

# 1. Modellazione

## 1.1 Processo di Modellazione

### 1. Identificare le variabili decisionali

- Preferire variabili a singolo indice
- Usare due indici solo se necessario (es. matrici)
- Attenzione a "indipendentemente da" → suggerisce singolo indice

### 2. Definire la funzione obiettivo

### 3. Scrivere i vincoli base

- Partire dai vincoli più semplici
- Aggiungere vincoli specifici gradualmente

### 4. Aggiungere vincoli speciali

- Vincoli logici → variabili binarie  $y_i$
- Big-M →  $x_i \leq M y_i$
- Costi fissi →  $(\text{costo\_var} \cdot x_i + \text{costo\_fisso} \cdot y_i)$
- Capacità →  $\{x_i \leq \text{cap\_max}, x_i \geq \text{cap\_min}\}$
- Budget →  $\sum x_i \leq \text{budget}$

### 5. Definire i domini

- Di default considerare variabili intere
- Usare reali solo per frazioni/proporzioni

## 1.2 Pattern Comuni

### Vincoli Quantitativi

#### 1. "Almeno k"

```
# Variabili binarie
 $\sum y_i \geq k$ 
# Variabili normali
 $x_i \geq k$ 
```

#### 2. "Al massimo k"

```
# Variabili binarie
 $\sum y_i \leq k$ 
```

```
# Variabili normali  
 $x_i \leq k$ 
```

### 3. "Esattamente k"

```
# Variabili binarie  
 $\sum y_i = k$   
# Variabili normali  
 $x_i = k$ 
```

## Modelli Standard

### 1. Copertura minima

```
min  $\sum (c_i * x_i)$   
s.t.  $\sum a_{ij} * x_i \geq b_j \quad \forall j$ 
```

### 2. Mix ottimo

```
max  $\sum (p_i * x_i)$   
s.t.  $\sum a_{ij} * x_i \leq q_j \quad \forall j$ 
```

### 3. Trasporto

```
min  $\sum_i \sum_j (c_{ij} * x_{ij})$   
s.t.  $\sum_j x_{ij} \leq O_i \quad \forall i$   
 $\sum_i x_{ij} \geq D_j \quad \forall j$ 
```

## 2. Metodo del Simplexso

### 2.1 Forma Standard

#### 1. Funzione obiettivo

- Se max  $\rightarrow$  min(-f.o.)
- Tutti i termini a sinistra

#### 2. Vincoli

- Tutti di uguaglianza (aggiungere slack)
- Termini noti  $\geq 0$
- Se  $\leq \rightarrow +\text{slack}$

- Se  $\geq \rightarrow -\text{slack}$

### 3. Variabili

- Tutte  $\geq 0$
- Se  $x_i$  libera  $\rightarrow x_i = x_i' - x_i'', x_i', x_i'' \geq 0$
- Se  $x_i \leq 0 \rightarrow x_i = -x_i', x_i' \geq 0$

## 2.2 Tableau

	$x_1$	$x_2$	$\dots$	$x_n$	$b$
-z	[costi ridotti]				val
	[coefficienti]				RHS

## 2.3 Iterazione

### 1. Verifica forma canonica

- Base = colonne matrice identità
- Costi ridotti base = 0

### 2. Verifica ammissibilità

- $\bar{b}_i \geq 0 \forall i$

### 3. Verifica ottimalità

- Costi ridotti  $\geq 0$

### 4. Verifica illimitatezza

- Colonna con costo ridotto  $< 0$
- Tutti coefficienti  $\leq 0$

### 5. Selezione variabile entrante

- Min indice tra costi ridotti  $< 0$  (Bland)

### 6. Selezione variabile uscente

- Min rapporto  $\bar{b}_i / \bar{a}_{ik} > 0$
- Se più minimi  $\rightarrow$  min indice (Bland)

## 2.4 Soluzione

### 1. Valore f.o.

- $z_{\max} = -z_{\min}$
- Migliora di  $\theta \cdot c_r$  ( $\theta = \text{min rapporto}$ )

### 2. Vincoli

- Laschi se slack  $> 0$
- Saturi se slack = 0

## 2.5 Casi Particolari

### 1. Base degenerare

- Due o più righe stesso rapporto minimo
- Variabile base = 0
- F.o. non migliora

### 2. Non pivot su elemento

- Non corrisponde a rapporto minimo
- Porterebbe a soluzione non ammissibile

## 3. Algoritmi Cammini Minimi

### 3.1 Scelta Algoritmo

- **Bellman-Ford** se:
  - Costi negativi
  - Limite massimo archi
  - Complessità:  $O(|N| \cdot |A|)$
- **Dijkstra** se:
  - Solo costi positivi
  - No limite archi
  - Complessità:  $O(|N|^2)$  o  $O(|A| + |N| \log |N|)$  con heap

### 3.2 Bellman-Ford

#### 1. Tabella iterazioni

Iter	Nodi	Aggiornati
0	[etichette]	[nodi]
1	[etichette]	[nodi]
...	...	...

#### 2. Fino a:

- $h = |N|$  iterazioni
- Due righe consecutive identiche

#### 3. Aggiornamento:

- Per nodi aggiornati all'iterazione  $h-1$
- Per archi uscenti da tali nodi
- Se  $\pi_j > \pi_i + c_{ij} \rightarrow \pi_j = \pi_i + c_{ij}$

#### 4. Output:

- Albero: predecessori ultima riga
- Grafo: tutti cammini costo  $\leq$  ottimo
- Ciclo negativo: se aggiornamenti a  $|N|$

## 3.3 Dijkstra

### 1. Tabella iterazioni

Iter	$\bar{S}$	$\hat{v}$	Nodi
0	[nodi]	–	[etich]
1	[nodi]	–	[etich]
...	...		

### 2. Simbologia

- • : nodo verificato
- • : non migliora
- × : già chiuso
- vuoto: non controllato

### 3. Fino a:

- $\bar{S} = \emptyset$

### 4. Output:

- Albero: predecessori ultima riga
- Grafo: tutti cammini costo  $\leq$  ottimo

## 4. Branch and Bound

### 4.1 Struttura

- **Minimo:** [LB; SA]
- **Massimo:** [SA; UB]

### 4.2 Proprietà

#### 1. Monotonia

- Minimo: LB cresce padre-figlio
- Massimo: UB decresce padre-figlio

#### 2. Chiusura nodi

- Minimo:  $LB \geq SA$
- Massimo:  $UB \leq SA$

#### 3. Intervallo ottimo

- Minimo: [min LB aperti; min SA totale]
- Massimo: [max SA totale; max UB aperti]

### 4.3 Strategie

#### 1. Best Bound First

- Minimo: nodo con min LB
- Massimo: nodo con max UB

## 2. Depth First

- Nodo più profondo nell'albero

## 4.4 Ottimalità

- Tutti nodi chiusi, oppure
- Intervallo contiene solo un intero

## 5. AMPL

### 5.1 File .mod

```
set I;          # Insiemi
param p{I};     # Parametri
var x{I} >=0;    # Variabili

maximize z:      # F.O.
    sum{i in I} p[i]*x[i];

s.t. vincolo{j in J}: # Vincoli
    sum{i in I} a[i,j]*x[i] <= b[j];
```

### 5.2 File .dat

```
set I := i1 i2 i3;

param p :=
i1  10
i2  20
i3  30;

param a :   j1  j2  j3 := # (tr) per trasposta
i1      2   3   4
i2      5   6   7
i3      8   9  10;
```

### 5.3 File .run

```
reset;
model file.mod;
```

```
data file.dat;  
option solver cplex;  
solve;  
display z, x;
```