, 0 altrimenti

- $|Q|$ è dell'ordine di 2^n (numero esponenziale di variabili)

Esempio

slot size: 100 sec.

jobs
durata (sec.), #copie
51, 2
27, 2
26, 3
23, 1

una soluzione che usa 4 slot:

$y_1 = 1$: slot con job 1 e 3

$y_5 = 1$: slot con job 2 e 4

$y_9 = 1$: slot con job 5, 6 e 8

$y_{|Q|} = 1$: slot con job 7

		y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}		$y_{\|Q\|}$		
job 1	51	1					1								1
job 2	51					1		1			1				1
job 3	27	1	1	1					1		1				1
job 4	27		1			1						…		=	1
job 5	26		1	1	1					1					1
job 6	26			1					1	1					1
job 7	26				1		1	1	1				1		1
job 8	23				1		1	1		1					1

- In questo caso anche un covering sarebbe andato bene. Perchè?

Esercizio

- Definire un modello di PLI per il seguente problema:

  - Un corriere espresso vuole dotarsi di una flotta di furgoni.
  - Il tipo di furgone scelto ha una portata massima di Q kg e, per semplicità, non ha problemi di capienza (ovvero, ogni furgone potrebbe contenere, in termini volumetrici, una qualsiasi combinazione di colli).
  - La previsione di consegna giornaliera è di n colli che pesano rispettivamente $p_1,\dots,p_n$ Kg.

  Quanti furgoni servono?

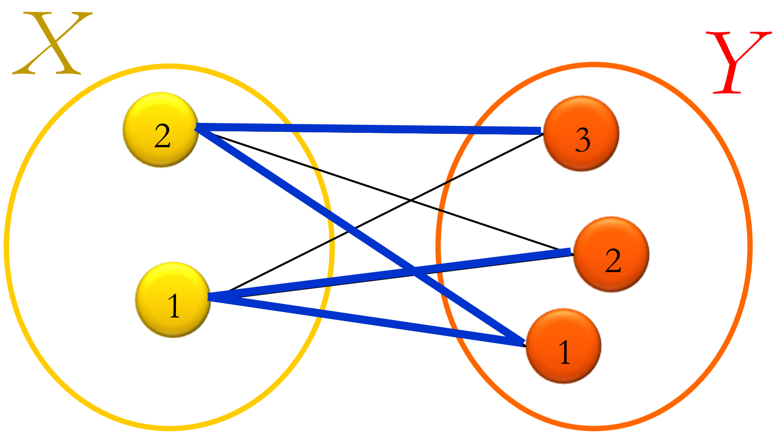
Variabile indicatrice

- Selezione
- Associazione

Variabili indicatrici: associazione

Dati due insiemi X e Y discreti e finiti, l'*associazione* tra elementi di X e Y è una selezione di elementi nell'insieme $X \times Y$ (*insieme base*).

Variabile indicatrice: $y_{ij} = \begin{cases} 1 & \text{se } i \in X \text{ è associato a } j \in Y \\ 0 & \text{altrimenti} \end{cases}$



$\mathbf{y_{11} = 1}$

$\mathbf{y_{12} = 1}$

$y_{13} = 0$

$\mathbf{y_{21} = 1}$

$y_{22} = 0$

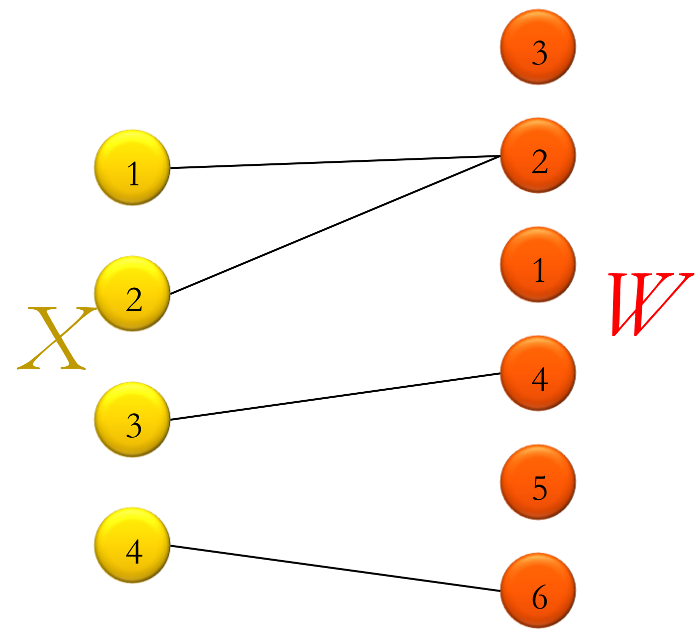
$\mathbf{y_{23} = 1}$

Associazione e assegnamento

- Un'**associazione** in generale può esprimere una *biezione* oppure un *assegnamento molti-a-uno*

$$\boxed{\sum_{j \in W} y_{ij} = 1}$$

$i \in X$ <u>deve</u> essere associato a <u>un solo</u> elemento di W, ma ogni elemento di W può essere associato a zero, uno o più elementi di X



Associazione e ordinamento

- Un'**associazione** in generale può esprimere un *ordinamento* degli elementi di X

  Un ordinamento è una *relazione d'ordine* $\prec$, ossia una relazione che soddisfa le proprietà

antisimmetrica	$x \prec y \wedge y \prec x \Longrightarrow x = y$
transitiva	$x \prec y \prec w \Longrightarrow x \prec w$
riflessiva	$x \prec x$

  Se la proprietà antisimmetrica è vera per tutte le coppie di elementi di X si parla di ordine *totale* (o anche di *permutazione senza ripetizione* $\sigma(X)$), altrimenti l'ordine si dice *parziale*.

Variabili indicatrici e permutazioni

- Una permutazione degli elementi di un insieme X può essere descritta definendo diversi tipi di associazione:

Alternative

1. associazione di elementi a elementi
   si stabilisce la posizione relativa di ogni elemento rispetto agli altri, cioè si decide chi precede chi (immediatamente o meno).
2. associazione di elementi a posizioni
   si stabilisce la posizione assoluta di ogni elemento.

Permutazioni: chi precede chi

1. si decide chi precede chi (non in senso stretto):

$$x_{ij} = \begin{cases} 1 \text{ se l'elemento } i \text{ precede l'elemento } j \\ 0 \text{ altrimenti} \end{cases}$$

Ogni sequenza ha una rappresentazione in termini di x_{ij}

[Esempio]

a	c	b	d
$x_{ab} = 1$	$x_{ca} = 0$	$x_{ba} = 0$	$x_{da} = 0$
$x_{ac} = 1$	$x_{cb} = 1$	$x_{bc} = 0$	$x_{db} = 0$
$x_{ad} = 1$	$x_{cd} = 1$	$x_{bd} = 1$	$x_{dc} = 0$

Ma quali sono gli assegnamenti di valori alle x_{ij} che **coerentemente** rappresentano sequenze?

- Proprietà antisimmetrica: $x \prec y \wedge y \prec x \Rightarrow x = y$
  per ogni coppia di elementi i e j, **esattamente** una delle due: i precede j **oppure** j precede i

  la condizione logica è $\boxed{x_{ij} \text{ XOR } x_{ji}}$ che si esprime con i vincoli

$$x_{ij} + x_{ji} = 1 \qquad \forall i, j \in A, i \neq j$$

Ma quali sono gli assegnamenti di valori alle x_{ij} che **coerentemente** rappresentano sequenze?

- Proprietà transitiva: $x \prec y \prec w \Rightarrow x \prec w$
  se i precede j e j precede k allora i precedere k

  la condizione logica è $\boxed{x_{ij} \text{ AND } x_{jk} \Rightarrow x_{ik}}$ che si esprime con i vincoli

$$x_{ij} + x_{jk} + x_{ki} \leq 2 \qquad \forall i, j, k \in A, i \neq j \neq k$$

Ma quali sono gli assegnamenti di valori alle x_{ij} che **coerentemente** rappresentano sequenze?

- Proprietà riflessiva: $x \prec x$
  assumendo che ogni elemento i preceda se stesso la proprietà si esprime coi vincoli

$$x_{ii} = 1 \qquad \forall i \in A$$

Alternativa 1: chi precede chi (non in senso stretto)

$$x_{ij} = \begin{cases} 1 \text{ se l'elemento } i \text{ precede l'elemento } j \\ 0 \text{ altrimenti} \end{cases}$$

$$x_{ij} + x_{ji} = 1 \qquad \forall i, j \in A, i \neq j$$

$$x_{ij} + x_{jk} + x_{ki} \leq 2 \qquad \forall i, j, k \in A, i \neq j \neq k$$

$$x_{ii} = 1 \qquad \forall i \in A$$

Esercizio

Quali vincoli servono se si decide di utilizzare la variabile decisionale $x_{ij} = 1$ se l'elemento i precede <u>immediatamente</u> l'elemento j ?

Permutazioni: posizioni assolute

2. Si stabilisce la posizione assoluta di ogni elemento:

$$x_{ij} = \begin{cases} 1 \text{ se l'elemento } i \text{ è in posizione } j \\ 0 \text{ altrimenti} \end{cases}$$

Ogni sequenza ha una rappresentazione in termini di x_{ij}

[Esempio]

a	c	b	d
$x_{a1} = 1$	$x_{a2} = 0$	$x_{a3} = 0$	$x_{a4} = 0$
$x_{b1} = 0$	$x_{b2} = 0$	$x_{b3} = 1$	$x_{b4} = 0$
$x_{c1} = 0$	$x_{c2} = 1$	$x_{c3} = 0$	$x_{c4} = 0$
$x_{d1} = 0$	$x_{d2} = 0$	$x_{d3} = 0$	$x_{d4} = 1$

Ma quali sono gli assegnamenti di valori alle x_{ij} che **coerentemente** rappresentano sequenze?

- Ogni elemento deve occupare **esattamente** una posizione

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \forall i \in A$$

- Ogni posizione è occupata **esattamente** da un elemento

$$\sum_{i \in A} x_{ij} = 1 \qquad \forall j = 1, \ldots, n$$

Alternativa 2: elementi a posizioni

$$x_{ij} = \begin{cases} 1 \text{ se l'elemento } i \text{ è in posizione } j \\ 0 \text{ altrimenti} \end{cases}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \forall i \in A$$

$$\sum_{i \in A} x_{ij} = 1 \qquad \forall j = 1, \dots, n$$

Permutazioni: posizioni assolute (variante)

2.b Si stabilisce la posizione assoluta di ogni elemento:

$x_i = j$ se l'elemento i è in posizione j con $1 \leq j \leq n$

Ogni sequenza ha una rappresentazione in termini di x_{ij}

[Esempio]

a	c	b	d
$x_a = 1$	$x_c = 2$	$x_b = 3$	$x_d = 4$

Ma quali sono gli assegnamenti di valori alle x_i che **coerente-mente** rappresentano sequenze?

dovrei esprimere la condizione $x_i \neq x_j \; \forall i, j$ con $i \neq j$.
Ma la relazione $x \neq y$ non è di tipo lineare.

$$\boxed{\textbf{?}}$$

Permutazioni: discussione

- **Alternativa 1**: $x_{ij} = 1$ se i precede j — circa n^2 variabili binarie
- **Alternativa 2**: $x_{ij} = 1$ se i è in posizione j — n^2 variabili binarie
- **Alternativa 2.b**: $x_i = j$ se i è in posizione j — n variabili intere

L'alternativa 2.b sembra la scelta più conveniente se si guarda alla dimensione del modello.

L'alternativa 2.b è la scelta migliore?

Esercizi

1. Modellare la condizione $x \neq y$ con soli vincoli lineari (equazioni/disequazioni) ed eventuali variabili ausiliarie reali/intere.

2. Si assuma di pagare un costo $c_{ij} > 0$ per ogni coppia di elementi i,j consecutivi nella sequenza. Per ognuna delle alternative viste in precedenza, esprimere la funzione obiettivo (eventualmente con l'aiuto di vincoli lineari e variabili ausiliarie) che determina la sequenza di costo minimo.

Permutazioni e problemi di scheduling

- Un problema di scheduling in generale consiste (anche) nel mettere in *sequenza* gli n elementi (compiti, lavorazioni, …) di un insieme $A = \{a, b, c, \ldots\}$.
- Una soluzione ammissibile è una sequenze degli n elementi $a, b, c, \ldots$ quindi lo spazio di ricerca è in generale compreso nell'insieme di tutte le permutazioni degli elementi in A

Ottimizzazione del ciclo di produzione

quando si passa da un item al successivo viene effettuato un setup (e si incorre in un tempo di attrezzaggio)

Come si calcola la miglior (più breve) sequenza di operazioni ?

Esercizio: il miglior ciclo di produzione

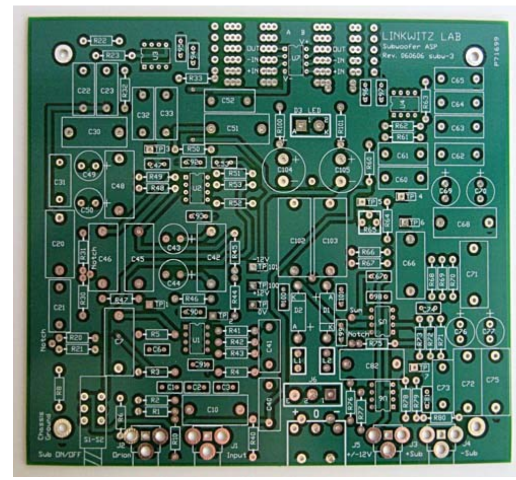
tempi di attrezzaggio in una produzione di 10 items

	1	2	3	4	5	6	7	8	9	10
1	0	2	4	5	3	2	2	3	3	5
2	4	0	1	2	1	3	5	2	5	2
3	2	3	0	2	5	2	1	5	3	3
4	3	2	5	0	2	3	5	2	4	4
5	2	3	3	3	0	4	2	4	5	3
6	2	1	2	5	2	0	1	2	3	1
7	1	5	4	3	3	2	0	3	1	5
8	5	2	3	4	2	5	3	0	2	1
9	3	1	4	2	5	3	4	2	0	2
10	4	2	3	1	3	1	5	4	5	0

Qual è la sequenza ottima ?

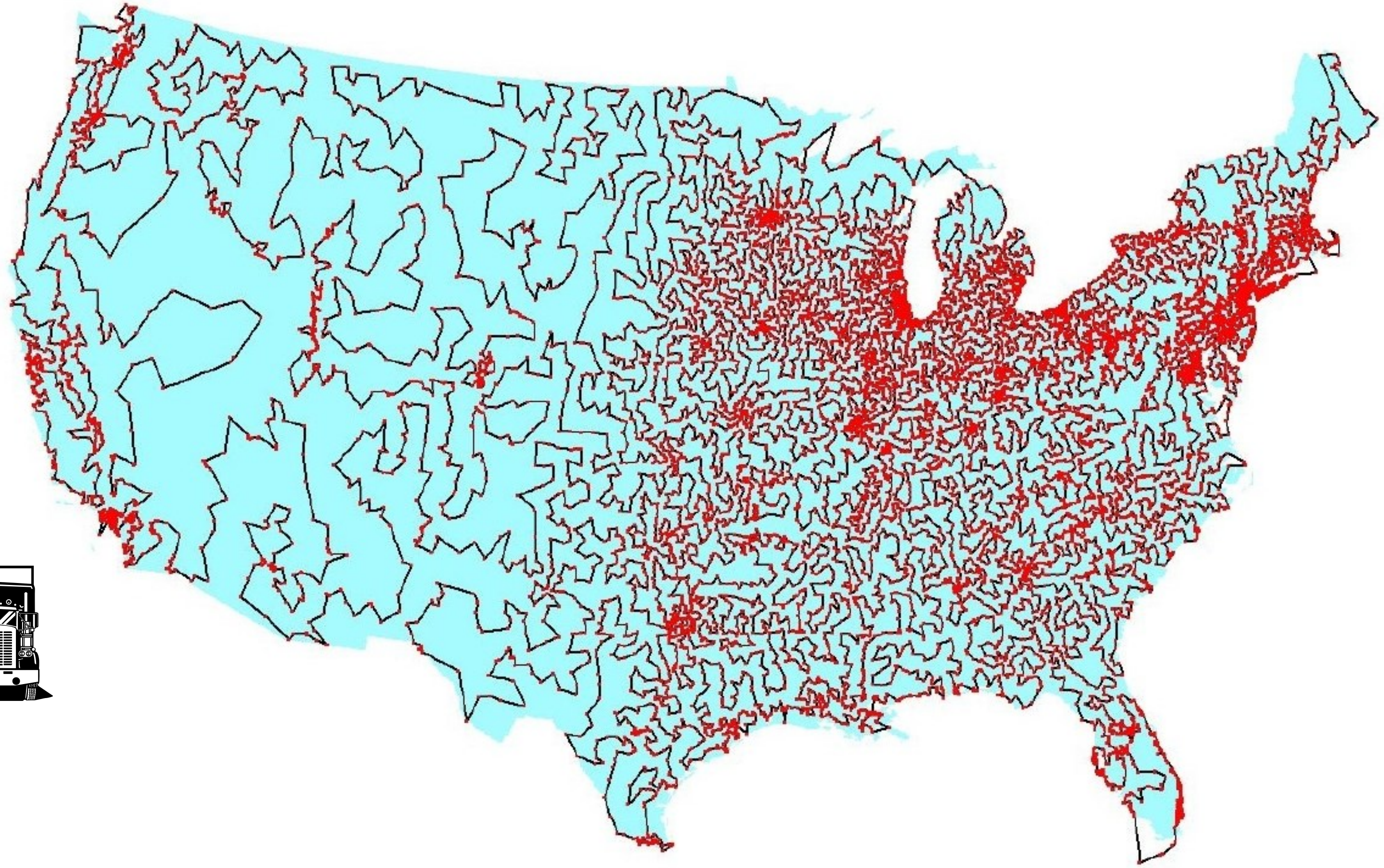
Qual è il ciclo ottimo ?

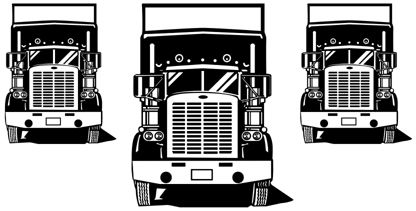
Smart manufacturing



- Dato un PCB (*Printed Circuit Board*) qual è la sequenza di assemblaggio (o di foratura) più efficiente ?

Instradamento di veicoli



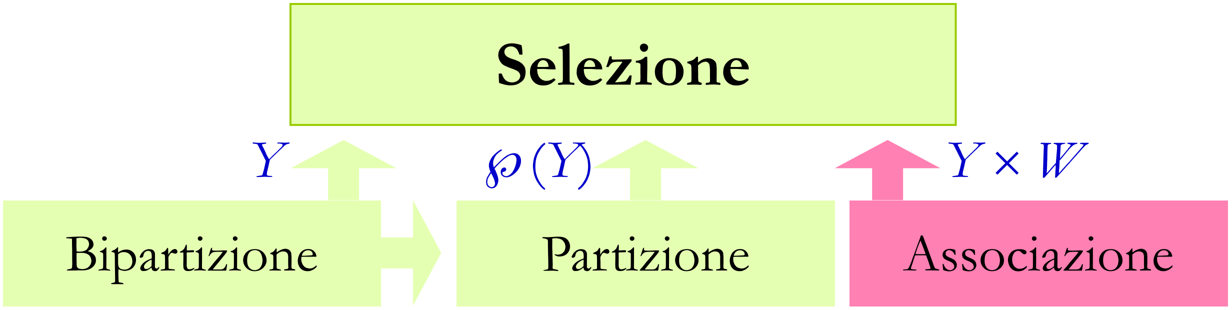


- Questo è il “miglior giro di consegna” che tocca le 13.509 città americane con più di 500 abitanti (calcolato nel 1998)

Esercizi

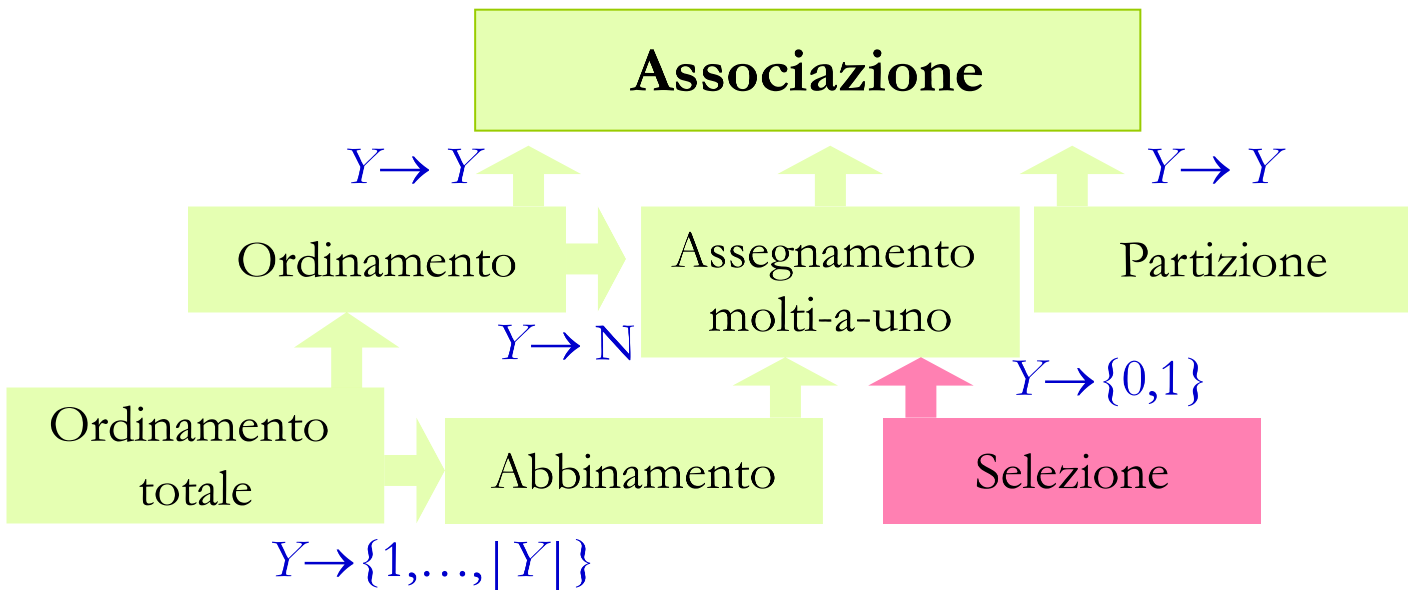
- Modellare con variabili di associazione una funzione iniettiva e una funzione suriettiva su insiemi discreti e finiti

Riduzione tra paradigmi



- Un'associazione tra Y e W è una selezione di elementi in $Y \times W$
- Una partizione di Y è una selezione in $\wp(Y)$
- Una bipartizione di Y è una selezione di un elemento di $\wp(Y)$

Riduzione tra paradigmi



- Un ordinamento è un'associazione tra coppie di elementi o un associazione di elementi a posizioni
- Una partizione è un'associazione degli elementi in parti
- Una selezione è un assegnamento dei valori $\{0,1\}$ agli elementi di Y
- Un abbinamento è un particolare assegnamento molti-a-uno
- Un ordinamento totale è un particolare ordinamento che esprime un abbinamento tra coppie di elementi in <u>stretta successione</u> o tra elementi e posizioni

Variabile logica

- Costi fissi
- Variabili semi-continue
- Vincoli condizionali
- Predicati logici

Variabili logiche

Utilizzate per modellare

- Costi fissi: costo con una componente fissa (costo di attrezzaggio, di trasporto) e una variabile (costo di produzione, di acquisto)
- Variabili semi-continue: quantità che può essere nulla o superiore a una determinata soglia minima (ordini di acquisto)
- Vincoli condizionali: alcuni vincoli subentrano solo se vengono fatte determinate scelte
- Predicati logici: per esempio, **se** *a* e *b* sono fatti veri **allora** *c* deve essere anch'esso un fatto vero

Applicazioni: un esempio classico

- La società *Merlin* produce i concimi *prato starter* (tipo A) e *prato estate* (tipo B) che vende rispettivamente a 25 e 28 €/Kg. Considerando la composizione dei singoli concimi e le disponibilità in magazzino (vedi tabella) qual è il guadagno massimo che si può ottenere producendo i concimi di tipo A e B?

	composizione			
	Azoto	Fosforo	Potassio	Magnesio
tipo A	40%	40%	10%	10%
tipo B	24%	45%	31%	-
disponibilità (Kg)	312	360	160	70

Applicazioni: un esempio classico

variabili decisionali:

$x_A \in \mathbf{R}$ = quantità (in Kg) che si decide di produrre del concime di tipo A
$x_B \in \mathbf{R}$ = quantità (in Kg) che si decide di produrre del concime di tipo B

$$z = \max 25x_A + 28x_B$$

$$\begin{aligned}
0.4x_A + 0.24x_B &\le 312 \\
0.4x_A + 0.45x_B &\le 360 \\
0.1x_A + 0.31x_B &\le 160 \\
0.1x_A &\le 70 \\
x_A, x_B &\ge 0
\end{aligned}$$

Obiettivo: massimizzazione del ricavo

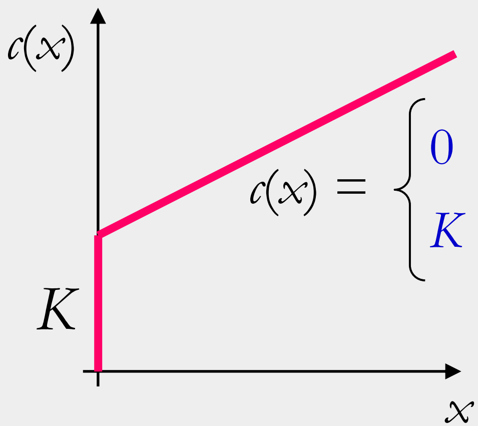
Vincoli: disponibilità di magazzino e non negatività delle quantità

Costi fissi (prob. di minimizzazione)

- **[variante]** I fertilizzanti A e B sono prodotti in due reparti distinti. I costi complessivi (manodopera, energia,…) sono rispettivamente di 5 e 4 €/Kg. I costi di avviamento dei reparti (il *setup*) sono rispettivamente di 150 e 195 €.

[Tecnica] Modellazione di *costi fissi*

- In molti casi reali, il costo ha una componente fissa e una variabile



$$c(x) = \begin{cases} 0 & \text{se } x = 0 \\ K + cx & \text{se } x > 0 \end{cases}$$

- La funzione costo $c\,(x)$ <u>non è</u> lineare
- $z = \min c \cdot x$ è una funzione obiettivo che minimizza solo la componente variabile del costo.

- Data una variabile non negativa x, in presenza di costi fissi è necessario distinguere la possibile scelta $x = 0$ dall'altrettanto possibile scelta $x > 0$.

- La funzione costo può essere scritta come $z = \min\,(cx + Ky)$ con y variabile binaria con il seguente significato

$$y = \begin{cases} 0 & \text{se } x = 0 \\ 1 & \text{se } x > 0 \end{cases}$$

- Ma come si traduce il *significato* che deve assumere la y in *vincoli matematici*?

  introducendo un cosiddetto ***vincolo bigM*** $\boxed{x \leq M\,y}$

  dove M è un limite superiore al valore che può assumere x

$$\boxed{x \leq M\, y}$$

Si osserva facilmente che

- $x > 0$ implica $y = 1$
- $y = 0$ implica $x = 0$
- $y = 1$ implica $x \leq M$

[Attenzione]:
Il vincolo **<u>non</u>** esprime l'equivalenza $\boxed{y = 1 \Leftrightarrow x > 0}$

(in particolare $x = 0$ **<u>non implica</u>** $y = 0$).

Tuttavia se $x = 0$ in una soluzione ottima, allora $y = 0$ (…perché?)

- I costi di produzione di A e B sono di 5 e 4 €/Kg e i costi di avviamento dei reparti sono di 150 e 195 €.

$$
\begin{aligned}
z = \max\ & (25 - 5)x_A + (28 - 4)x_B - 150y_A - 195y_B \\
& 0.4x_A + 0.24x_B \le 312 \\
& 0.4x_A + 0.45x_B \le 360 \\
& 0.1x_A + 0.31x_B \le 160 \\
& 0.1x_A \le 70 \\
& x_A \le 700y_A \\
& x_B \le 517y_B \\
& x_A, x_B \ge 0 \\
& y_A, y_B \in \{0,1\}
\end{aligned}
$$

- Definiamo le variabili binarie y_A, y_B
- Nella f.o. scontiamo i costi variabili dai profitti e introduciamo le componenti di costo fisso.
- Calcoliamo i *bigM* e introduciamo i *vincoli bigM*

Vincoli big-M

In generale, l'introduzione di *vincoli bigM* è la principale tecnica utilizzata per legare variabili reali a variabili binarie

[Esercizi]

- Verificare la correttezza dei valori di *bigM* utilizzati nella slide precedente
- Verificare che in un problema di massimo (con coefficienti tutti positivi) il vincolo *bigM* non funziona e proporre una soluzione.

Variabili semi-continue

- **[variante]** La procedura di avviamento dei reparti implica una produzione iniziale di almeno 30 Kg di fertilizzante. Quindi, se si decide di produrre uno dei fertilizzanti A oppure B, se ne produrrà una quantità **almeno pari** alla produzione iniziale.

[Tecnica] utilizzo di variabili *semi-continue*

- Una variabile non negativa x è detta semi-continua se assume valori nell'insieme $\{0\} \cup [L, M]$ con $L > 0$,

cioè $$\boxed{x = 0 \ \textbf{OR} \ L \leq x \leq M.}$$

- Una variabile semi-continua è utile per esempio per modellare un acquisto con quantità minima: posso non acquistare nulla ($x = 0$), ma se decido di acquistare qualcosa, devo emettere un ordine di almeno L unità ($x \geq L$).

- Di nuovo, è necessario distinguere la possibile scelta $x = 0$ dall'altrettanto possibile scelta $L \leq x \leq M$.

$$\boxed{Ly \leq x \leq My} \qquad \begin{cases} y = 1 \iff & L \leq x \leq M \\ y = 0 \iff & x = 0 \end{cases}$$

- **[Osservazione]** ponendo $L = \varepsilon$, con ε numero positivo *sufficientemente* piccolo, il vincolo esprime in modo *approssimato* l'**equivalenza** $y = 1 \iff x > 0$.

Se si decide di produrre un fertilizzante, se ne dovrà produrre una quantità almeno pari a 30 kg

$$
\begin{aligned}
z = \max\ & 25x_A + 28x_B \\
& 0.4x_A + 0.24x_B \leq 312 \\
& 0.4x_A + 0.45x_B \leq 360 \\
& 0.1x_A + 0.31x_B \leq 160 \\
& 0.1x_A \leq 70 \\
& 30y_A \leq x_A \leq 700y_A \\
& 30y_B \leq x_B \leq 517y_B \\
& x_A, x_B \geq 0 \\
& y_A, y_B \in \{0,1\}
\end{aligned}
$$

- Introduciamo le variabili binarie y_A e y_B
- $x_A = 0$ soddisfa i vincoli di disponibilità.
  Se invece $x_A > 0$ allora il vincolo $x_A \leq 700y_A$ impone $y_A = 1$ e di conseguenza il vincolo $30y_A \leq x_A$ impone $x_A \geq 30$
- Lo stesso ragionamento vale per x_B

Uso del big-M: schema riassuntivo

I vincoli big-M in generale modellano l'implicazione $A \Rightarrow B$

Siano $0 \le x \le M$, $y \in \{0,1\}$ e $\varepsilon > 0$ suff. piccolo

Se	Allora	vincolo
$y = 0$	$x = 0$	$x \le M\,y$
$x > 0$	$y = 1$	$x \le M\,y$
$y = 0$	$x \ge \varepsilon$	$x \ge \varepsilon\,(1 - y)$
$x = 0$	$y = 1$	$x \ge \varepsilon\,(1 - y)$

$A \Rightarrow B$
è equivalente a
not $B \Rightarrow$ not A

Uso del big-M: schema riassuntivo

I vincoli big-M in generale modellano l'implicazione A $\Rightarrow$ B

Siano $0 \le x \le M$, $y \in \{0,1\}$ e $\varepsilon > 0$ suff. piccolo

Se	Allora	vincolo
$y = 1$	$x = 0$	$x \le M(1-y)$
$x > 0$	$y = 0$	$x \le M(1-y)$
$y = 1$	$x \ge \varepsilon$	$x \ge \varepsilon y$
$x = 0$	$y = 0$	$x \ge \varepsilon y$

A $\Rightarrow$ B
è equivalente a
not B $\Rightarrow$ not A

Vincoli condizionali

- **[variante]** Una strategia commerciale definita ai piani alti prevede che se la produzione di B è uguale o supera 150 kg, allora la produzione di A non debba superare 450 Kg

(…che non equivale a dire che la produzioni totale deve essere ≤ 600)

[Tecnica] vincoli *condizionali*

$x_B \geq 150$?

No

$$z = \max\ 25x_A + 28x_B$$

$$\begin{aligned} 0.4x_A + 0.24x_B &\leq 312 \\ 0.4x_A + 0.45x_B &\leq 360 \\ 0.1x_A + 0.31x_B &\leq 160 \\ 0.1x_A &\leq 70 \\ x_A, x_B &\geq 0 \end{aligned}$$

Sì

$$z = \max\ 25x_A + 28x_B$$

$$\begin{aligned} 0.4x_A + 0.24x_B &\leq 312 \\ 0.4x_A + 0.45x_B &\leq 360 \\ 0.1x_A + 0.31x_B &\leq 160 \\ 0.1x_A &\leq 70 \\ \boxed{x_A} &\boxed{\leq 450} \\ x_A, x_B &\geq 0 \end{aligned}$$

vincolo condizionale

Come rendere un vincolo (in questo caso $x_A \leq 450$) condizionale al valore di una variabile (in questo caso x_B), o viceversa ?

In generale, una soluzione ammissibile deve soddisfare **tutti** i vincoli del modello. In questo caso siamo di fronte a un vincolo che **non deve** essere soddisfatto da tutte le soluzioni «ammissibili».

La scelta di soddisfare la condizione (o vincolo) $x_B \geq 150$ è essa stessa parte delle decisioni espresse dal modello ed è quindi descritta da una variabile binaria y, che assume il valore 1 quando il vincolo è soddisfatto

- Siano L una limitazione inferiore e M una limitazione superiore del vincolo $\mathbf{a}^{\mathrm{T}}\mathbf{x} \leq b$, cioè

$$L \leq \mathbf{a}^{\mathrm{T}}\mathbf{x} - b \leq M$$

[Esempio]

I limiti del vincolo $3x_1 + 4x_2 + 2x_3 \leq 7$ (con $x_1, x_2, x_3 \in \{0,1\}$) si ottengono ponendo tutte le variabili prima a 0 e poi a 1:

$$L = -7 \leq 3x_1 + 4x_2 + 2x_3 - 7 \leq 2 = M$$

- In generale, se $\mathbf{a} \geq \mathbf{0}$ e $\mathbf{0} \leq \mathbf{x} \leq \mathbf{m}$ allora

$$L = -b$$

$$M = \mathbf{a}^{\mathrm{T}}\mathbf{m} - b$$

Vincoli condizionali e bigM

Siano $L \leq \mathbf{a}^T\mathbf{x} - b \leq M, \quad y \in \{0,1\}$

se	allora	vincolo
$y = 0$	$\mathbf{a}^T\mathbf{x} \leq b$	$\mathbf{a}^T\mathbf{x} - b \leq My$
$\mathbf{a}^T\mathbf{x} > b$	$y = 1$	$\mathbf{a}^T\mathbf{x} - b \leq My$
$y = 0$	$\mathbf{a}^T\mathbf{x} \geq b$	$\mathbf{a}^T\mathbf{x} - b \geq Ly$
$\mathbf{a}^T\mathbf{x} < b$	$y = 1$	$\mathbf{a}^T\mathbf{x} - b \geq Ly$

Reminder: A $\Rightarrow$ B è equivalente a not B $\Rightarrow$ not A

Vincoli condizionali e bigM

Siano $L \leq \mathbf{a}^{\mathrm{T}}\mathbf{x} - b \leq M, \quad y \in \{0,1\}$

se	allora	vincolo
$y = 1$	$\mathbf{a}^{\mathrm{T}}\mathbf{x} \leq b$	$\mathbf{a}^{\mathrm{T}}\mathbf{x} - b \leq M(1 - y)$
$\mathbf{a}^{\mathrm{T}}\mathbf{x} > b$	$y = 0$	$\mathbf{a}^{\mathrm{T}}\mathbf{x} - b \leq M(1 - y)$
$y = 1$	$\mathbf{a}^{\mathrm{T}}\mathbf{x} \geq b$	$\mathbf{a}^{\mathrm{T}}\mathbf{x} - b \geq L(1 - y)$
$\mathbf{a}^{\mathrm{T}}\mathbf{x} < b$	$y = 0$	$\mathbf{a}^{\mathrm{T}}\mathbf{x} - b \geq L(1 - y)$

Reminder: A $\Rightarrow$ B è equivalente a not B $\Rightarrow$ not A

$$\mathbf{a}^T\mathbf{x} > b \Rightarrow y = 1$$

$$\boxed{\mathbf{a}^T\mathbf{x} - b \le My}$$

$\mathbf{a}^T\mathbf{x} > b$

se $\mathbf{a}^T\mathbf{x} - b > 0$ allora $y = 1$

$\mathbf{a}^T\mathbf{x} \le b$

se $\mathbf{a}^T\mathbf{x} - b \le 0$ allora y è libera

Analogamente

$\mathbf{a}^T\mathbf{x} \ge b \Rightarrow y = 1$ si traduce con $\boxed{\mathbf{a}^T\mathbf{x} - b + \varepsilon \le My}$

$$y = 1 \Rightarrow \mathbf{a}^{\mathrm{T}}\mathbf{x} \le b$$

$$\boxed{\mathbf{a}^{\mathrm{T}}\mathbf{x} - b \le M(1 - y)}$$

Infatti: $y = \begin{cases} 1 & \mathbf{a}^{\mathrm{T}}\mathbf{x} \le b \\ 0 & \mathbf{a}^{\mathrm{T}}\mathbf{x} \le b + M \end{cases}$

- per $y = 1$: il vincolo è “presente”
- per $y = 0$: il vincolo è ridondante perché è sempre soddisfatto

- Se la produzione di B è $\geq$ 150 kg allora la produzione di A non può eccedere 450 Kg. In formule:

$$x_B \geq 150 \Rightarrow x_A \leq 450$$

O analogamente $x_A > 450 \Rightarrow x_B < 150$

$$
\begin{array}{lll}
& z = \max 25x_A + 28x_B & \\
& 0.4x_A + 0.24x_B & \leq 312 \\
& 0.4x_A + 0.45x_B & \leq 360 \\
& 0.1x_A + 0.31x_B & \leq 160 \\
& 0.1x_A & \leq 70 \\
\text{(a)} & x_B - 150 + \varepsilon \leq 700\, y & \\
\text{(b)} & x_A - 450 \leq (1 - y)700 & \\
& x_A, x_B \geq 0 & \\
& y \in \{0,1\} &
\end{array}
$$

- se $x_B \geq 150$, il vincolo (a) impone $y = 1$ e questo valore di y «attiva» il vincolo (b)
- se invece $x_B < 150$, il vincolo (a) lascia libera la y che così potrà assumere qualsiasi valore nel vincolo (b), rendendolo di fatto ridondante.

- Se la produzione di B è ≥ 150 kg allora la produzione di A non può eccedere 450 Kg. In formule:

$$x_B \geq 150 \Rightarrow x_A \leq 450$$

O analogamente $x_A > 450 \Rightarrow x_B < 150$

$$
\begin{aligned}
& z = \max 25x_A + 28x_B \\
& \qquad 0.4x_A + 0.24x_B \leq 312 \\
& \qquad 0.4x_A + 0.45x_B \leq 360 \\
& \qquad 0.1x_A + 0.31x_B \leq 160 \\
& \qquad 0.1x_A \leq 70 \\
\text{(a)} \quad & \qquad x_B - 150 + \varepsilon \leq 700\,y \\
\text{(b)} \quad & \qquad x_A - 450 \leq (1-y)700 \\
& \qquad x_A, x_B \geq 0 \\
& \qquad y \in \{0,1\}
\end{aligned}
$$

- Analogamente, se $x_A > 450$, il vincolo (b) impone $y = 0$ e questo valore di y «attiva» il vincolo (a) e quindi $x_B < 150$

- Invece, se $x_A \leq 450$, il vincolo (b) lascia libera la y che così potrà assumere qualsiasi valore nel vincolo (a), rendendolo di fatto ridondante.

Vincoli condizionali: vincoli disgiuntivi

- Un caso particolare di vincoli condizionali è quello dei *vincoli disgiuntivi* (o *either-or*): Una coppia di condizioni $\mathbf{a}^{\mathrm{T}}\mathbf{x} \leq \mathbf{b}$ e $\mathbf{c}^{\mathrm{T}}\mathbf{x} \leq \mathbf{d}$ è disgiuntiva se **esattamente** una delle due **deve** essere soddisfatta (*or esclusivo*)

$$\begin{cases} \mathbf{a}^{\mathrm{T}}\mathbf{x} \leq \mathbf{b} + My \\ \mathbf{c}^{\mathrm{T}}\mathbf{x} \leq \mathbf{d} + M(1-y) \\ y \in \{0,1\} \end{cases}$$

vincoli disgiuntivi

[Esempio]

esprimere la condizione $x \neq y$, con x, y interi non negativi.

$x \neq y$ è equivalente alla coppia di condizioni:

$$\begin{cases} x \leq y - 1 \\ \underline{\text{oppure}} \\ x \geq y + 1 \end{cases}$$

nessuna soluzione soddisfa contemporaneamente entrambi i vincoli; i vincoli devono essere resi condizionali

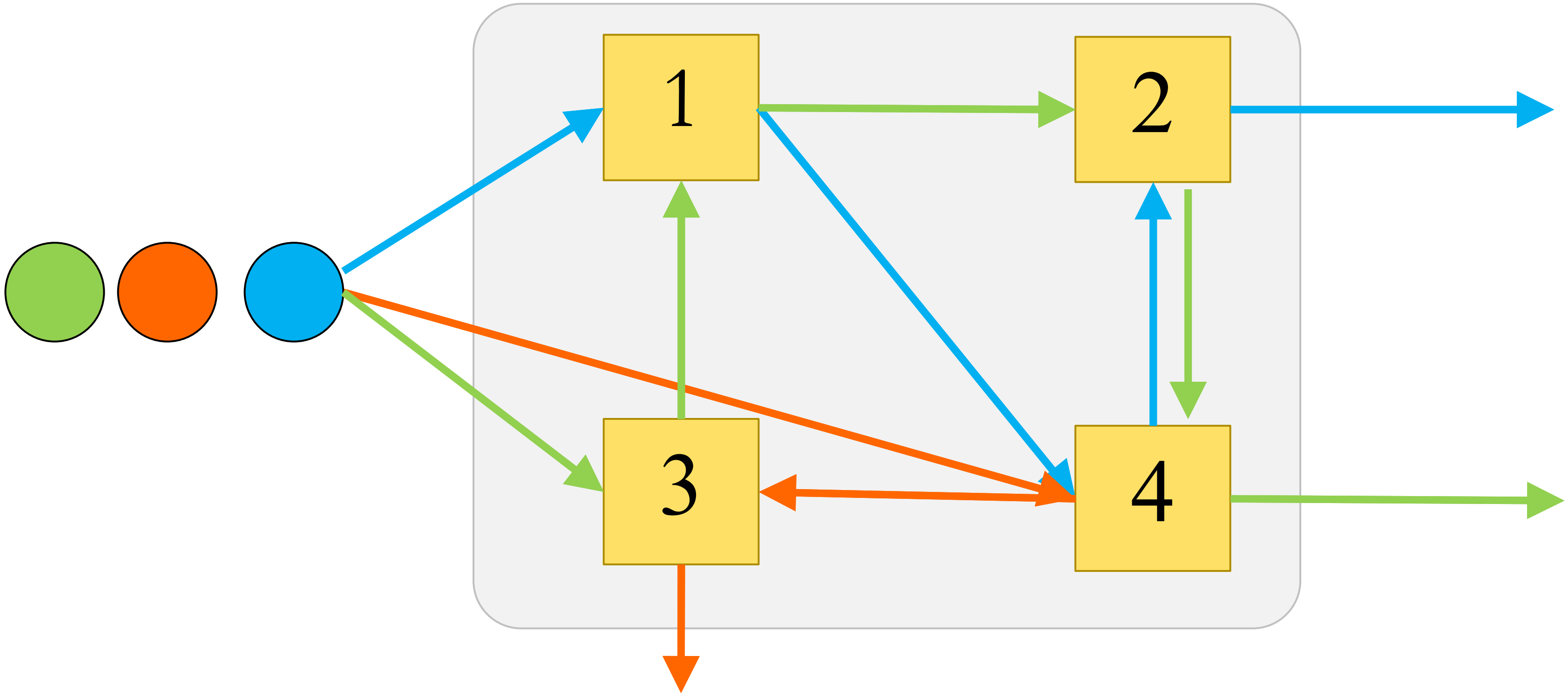
$$\begin{cases} x - y + 1 \leq M\lambda \\ y - x + 1 \leq M(1 - \lambda) \\ \lambda \in \{0,1\} \end{cases}$$

vincoli disgiuntivi

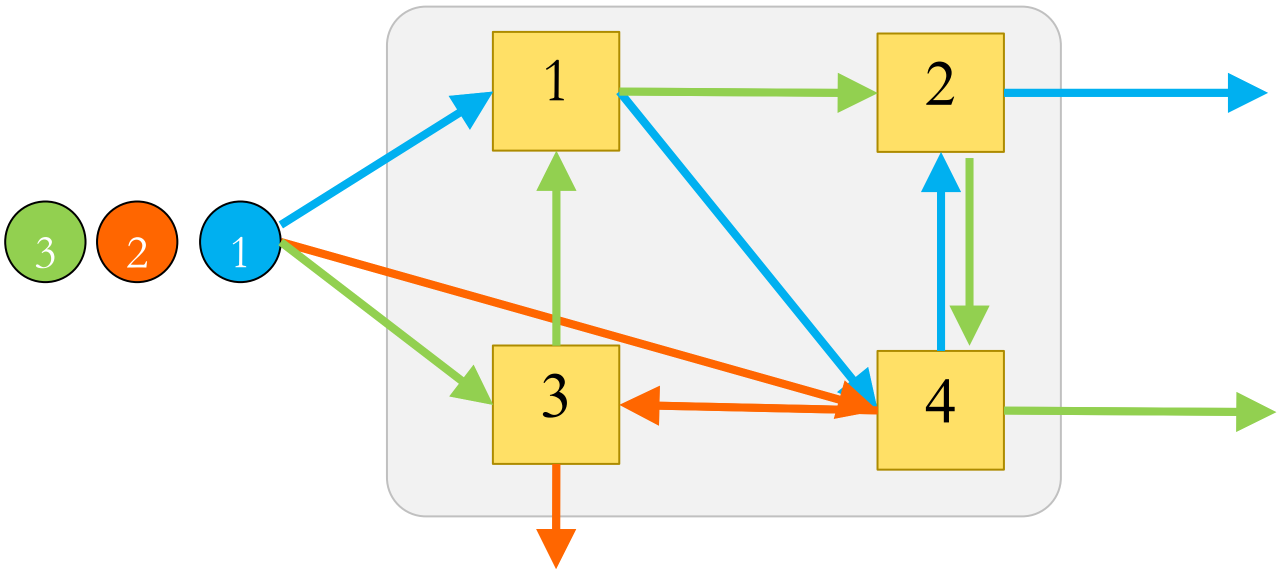
Un esempio:
job shop scheduling

Job shop scheduling

ogni job è formato da un certo numero di task, ognuno eseguito su una data macchina (ogni job ha un proprio ordine di visita – *routing* – delle macchine).



Job shop scheduling



- Il job j è formato dai task $(j,1),\ldots,(j,m_j)$
- Il task (j,k) deve essere processato sulla macchina $m(jk)$ e richiede un tempo di processamento p_{jk}

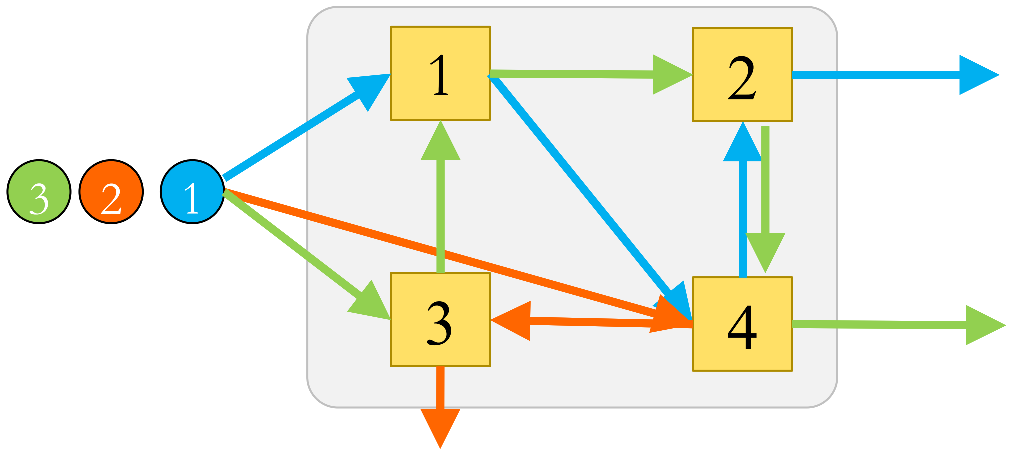
Job shop scheduling

- Tutte le macchine hanno capacità unitaria.
- Ogni macchina è dotata di un buffer di ingresso di capacità illimitata

Problema

- Determinare la sequenza di processamento dei task su ogni macchina che minimizzi il tempo totale di completamento di tutti i job (*makespan*)

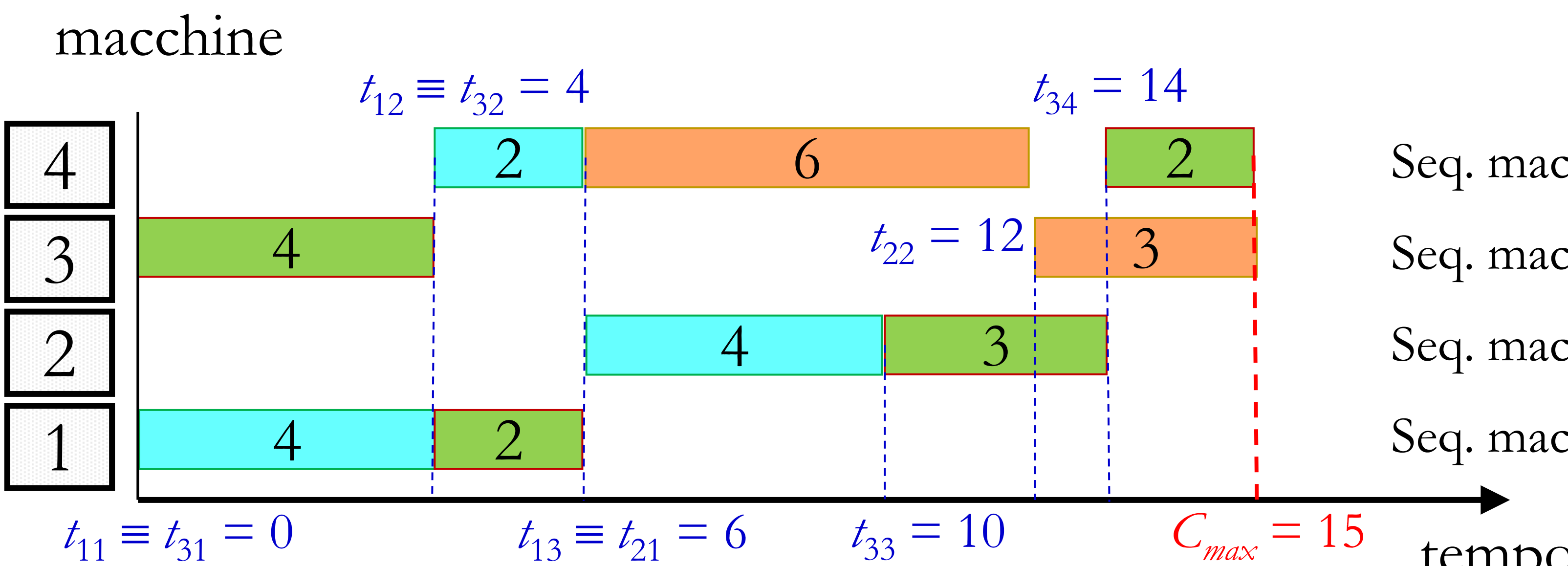
Job shop scheduling: esempio



Job/Task	1		2		3		4	
	$m(j1)$	p_{j1}	$m(j2)$	p_{j2}	$m(j3)$	p_{j3}	$m(j4)$	P_{j4}
1	1	4	4	2	2	4	-	-
2	4	6	3	3	-	-	-	-
3	3	4	1	2	2	3	4	2

Esempio: una soluzione

Job/Task	1		2		3		4	
	$m(j1)$	p_{j1}	$m(j2)$	p_{j2}	$m(j3)$	p_{j3}	$m(j4)$	P_{j4}
1	1	4	4	2	2	4	-	-
2	4	6	3	3	-	-	-	-
3	3	4	1	2	2	3	4	2



Job shop scheduling: formulazione matematica

Indichiamo con T l'insieme dei task di tutti i job

Variabili decisionali

$t_k \in \mathbb{R}^n$ = istante di inizio del task $k \in T$

Vincoli

- I task di ogni job devono essere eseguiti nell'ordine specificato

  $t_k + p_k \le t_h$

  $\forall$ coppia k, h di task consecutivi di uno stesso job

- I task processati su una macchina devono essere messi in sequenza

  $t_k + p_k \le t_h$ **oppure** $t_h + p_h \le t_k$

  $\forall$ coppia k, h di task processati su una macchina

$$
\begin{aligned}
&\min \; t_n \\
&t_k + p_k \le t_h && \forall (k,h) \in A_T \\
&\boxed{\begin{aligned} &t_k + p_k - t_h \le M(1 - y_{kh}) && \forall (k,h) \in A_M \\ &t_h + p_h - t_k \le M y_{kh} && \forall (k,h) \in A_M \end{aligned}} \\
&t_k \ge 0 && \forall k \in J \\
&y_{kh} \in \{0,1\} && \forall (k,h) \in A_M
\end{aligned}
$$

vincoli disgiuntivi:

$y_{kh} = 1$ se il task k precede il task h sulla macchina che li esegue entrambi

Esercizi

1. Discutere il caso $y = 0 \Rightarrow \mathbf{a}^{\mathrm{T}}\mathbf{x} \geq b$

2. Discutere il caso di vincoli condizionali di uguaglianza.

Predicati logici

- **[variante]** Dato che il magazzino è condiviso con un altro impianto, se la produzione di A e B porta entrambi i livelli di azoto e fosforo sotto la soglia di 35 Kg, si richiede una scorta di sicurezza di potassio di almeno 90 Kg.

[Tecnica] modellazione di *predicati logici*

Predicato logico:

[Azoto < 35] **AND** [fosforo < 35] $\Rightarrow$ [potassio $\geq$ 90]

$$
\begin{aligned}
z = \max\ & 25x_A + 28x_B \\
& 0.4x_A + 0.24x_B + s_a = 312 \\
& 0.4x_A + 0.45x_B + s_f = 360 \\
& 0.1x_A + 0.31x_B + s_p = 160 \\
& 0.1x_A \leq 70 \\
\text{(a)}\quad & 35 - s_a \leq 312\, y_a \\
\text{(b)}\quad & 35 - s_f \leq 360\, y_f \\
\text{(c)}\quad & y_a + y_f - 1 \leq y_p \\
\text{(d)}\quad & s_p \geq 90 y_p \\
& x_A, x_B, s_a, s_f, s_p \geq 0 \\
& y_a, y_f, y_p \in \{0,1\}
\end{aligned}
$$

- Introduciamo 3 variabili di slack s_a, s_f e s_p e 3 variabili di controllo y_a, y_f e y_p

- dal vincolo (a) se $s_a < 35$ allora $y_a = 1$
- dal vincolo (b) se $s_f < 35$ allora $y_f = 1$

- dal vincolo (c) $y_a = 1$ e $y_f = 1$ implica $y_p = 1$
- dal vincolo (d) se $y_p = 1$ allora $s_p \geq 90$

- Se $s_a \geq 35$ oppure $s_f \geq 35$ allora y_p è una variabile libera.

Calcolo proposizionale

Un **predicato logico** è un enunciato $p(X, Y, \dots, Z)$

i cui argomenti possono essere:

- una costante di verità (T oppure F),
- oppure un'asserzione, cioè un fatto che può assumere i valori di verità T oppure F

Gli argomenti sono legati tra loro da **connettivi logici**

$\neg$	negazione	(NOT)
$\wedge$	congiunzione	(AND)
$\vee$	disgiunzione	(OR)
$\otimes$	esclusione	(XOR)
$\Rightarrow$	implicazione	
$\Leftrightarrow$	equivalenza	

… che hanno i significati descritti nelle seguenti **tabelle di verità**

X	Y	$X \wedge Y$	$X \vee Y$	$X \otimes Y$	$X \Rightarrow Y$	$X \Leftrightarrow Y$
F	F	F	F	F	T	T
T	F	F	T	T	F	F
F	T	F	T	T	T	F
T	T	T	T	F	T	T

inoltre $\neg$F = T e $\neg$T = F

- I connettivi NOT, AND e OR costituiscono un insieme minimale della semantica del calcolo proposizionale; è facile verificare che lo XOR, l'implicazione e l'equivalenza si possono esprimere in funzione di essi

- Una volta **assegnati** i valori di verità alle asserzioni si ottiene una **proposizione logica** il cui valore di verità può essere facilmente calcolato utilizzando le tabelle della verità.

- L'operazione inversa, cioè determinare i valori di verità delle asserzioni che trasformano il predicato in una proposizione vera (o falsa), è invece un problema in generale molto difficile (SODDISFACIBILITÀ)

proprietà dell'algebra di Boole

- I predicati possono essere trasformati utilizzando le proprietà dell'algebra di Boole e alcune regole di riscrittura:

Commutativa:	$X \vee Y \equiv Y \vee X$
Associativa:	$X \wedge (Y \wedge Z) \equiv X \wedge Y \wedge Z$
Distributiva:	$X \wedge (Y \vee Z) \equiv (X \wedge Y) \vee (X \wedge Z)$
Assorbimento:	$X \vee (X \wedge Y) \equiv X$
Idempotenza:	$X \wedge X \equiv X$
Esistenza estremi:	$X \wedge \mathsf{F} \equiv \mathsf{F}$
Esistenza complemento:	$X \wedge \neg X \equiv \mathsf{F}$

…e per simmetria

Commutativa:	$X \wedge Y \equiv Y \wedge X$
Associativa:	$X \vee (Y \vee Z) \equiv X \vee Y \vee Z$
Distributiva:	$X \vee (Y \wedge Z) \equiv (X \vee Y) \wedge (X \vee Z)$
Assorbimento:	$X \wedge (X \vee Y) \equiv X$
Idempotenza:	$X \vee X \equiv X$
Esistenza estremi:	$X \vee \mathsf{T} \equiv \mathsf{T}$
Esistenza complemento:	$X \vee \neg X \equiv \mathsf{T}$

Regole di riscrittura:

Implicazione:	$X \Rightarrow Y \equiv \neg X \vee Y$
Equivalenza:	$X \Leftrightarrow Y \equiv X \Rightarrow Y \wedge Y \Rightarrow X$
De Morgan I:	$\neg (X \vee Y) \equiv \neg X \wedge \neg Y$
De Morgan II:	$\neg (X \wedge Y) \equiv \neg X \vee \neg Y$

- Associando una variabile booleana x ad ogni asserzione X si ottengono le **formule logiche booleane**

Forma Normale Congiuntiva (CNF)

Ogni formula logica può essere espressa come congiunzione di **clausole** nella cosiddetta *Forma Normale Congiuntiva*

- una *clausola* è una disgiunzione di letterali;
- un *letterale* è una variabile booleana x o la sua negazione $\neg x$

clausola letterali

CNF: $\boxed{(x_1 \lor x_2 \lor x_3)} \land (\boxed{\neg x_1} \lor \boxed{x_3} \lor \neg x_4)$

- un predicato logico non può essere direttamente trasformato in una espressione lineare sostituendo operatori logici con operatori algebrici e interpretando 1 = vero e 0 = falso.

- Sia x la variabile binaria associata all'asserzione X ($x = 1$ se l'asserzione è T e $x = 0$ se l'asserzione è F)

> Un vincolo (o un insieme di vincoli) modella correttamente un predicato logico se le soluzioni che soddisfano il vincolo sono <u>le uniche</u> che trasformano il predicato in una proposizione vera (ponendo 1 = T e 0 = F).

Connettivi logici: negazione

NOT: il predicato è vero se l'asserzione è falsa e viceversa

X	NOT X	x	$1 - x = 1$
T	F	1	violato
F	T	0	soddisfatto

Connettivi logici: congiunzione

AND: il predicato è vero se e solo se entrambe le asserzione sono vere

X_1	X_2	X_1 AND X_2	x_1	x_2	$x_1 + x_2 \geq 2$
T	T	T	1	1	soddisfatto
T	F	F	1	0	violato
F	T	F	0	1	violato
F	F	F	0	0	violato

Connettivi logici: disgiunzione

OR: il predicato è vero se e solo se almeno un'asserzione è vera

X_1	X_2	X_1 OR X_2	x_1	x_2	$x_1 + x_2 \geq 1$
T	T	T	1	1	soddisfatto
T	F	T	1	0	soddisfatto
F	T	T	0	1	soddisfatto
F	F	F	0	0	violato

Connettivi logici: esclusione

XOR: il predicato è vero se e solo se esattamente un'asserzione è vera

X_1	X_2	X_1 XOR X_2	x_1	x_2	$x_1 + x_2 = 1$
T	T	F	1	1	violato
T	F	T	1	0	soddisfatto
F	T	T	0	1	soddisfatto
F	F	F	0	0	violato

Connettivi logici: implicazione

$\Rightarrow$: il predicato è falso se la premessa è vera e la conseguenza è falsa: " se piove allora è nuvoloso"

piove è condizione *sufficiente* per essere nuvolo

essere nuvoloso è condizione *necessaria* per piovere

X_1	X_2	$X_1 \Rightarrow X_2$
T	T	T
T	F	F
F	T	T
F	F	T

$X_1 \Rightarrow X_2$ equivale a (NOT X_1 OR X_2)

che si traduce in $(1 - x_1) + x_2 \geq 1$

ossia $x_1 \leq x_2$

Infatti:

X_1	X_2	$X_1 \Rightarrow X_2$	x_1	x_2	$x_1 \leq x_2$
T	T	T	1	1	soddisfatto
T	F	F	1	0	violato
F	T	T	0	1	soddisfatto
F	F	T	0	0	soddisfatto

Connettivi logici: equivalenza

$\Leftrightarrow$: il predicato è vero se e solo se le due asserzioni sono entrambe vere o entrambe false

X_1	X_2	$X_1 \Leftrightarrow X_2$	x_1	x_2	$x_1 = x_2$
T	T	T	1	1	soddisfatto
T	F	F	1	0	violato
F	T	F	0	1	violato
F	F	T	0	0	soddisfatto

Alcune formule

$$\text{Sia } x_i = \begin{cases} 1 & \text{se } X_i \\ 0 & \text{se } \neg X_i \end{cases}$$

Predicati	Vincoli
X_1 OR … OR X_n	$x_1 + \ldots + x_n \geq 1$
X_1 AND … AND X_n	$x_1 + \ldots + x_n \geq n$
Almeno k	$x_1 + \ldots + x_n \geq k$
Esattamente k	$x_1 + \ldots + x_n = k$
Al più k	$x_1 + \ldots + x_n \leq k$

Regola generale di trasformazione

1. Porre la formula logica in Forma Normale Congiuntiva utilizzando le proprietà dell'algebra di Boole e le regole di riscrittura;
2. sostituire ogni letterale y con la variabile binaria x e ogni letterale $\neg y$ con l'espressione $1 - x$ ($X \Leftrightarrow \neg Y$ equivale a $x + y = 1$)
3. Ogni clausola (disgiunzione di letterali $y_1, \ldots, y_k$) si traduce nel vincolo $x_1 + \ldots + x_k \geq 1$
4. La congiunzione delle clausole si ottiene combinando in un unico sistema le disequazioni associate alle clausole ottenute nel punto 3.

Predicati logici: esercizi

- Modellare con variabili binarie e vincoli lineari le seguenti proposizioni

1. not (A_1 or A_2)
2. not (A_1 and A_2)
3. $A_1 \Rightarrow$ not A_2
4. $A_1 \Rightarrow$ (A_2 and A_3)
5. $A_1 \Rightarrow$ (A_2 or A_3)
6. (A_2 and A_3) $\Rightarrow A_1$
7. (A_2 and A_3 and A_4) $\Rightarrow A_1$
8. (A_2 or A_3) $\Rightarrow A_1$
9. A_1 and (A_2 or A_3)
10. A_1 or A_2 and A_3
11. $A_1 \Leftrightarrow A_2$ or A_3 or A_4
12. $A_1 \Leftrightarrow A_2$ and A_3 and A_4
13. $A_1 \Leftrightarrow A_2$ xor A_3

Esempio: una pianificazione della produzione

- Si valuta l'opportunità di mettere in produzione 3 nuovi articoli. L'azienda dispone di 2 stabilimenti ognuno in grado di produrre tutti gli articoli con i tempi riportati in tabella.

  Si vuole massimizzare il profitto. Però per evitare un'eccessiva diversificazione e per ridurre i costi logistici, si impone che

  a. solo uno degli stabilimenti può essere utilizzato per la nuova produzione;

  b. al massimo 2 articoli su 3 possono essere messi in produzione.

		Tempi unitari di lavorazione (ore)	
	Profitto per item(€)	Impianto 1	Impianto 2
Articolo 1	5	2	2
Articolo 2	4	3	1
Articolo 3	6	1	4
Tempo disponibilità (ore)		30	35

Esempio (cont.)

	Profitto per item(€)	Tempi unitari di lavorazione (ore) Impianto 1	Impianto 2
Articolo 1	5	2	2
Articolo 2	4	3	1
Articolo 3	6	1	4
Tempo disponibilità (ore)		30	35

Partiamo da un modello del problema che non tenga conto delle restrizioni a. e b.

$x_{ij} \in \mathrm{R}$ = quantità di prodotto i che si decide di produrre nello stabilimento j

$$
\begin{aligned}
z = \max\ & 5(x_{11} + x_{12}) + 4(x_{21} + x_{22}) + 6(x_{31} + x_{32}) \\
& 2x_{11} + 3x_{21} + 1x_{31} \leq 30 \\
& 2x_{12} + 1x_{22} + 4x_{32} \leq 35 \\
& x_{ij} \geq 0 \qquad i = 1,2,3;\ j = 1,2
\end{aligned}
$$

… notate qualcosa di particolare?

Esempio (cont.)

a) solo uno degli stabilimenti può essere utilizzato per la nuova produzione

Non è più necessario specificare per ogni prodotto lo stabilimento di provenienza; basta una variabile binaria che esprime la scelta dello stabilimento

- $x_i \in \mathbb{R}$ = quantità che si decide di produrre di prodotto i
- $y = 1$ se si sceglie lo stabilimento 2 e 0 altrimenti

Ora però i vincoli di capacità sono disgiuntivi: occorre soddisfare solo quello relativo allo stabilimento scelto

$$
\begin{aligned}
z = \max\ & 5x_1 + 4x_2 + 6x_3 \\
& 2x_1 + 3x_2 + 1x_3 \le 30 + M\,y \\
& 2x_1 + 1x_2 + 4x_3 \le 35 + M\,(1-y) \\
& x_i \ge 0 \qquad i = 1,2,3 \\
& y \in \{0,1\}
\end{aligned}
$$

Esempio (cont.)

b) al massimo 2 prodotti su 3 possono essere messi in produzione.

E' necessario *contare* i prodotti scelti e limitarli a 2:

- $\mu_i = 1$ se il prodotto i viene scelto (cioè se $x_i > 0$)

Utilizziamo la tecnica del *bigM*

$$
\begin{aligned}
z = \max\ & 5x_1 + 4x_2 + 6x_3 \\
& 2x_1 + 3x_2 + 1x_3 \leq 30 + M_4\, y \\
& 2x_1 + 1x_2 + 4x_3 \leq 35 + M_5\,(1 - y) \\
& x_1 \leq M_1\, \mu_1 \\
& x_2 \leq M_2\, \mu_2 \\
& x_3 \leq M_3\, \mu_3 \\
& \mu_1 + \mu_2 + \mu_3 \leq 2 \\
& x_i \geq 0 \quad i = 1,2,3 \\
& y \in \{0,1\} \\
& \mu_i \in \{0,1\}\ i = 1,2,3
\end{aligned}
$$

[Esercizio] indicare una possibile scelta dei coefficienti M_i

Esercizio

Si consideri il problema di pianificazione dell'esempio precedente con le seguenti modifiche:

a) l'azienda dispone di uno stabilimento aggiuntivo che ha una capacità produttiva di 33 ore e in cui una unità di prodotto 1 (2 e 3) richiede 1 ora (rispettivamente 2 e 3 ore) di lavorazione.

b) al più 2 dei 3 stabilimenti possono essere utilizzati per la produzione.

Definire un modello di Programmazione Lineare Intera

Tecniche di modellazione matematica

- vincoli *hard* e *soft*
- Linearizzazione di valori assoluti
- Linearizzazione di f. convesse lineari a tratti
- Linearizzazione di funzioni esponenziali

Vincoli *hard* e *soft*

- **[variante]** Le disponibilità di materia prima possono essere incrementate acquistandone (a un prezzo leggermente maggiorato) una quantità aggiuntiva.
  I costi sono rispettivamente di 10 €/Kg per l'Azoto, 12 per il Fosforo, 14 per il Potassio e 11 per il Magnesio.

[Tecnica] Trasformazione da vincolo *hard* a vincolo *soft*

- Un vincolo è detto

  - *hard* se non può essere assolutamente violato
  - *soft* se la violazione è accettabile ma non desiderabile

- La trasformazione da vincolo *hard* a vincolo *soft* è utile per esempio nei casi di inammissibilità o per f.o. multi-criterio

Vincolo di $\leq$

$$\begin{aligned} \max \quad & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ & \mathbf{a}^{\mathrm{T}}\mathbf{x} \leq b \end{aligned}$$

$\Rightarrow$

$$\begin{aligned} \max \quad & \mathbf{c}^{\mathrm{T}}\mathbf{x} - Ku \\ & \mathbf{a}^{\mathrm{T}}\mathbf{x} \leq b + u \\ & u \geq 0 \end{aligned}$$

con $K > 0$ fattore di penalità

Vincolo di =

$$\max \quad \mathbf{c}^{\mathrm{T}}\mathbf{x}$$
$$\mathbf{a}^{\mathrm{T}}\mathbf{x} = b$$

$$\max \quad \mathbf{c}^{\mathrm{T}}\mathbf{x} - Ku - Hv$$
$$\mathbf{a}^{\mathrm{T}}\mathbf{x} - u + v = b$$
$$u, v \geq 0$$

con $K, H > 0$ fattori di penalità

Introduciamo una variabile non negativa u per ogni vincolo allo scopo di ammettere una soluzione che richieda una disponibilità di magazzino superiore a quella attuale.

$$z = \max 25x_A + 28x_B - 10u_1 - 12u_3 - 14u_3 - 11u_4$$

$$\begin{aligned}
0.4x_A + 0.24x_B &\leq 312 + u_1 \\
0.4x_A + 0.45x_B &\leq 360 + u_2 \\
0.1x_A + 0.31x_B &\leq 160 + u_3 \\
0.1x_A &\leq 70 + u_4
\end{aligned}$$

$$x_A, x_B, u_1, u_2, u_3, u_4 \geq 0$$

I valori delle variabili u_1, u_2, u_3, u_4 indicano le violazioni dei rispettivi vincoli (cioè la materia prima che dovrà essere ulteriormente acquistata). Tali violazioni vengono pagate in funzione obiettivo con delle penalizzazioni.

Valori assoluti

- **[variante]** Si decide di vendere A e B allo stesso prezzo di 25 €/kg e si suppone che le quantità di materia prima (312 kg di Azoto, 360 Kg di Fosforo, 160 Kg di Potassio e 70 Kg di Magnesio) non siano disponibilità di magazzino ma valori nominali forniti dalla pianificazione. Qual è il mix produttivo che *approssima meglio* i valori nominali di magazzino?

[Nota] lo scostamento dai valori nominali può essere sia positivo sia negativo

[Tecnica] modellazione di *valori assoluti*

$$z = \min |s_a| + |s_f| + |s_p| + |s_m|$$

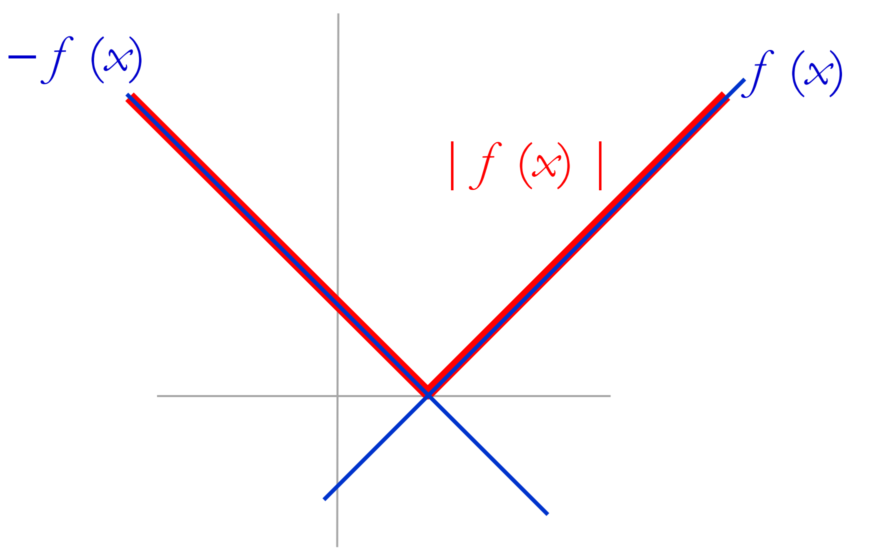
$$
\begin{aligned}
0.4x_A + 0.24x_B - 312 &= s_a \\
0.4x_A + 0.45x_B - 360 &= s_f \\
0.1x_A + 0.31x_B - 160 &= s_p \\
0.1x_A \qquad\qquad - 70 &= s_m \\
x_A , x_B &\geq 0
\end{aligned}
$$

- Dato che A e B hanno lo stesso prezzo non è importante determinare il mix-produttivo.
- Introduciamo le variabili reali s_a , s_f , s_p e s_m che indicano lo scostamento del consumo di materia prima dai valori nominali.
- La soluzione cercata è quella che minimizza la somma dei valori assoluti degli scostamenti

ma il valore assoluto non è una funzione lineare!

Valori assoluti

$f(x) = x - 2$



Il modello lineare ha una variabile e due vincoli in più per ogni valore assoluto

$$|f(x)| = \max\{f(x), -f(x)\}$$

- Introduciamo la variabile

$$y \geq f(x)$$
$$y \geq -f(x)$$

- Casi:

$\boxed{f(x) \geq 0}$ $\quad y \geq f(x) \geq 0 \geq -f(x)$

$\boxed{f(x) \leq 0}$ $\quad y \geq -f(x) \geq 0 \geq f(x)$

$$y \geq |f(x)| \geq 0$$

Valori assoluti: esempio

$$\min 2\,x_1 + 3|x_2 - 10|$$
$$|x_1 + 2| + |x_2| \leq 5$$

- Siano
  - $z_1 = \max\{x_2 - 10, -x_2 + 10\}$
  - $z_2 = \max\{x_1 + 2, -x_1 - 2\}$
  - $z_3 = \max\{x_2, -x_2\}$

$$\min 2\,x_1 + 3z_1$$
$$z_2 + z_3 \leq 5$$
$$z_1 \geq x_2 - 10$$
$$z_1 \geq -x_2 + 10$$
$$z_2 \geq x_1 + 2$$
$$z_2 \geq -x_1 - 2$$
$$z_3 \geq x_2$$
$$z_3 \geq -x_2$$

Valori assoluti: un'altra linearizzazione

$$\min \sum_{i=1}^{n} c_i \, |x_i|$$

$$\mathbf{Ax} \geq \mathbf{b}$$

$$\min \sum_{i=1}^{n} c_i \, (x_i^+ + x_i^-)$$

$$\mathbf{Ax}^+ - \mathbf{Ax}^- \geq \mathbf{b}$$

$$\mathbf{x}^+, \mathbf{x}^- \geq \mathbf{0}$$

Ogni variabile reale x_i può essere scritta come:

$$x_i = x_i^+ - x_i^- \text{ con } x_i^+, x_i^- \geq 0$$

Sostituiamo

$$|\, x_i \,| \text{ con } x_i^+ + x_i^- \text{ e}$$

$$x_i \text{ con } x_i^+ - x_i^-$$

$x_i = x_i^+ - x_i^-$ e $x_i^+, x_i^- \geq 0$ non implicano $|\, x_i \,| = x_i^+ + x_i^-$. Tuttavia, in ogni soluzione **ottima** $x_i^+ = 0$ oppure $x_i^- = 0$. **Perché?**

Il modello lineare ha una variabile in più per ogni valore assoluto

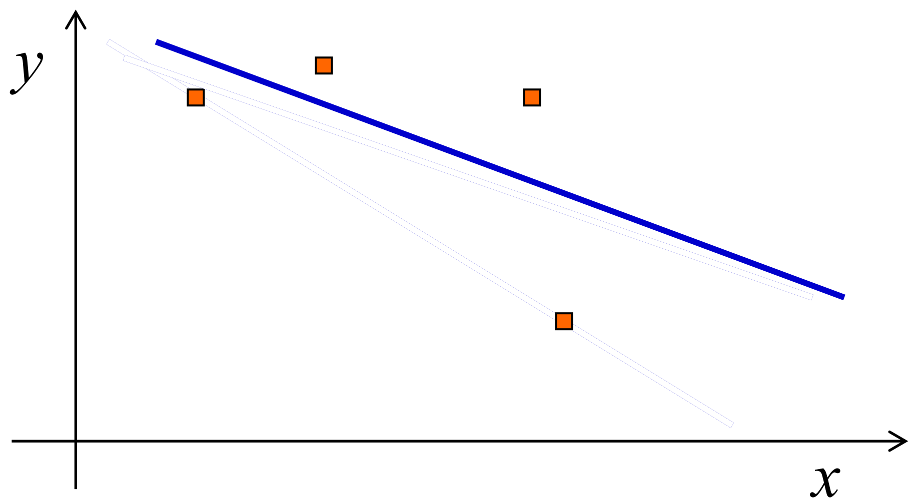
Analisi di regressione

- **Analisi di regressione:** tecnica usata per analizzare serie di dati (rappresentati come punti dello spazio R^n) allo scopo di effettuare previsioni, inferenza statistica, per testare ipotesi o per modellare delle relazioni di dipendenza

  - E' basata sulla stima **dei parametri** che definiscono una *curva di regressione*, curva che ha distanza minima dalla nuvola dei punti e che quindi meglio descrive la relazione esistente tra una variabile dipendente e una o più variabili indipendenti.

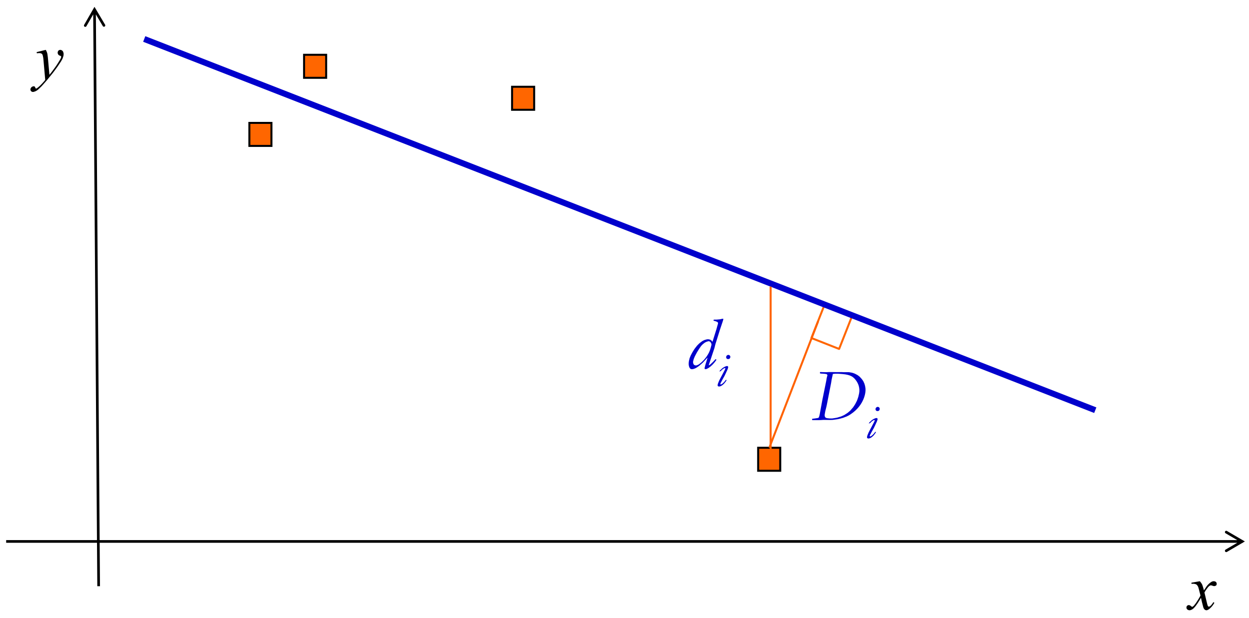
Quando si presume una relazione di tipo lineare si parla di regressione lineare. Se la relazione è tra una sola variabile dipendente y e una sola variabile indipendente x la curva di regressione è una retta definita dalla coppia di parametri (a, b)

$$\boxed{y = ax + b}$$



La retta di regressione è la retta a distanza minima dalla nuvola di punti.

La distanza di una retta da un insieme di punti è la somma delle distanze di ciascun punto dalla retta.



distanza d_i di un punto (x_i, y_i) da una retta $f(x) = ax + b$

differenza tra ordinate: $d_i = |y_i - f(x_i)| = |y_i - (ax_i + b)|$

euclidea: $D_i = \dfrac{|y_i - (ax_i + b)|}{\sqrt{1 + a^2}}$

$$d_i = |y_i - (ax_i + b)|$$

- x_i e y_i sono noti, quindi la distanza d_i dipende dai parametri a e b, gli stessi che descrivono completamente la retta di regressione incognita.

- La retta di regressione è quella descritta dai parametri a e b tali da minimizzare la somma $d_1 +,\ldots, + d_n$ delle distanze dei punti
- Il problema da risolvere è quindi

$$\min d_1 +,\ldots, + d_n =$$

$$\min |y_1 - (ax_1 + b)| +,\ldots, + |y_n - (ax_n + b)|$$

nelle variabili $a, b \in \mathbb{R}$

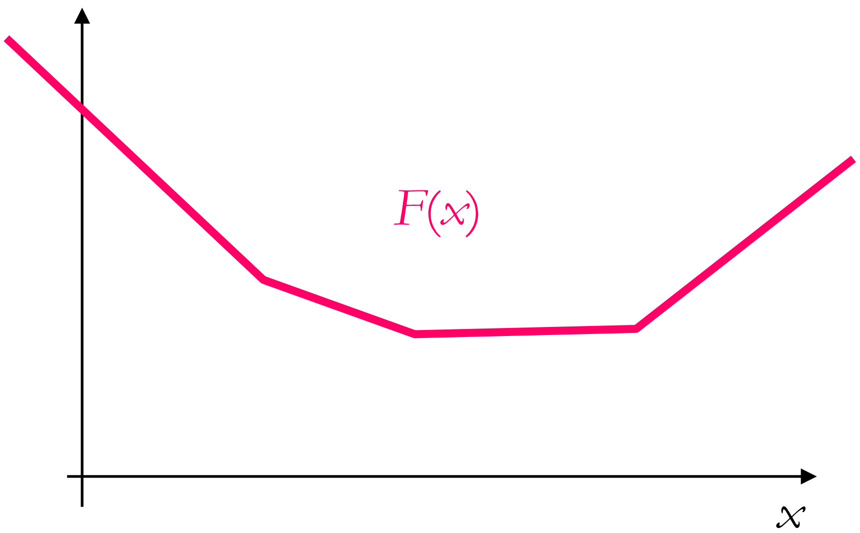
$$\min \ |y_1 - (ax_1 + b)| + ,\ldots, + |y_n - (ax_n + b)|$$

è equivalente a un modello di programmazione **lineare** con $n + 2$ variabili e $2n$ vincoli

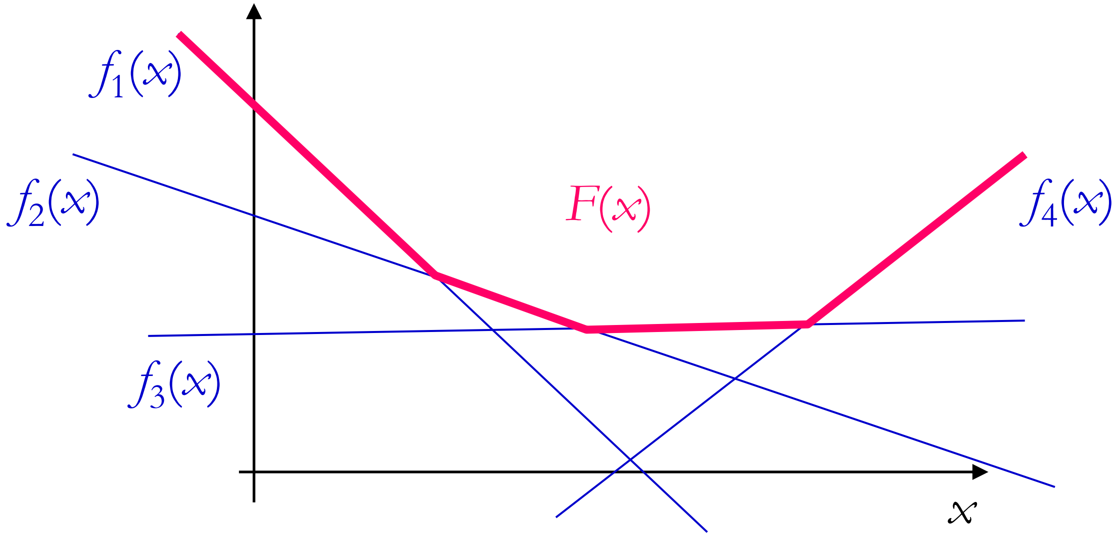
$$\begin{aligned}
\min \ & d_1 +,\ldots, + d_n \\
& d_1 \geq y_1 - (ax_1 + b) \\
& d_1 \geq -y_1 + (ax_1 + b) \\
& \qquad \ldots \\
& d_n \geq y_n - (ax_n + b) \\
& d_n \geq -y_n + (ax_n + b)
\end{aligned}$$

Caso generale: min di f.o. convessa lineare a tratti

- Si vuole minimizzare una funzione $F(x)$ convessa lineare a tratti

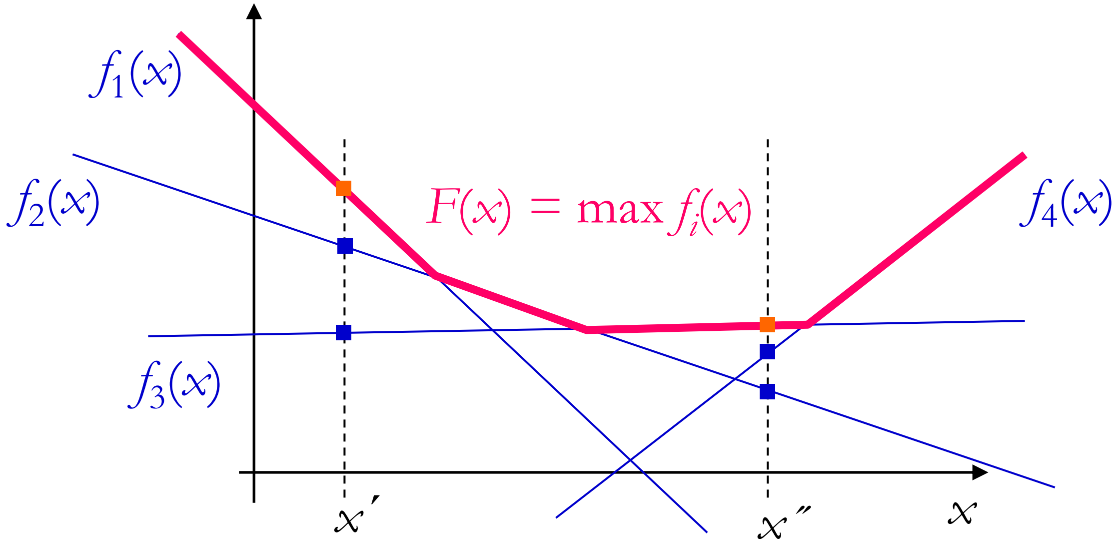


- Si vuole minimizzare una funzione $F(x)$ convessa lineare a tratti



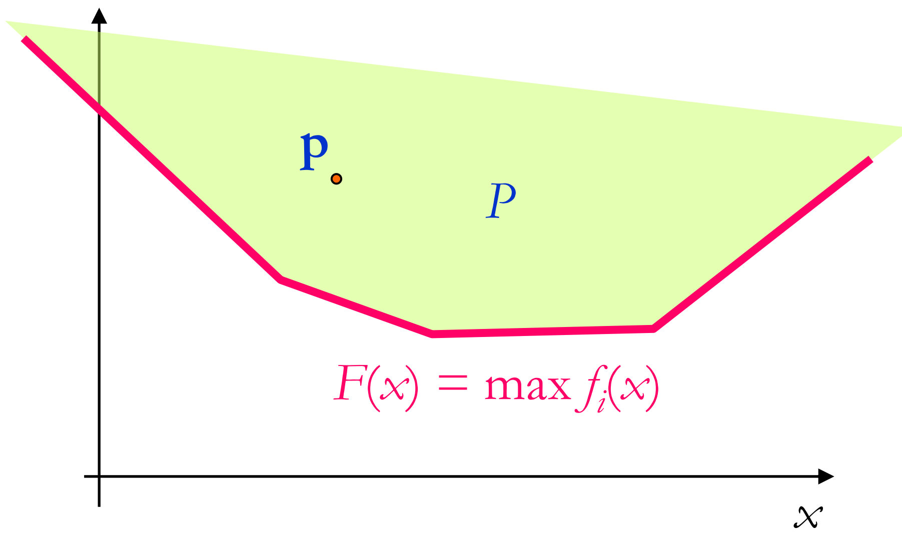
- Ogni tratto i della funzione lineare a tratti $F(x)$ è descritto da una funzione lineare $f_i(x) = c_i x + d_i$

- Si vuole minimizzare una funzione $F(x)$ convessa lineare a tratti



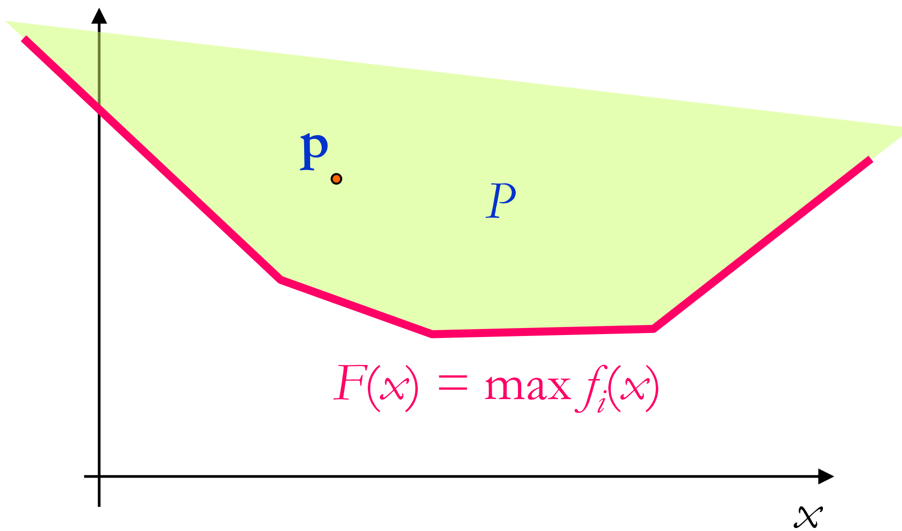
- Per ogni $x \in \mathbb{R}$, $F(x)$ è il massimo valore assunto dalle $f_i(x)$ in x

- Si vuole minimizzare una funzione $F(x)$ convessa lineare a tratti



- Per ogni $\mathbf{p} = (x, y) \in P$ si ha $y \geq f_i(x) \quad \forall i$. Ma P è il poliedro definito dalle funzioni lineari $f_i(x)$, quindi $\min F(x)$ equivale a minimizzare y su P

- Si vuole minimizzare una funzione $F(x)$ convessa lineare a tratti

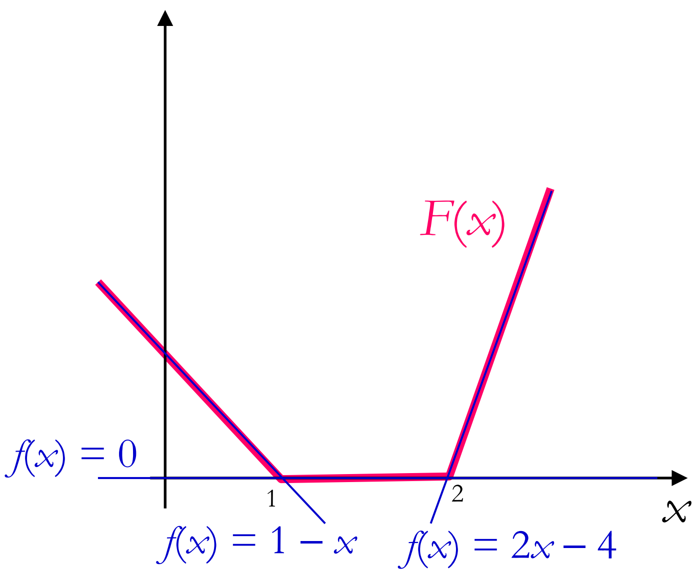


$$\min_{x,y} y$$
$$y \geq c_i x + d_i \qquad \forall i$$

Esempio

$$\min\left(\mathbf{c}^{\mathrm{T}}\mathbf{x} + F(\mathbf{d}^{\mathrm{T}}\mathbf{x})\right)$$
$$\mathbf{A}\mathbf{x} \geq \mathbf{b}$$

con $F(x) = \max\{1 - x, 0, 2x - 4\}$



- ponendo $y = F(x)$ si ottiene il modello lineare

$$\min\left(\mathbf{c}^{\mathrm{T}}\mathbf{x} + y\right)$$
$$y \geq 1 - \mathbf{d}^{\mathrm{T}}\mathbf{x}$$
$$y \geq 0$$
$$y \geq 2\mathbf{d}^{\mathrm{T}}\mathbf{x} - 4$$
$$\mathbf{A}\mathbf{x} \geq \mathbf{b}$$

Esercizi

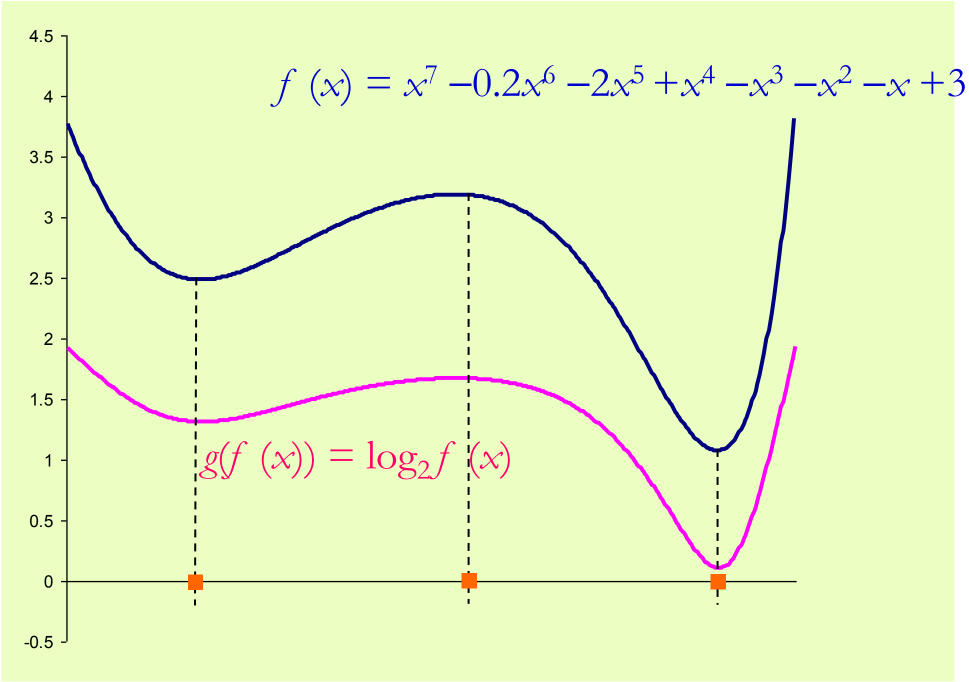
- Discutere i problemi che si incontrano quando si vuole minimizzare una funzione concava lineare a tratti e proporre una soluzione.

- Scrivere un modello di PLI per la massimizzazione di una funzione concava lineare a tratti.

[Tecnica] modellazione di una f.o. *esponenziale*

f.o. esponenziale

- Se g è una funzione *monotona* (per esempio il logaritmo) la composizione $g \circ f$ **ha gli stessi** punti stazionari di f



- La funzione g può essere utilizzata per linearizzare la f.o.

Esempio

$$f(\mathbf{x}) = (7^{x_1+3x_2})$$

$$g \circ f = \log(f(\mathbf{x})) = \log(7^{x_1+3x_2}) = (x_1 + 3x_2) \cdot \log 7$$

$$\min f \equiv \min(g \circ f) = \min \log(f(\mathbf{x}))$$

$$= \log 7 \cdot \min(x_1 + 3x_2)$$

Esempio

$$f(\mathbf{x}) = \prod_i x_i \qquad\qquad \text{with} \quad x_i > 0$$

$$g \circ f = \log(f(\mathbf{x})) = \log \prod_i x_i = \sum_i \log x_i \equiv \sum_i x_i$$

$\log(x_i)$ è una composizione tra logaritmo e funzione identità quindi possiamo sostituire $\log(x_i)$ con x_i

Bibliografia

1. F.S. Hillier, G.J. Lieberman,
   Ricerca Operativa,
   Mc Graw-Hill, IX ed., 2010

2. A. Agnetis, C. Arbib, M. Lucertini, S. Nicoloso,
   Il Processo Decisionale,
   Nuova Italia Scientifica, 1992

3. Lezioni del Prof. Claudio Arbib (www.oil.di.univaq.it)