

# Laboratorio: Ottimizzazione su reti

Luigi De Giovanni

Dipartimento di Matematica, Università di Padova

# Cammino minimo: modello

Variabili decisionali:  $x_{ij} = \begin{cases} 1, & \text{l'arco } (i,j) \text{ è sul cammino minimo;} \\ 0, & \text{altrimenti.} \end{cases}$

$$\min \quad \sum_{(i,j) \in A} c_{ij} x_{ij}$$

s.t.

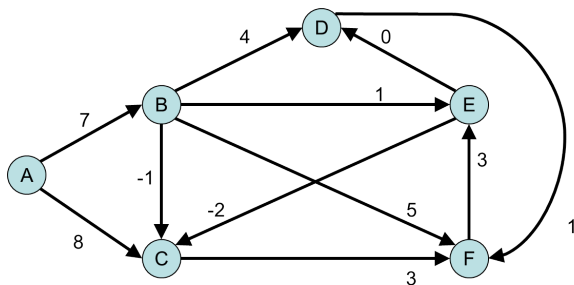
$$\sum_{(i,v) \in A} x_{iv} - \sum_{(v,j) \in A} x_{vj} = \begin{cases} -1, & v = s; \\ +1, & v = d; \\ 0, & v \in N \setminus \{s, d\}. \end{cases}$$

$$x_{ij} \in \{0,1\} \equiv \mathbb{R}_+ \quad \forall (i,j) \in A$$

Caso **particolare MCF**: un nodo domanda  $s$  e un nodo offerta  $d$

## Cammino minimo: esempio

Si determini con AMPL il cammino minimo da A a F nel seguente grafo



**Suggerimento:** si considerino variabili  $x_{ij}$  per ogni coppia di nodi, assegnando un valore di  $c_{ij}$  molto alto per gli archi che non (utilizzare la parola chiave `default`)

# Soluzioni e discussione

## Cammini minimi

- `sp0.mod`, `sp0.dat` (implementazione base)
- `sp.mod`, `sp.dat` (implementazione con archi espliciti)

## Max hop

- `spH.mod`, `sp.dat`

La soluzione del rilassamento continuo è intera!

Per calcolare il rilassamento continuo: `option relax_integrality 1`

# Cammini minimi vincolati

Ad esempio, sono dati i consumi per tratta  $w_{ij}$  e la disponibilità di carburante nel serbatoio  $W$

- spW.mod, spW.dat

La soluzione del rilassamento continuo potrebbe non essere intera:  
**si perde la totale unimodularità**

# Albero dei cammini minimi

Calcolare con AMPL l'albero dei cammini minimi da A verso gli altri nodi

Si potrebbe usare il modello precedente cambiando il nodo destinazione nel file .dat, o, in alternativa (più efficiente)...

**Suggerimento:** usare il modello del flusso di costo minimo, con il nodo offerta A e tutti gli altri nodi domanda, con domanda unitaria (come indicato nella slide che segue).

- `sptree.mod`, `sptree.dat`

*Esercizio.* Considerare un vincolo sul peso totale degli archi dell'albero (notare che si perde la totale unimodularità della matrice)

- `sptreeW.mod`, `sptreeW.dat`

# Modello PL per flusso di costo minimo

- parametri  $b_i$ : bilanciamento al nodo  $i$ 
  - $b_A = 5 = |N| - 1$
  - $b_{i \neq A} = 1$
- variabili: quantità  $x_{ij}$  da far fluire sull'arco  $(i, j) \in A$

$$\min \quad \sum_{(i,j) \in A} c_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_{(i,v) \in A} x_{iv} - \sum_{(v,j) \in A} x_{vj} = b_v \quad \forall v \in N$$

$$x_{ij} \in \mathbb{R}_+[\{0, 1\}, \mathbb{Z}_+]$$

## Esercizio

Si vogliono trasferire 50 unità da A a F: calcolare in che modo le unità devono essere distribuite sulla rete in modo da minimizzare il costo, considerando che ciascun arco ha una capacità limitata (ed esistono archi con capacità inferiore a 50 sul cammino minimo, altrimenti basterebbe in questo caso il cammino minimo da A a F). Si consideri il trasferimento di un numero intero di unità.

**Suggerimento:** *si tratta da un problema di flusso di costo minimo, con un nodo offerta e un nodo domanda.*

- `mcf.mod`, `mcf.dat`

**Domanda:** cambierebbe la soluzione se le unità fossero frazionabili?

## Esercizio (Max-Flow)

Si determini, in base alle capacità degli archi, il massimo numero  $maxf$  di unità che possono essere trasferite da A a F.

**Suggerimento:** *il problema può essere modellato come flusso di costo minimo, introducendo una variabile relativa a  $maxf$  da trasferire da A (con bilanciamento  $-maxf$ ) a F (con bilanciamento  $+maxf$ ).*

- `maxf.mod`, `maxf.dat`

**Osservazione:** la soluzione rimane intera se le unità sono frazionabili e risolviamo con il semplice (si può dimostrare che la matrice dei vincoli resta totalmente unimodulare)

**Osservazione:** l'interezza del rilassamento continuo si perde con l'introduzione di un vincolo di budget (si può dimostrare che la matrice dei vincoli NON resta totalmente unimodulare)