LABORATORIO DI RICERCA OPERATIVA 2022-2023

Laboratorio OPL - LEZIONE 3

Grafi orientati e pesati in OPL

Due definizioni equivalenti

```
tuple nodo {
        int IndiceNodo;
        int divergenza;
};
{nodo} InsiemeNodi = ...;
```

```
tuple arco {
        int NodoOut;
        int NodoIn;
        int costoA;
        int capA;
};
{arco} InsiemeArchi = ...;
```

Implementazione 2

```
int NNodi = ...;
range Nodi = 1..NNodi;
int divergenza[Nodi] = ...;
```

```
tuple arco {
        int NodoOut;
        int NodoIn;
};
{arco} InsiemeArchi = ...;
int costoA[InsiemeArchi] = ...;
int capA[InsiemeArchi] = ...;
```

Flusso di costo minimo

```
int NNodi = ...;
range Nodi = 1..NNodi;

tuple arco{
   int NodoOut;
   int NodoIn;
 }

{arco} Archi = ...;

int Divergenza[Nodi] = ...;
int Costo[Archi] = ...;
dvar float+ F[Archi];
```

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

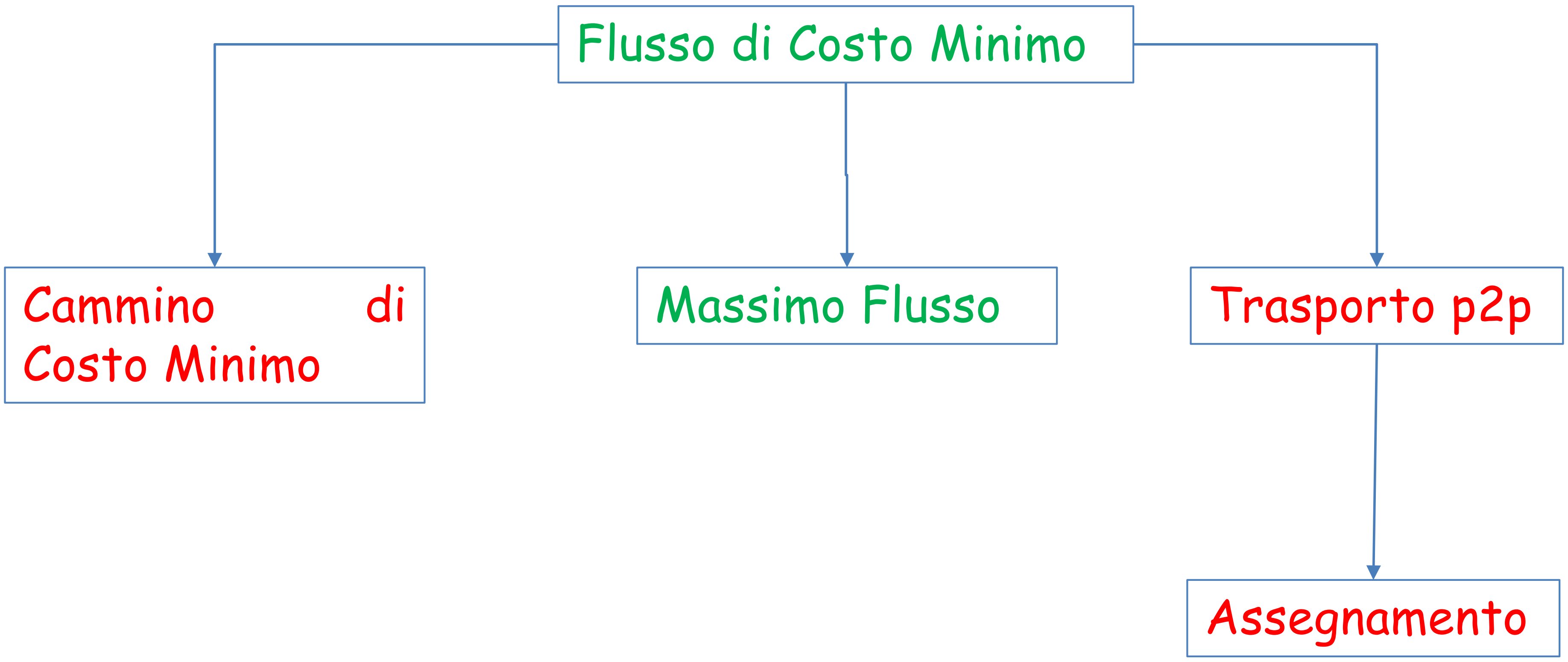
```
minimize sum(<i,j> in Archi) Costo[<i,j>]*F[<i,j>];
subject to{
```

$$\sum_{j|(i,j)\in A} f_{ij} - \sum_{j|(j,i)\in A} f_{ji} = b_i \qquad \forall i \in V$$

```
forall(i in Nodi)

sum(j in Nodi: <i,j> in Archi) F[<i,j>] -
sum(j in Nodi: <j,i> in Archi) F[<j,i>]==Divergenza[i];
}
```

$$f_{ij} \geq 0 \qquad (i,j) \in A$$

Problemi di Ottimizzazione su Rete



Cammino Elementare di Costo Minimo dal nodo S al nodo T (SPP)

Consideriamo un problema di flusso di costo minimo sul grafo $G = \langle V, E \rangle$, non capacitato, con le seguenti caratteristiche:

1. In G è presente un solo nodo sorgente S con divergenza $d_S = +1$; inoltre, S ha soli archi uscenti (sorgente in senso forte);
2. In G è presente un solo nodo pozzo T con divergenza $d_T = -1$; il nodo T , chiamato anche terminale, ha soli archi entranti (pozzo in senso forte);
3. Tutti gli altri nodi $\neq S, T$ hanno divergenza nulla;
4. I costi unitari di flusso c_{ij} sono tutti non negativi $\forall (i,j) \in E$

$$\min \sum_{(i,j)\in E} c_{ij}\, f_{ij}$$

$$\sum_{j|(S,j)\in E} f_{Sj} = 1$$

Equazione di continuità sul nodo S

$$\sum_{j|(i,j)\in E} f_{ij} - \sum_{j|(j,i)\in E} f_{ji} = 0 \quad \forall i \neq S, T$$

Equazioni di continuità sui nodi $\neq S, T$

$$-\sum_{i|(i,T)\in E} f_{iT} = -1$$

Equazione di continuità sul nodo T

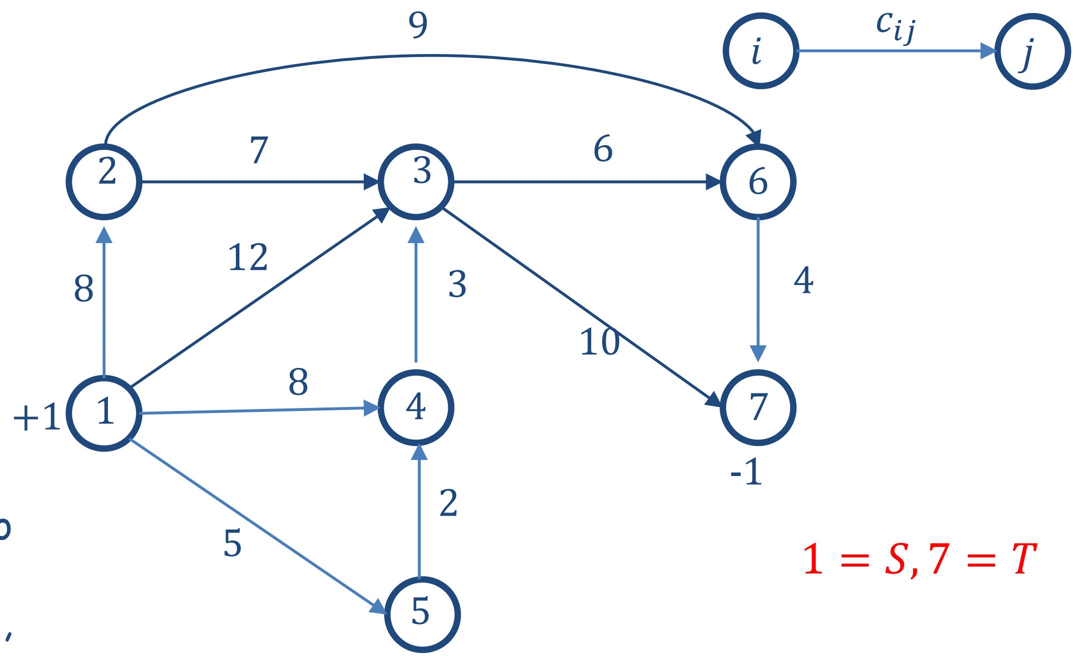
$$f_{ij} \geq 0 \quad \forall (i,j) \in E$$

Cammino Elementare di Costo Minimo dal nodo S al nodo T (SPP)

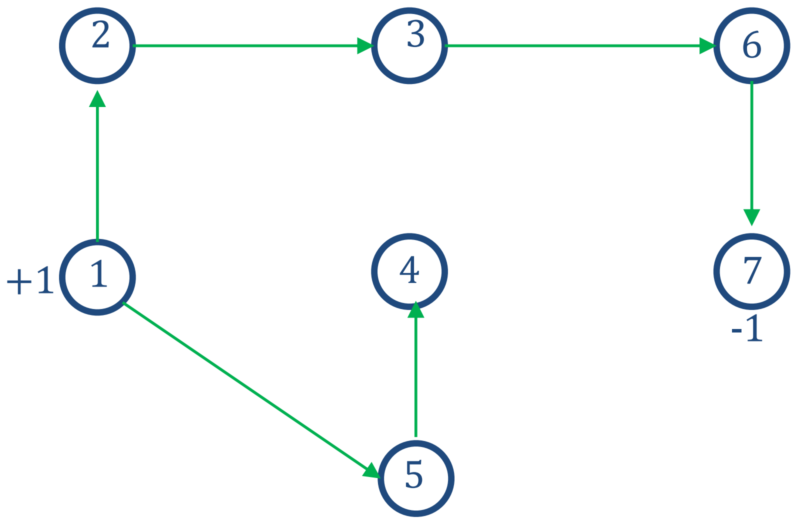
$$\min \sum_{(i,j)\in E} c_{ij} f_{ij}$$

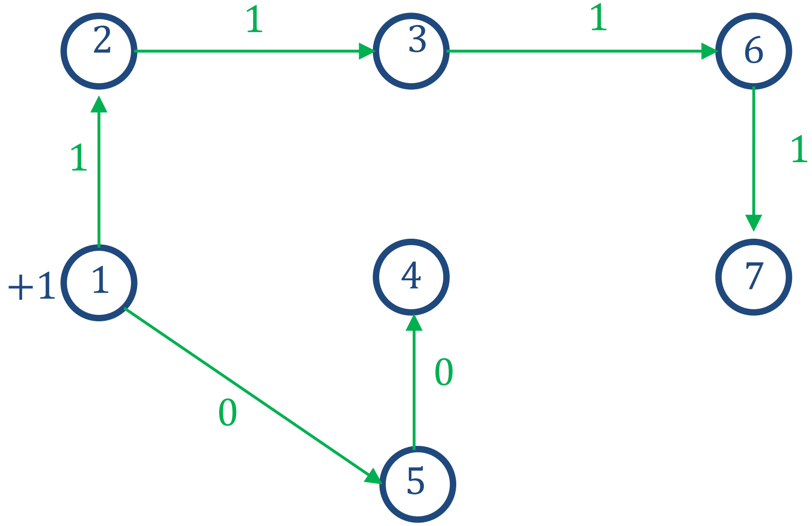
$$\sum_{j|(i,j)\in E} f_{ij} - \sum_{j|(j,i)\in E} f_{ji} = \begin{cases} 1 & i = S \\ 0 & i \neq S,T \\ -1 & i = T \end{cases}$$

$$f_{ij} \geq 0 \quad \forall (i,j) \in E$$



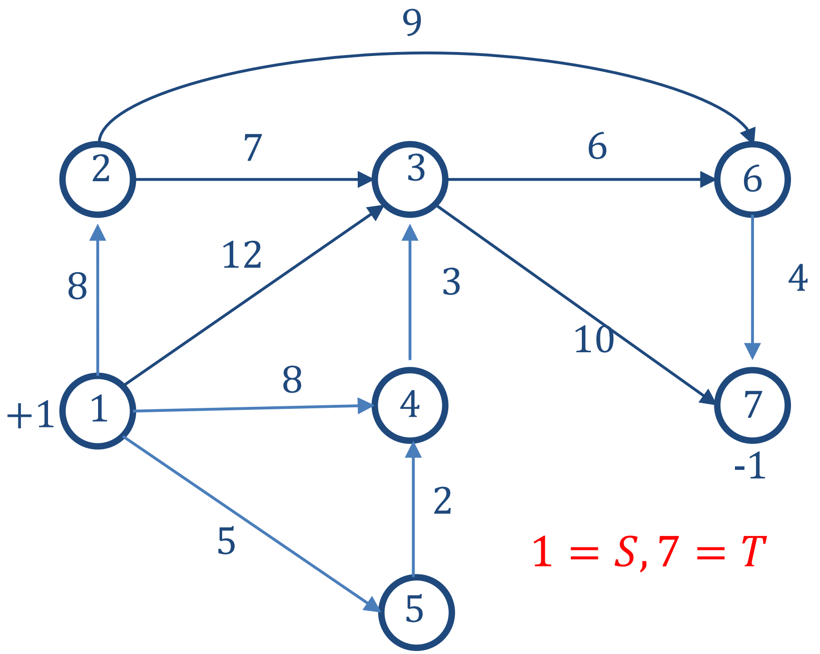
1) Le soluzioni ammissibili di base sono rappresentate da alberi
2) Le variabili di base assumono valore intero, poiché le divergenze sono intere





3) Le soluzioni ammissibili di base sono punti a coordinate 0-1 e sono tutte degeneri

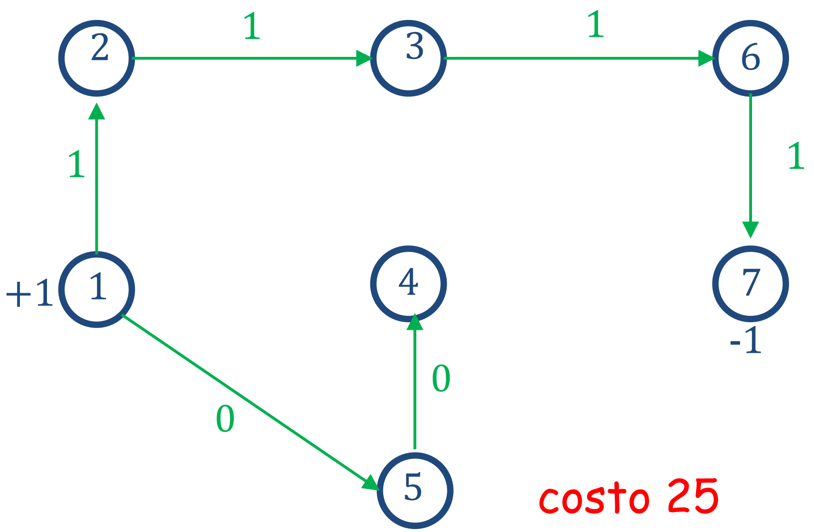
Cammino Elementare di Costo Minimo dal nodo S al nodo T (SPP)



1) In ogni soluzione ammissibile di base l'unica unità di flusso disponibile presso il nodo S raggiungerà il nodo T attraversando archi, senza mai dividersi.

2) In ogni soluzione ammissibile di base, gli archi con flusso non nullo (e pari ad 1), individuano un cammino elementare orientato da S a T.

3) La soluzione ottima rappresenta il cammino elementare di costo minimo da S a T.



costo 25

Cammino Elementare di Costo Minimo dal nodo S al nodo T (SPP)

Applicazioni

1) Google Maps e qualsiasi sistema di navigazione

2) Telecomunicazioni: instradamento dei dati per minimizzare il ritardo di comunicazione

3) Optimal wiring of VLSI chip

4) ....

Metodi di Risoluzione

E' un problema di PL in forma standard e può essere risolto con l'Algoritmo del Simplesso su rete.

E' un caso particolare di MCFP e, come MaxF, può essere risolto con algoritmi «ad hoc»

1) Algoritmo di Dijkstra (se i costi sugli archi sono tutti non negativi)

2) Algoritmo di Floyd-Warshall (in presenza di costi negativi sugli archi)

Implementazione OPL di SPP

```
int NNodi = ...;
range Nodi = 1..NNodi;
tuple arco{
   int NodoOut;
   int NodoIn;
 }
{arco} Archi = ...;
```

~~int Divergenza[Nodi] = ...;~~ //non serve: i nodi hanno divergenza 1,-1,0.

```
int Costo[Archi] = ...;
int S = ...;                   //Modello parametrico rispetto S e T
int T = ...;
dvar float+ F[Archi];
```

$$\min \sum_{(i,j)\in E} c_{ij} f_{ij}$$

```
minimize sum(<i,j> in Archi) Costo[<i,j>]*F[<i,j>];
  subject to{
```

$$\sum_{j|(S,j)\in E} f_{Sj} = 1$$

```
sum(<S,j> in Archi) F[<S,j>]==1;
```

$$\sum_{j|(i,j)\in E} f_{ij} - \sum_{j|(j,i)\in E} f_{ji} = 0 \quad \forall i \neq S, T$$

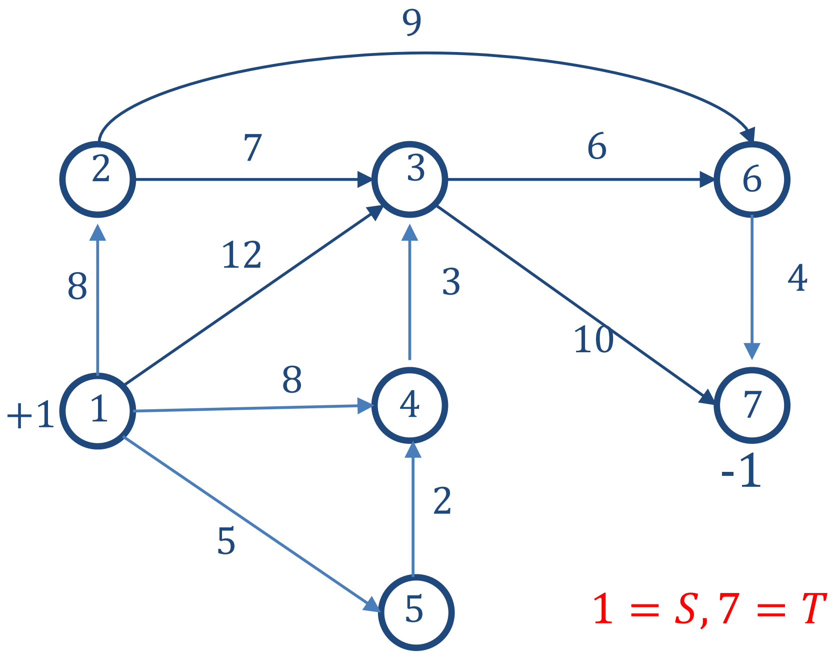
```
forall(i in Nodi: i != S && i!=T)

   sum(j in Nodi: <i,j> in Archi) F[<i,j>] -
   sum(j in Nodi: <j,i> in Archi) F[<j,i>]==0;
```

$$-\sum_{i|(i,T)\in E} f_{iT} = -1$$

```
sum(<i,T> in Archi) -1*F[<i,T>]==-1 ;
```

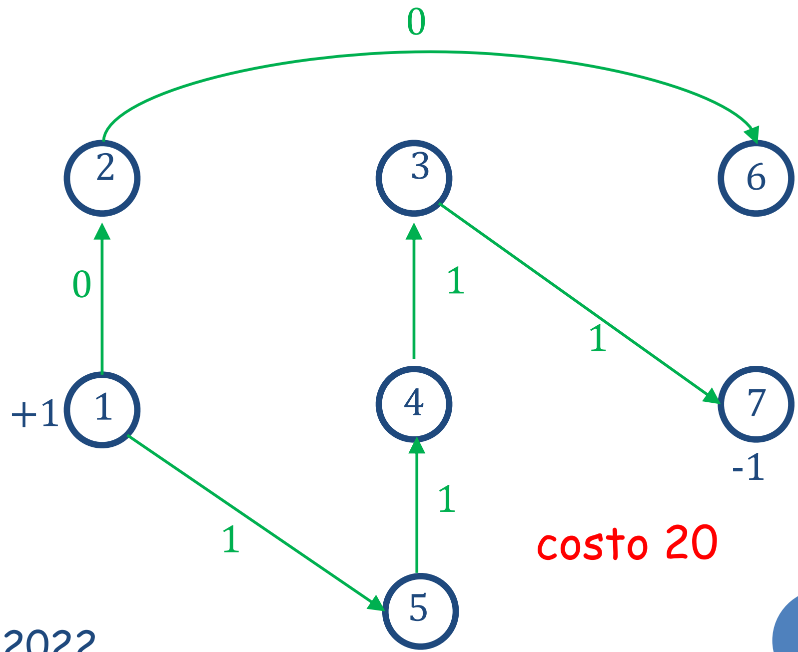
$$f_{ij} \geq 0 \quad \forall (i,j) \in E$$

```
}
```



$1 = S, 7 = T$

Soluzione ottima



costo 20

Implementazione OPL di Grafi Completi

Elencare tutti gli elementi dell'insieme degli archi è l'unico modo possibile per modellare un grafo "non completo".

Nel caso di grafi con insieme completo di Archi (con esclusione degli archi (i,i) o autoanelli), è possibile evitare di elencare tutti gli archi. OPL offre due possibilità.

Definizione di insieme tramite proprietà degli elementi che vi appartengono

```
int NNodi = ...;
range Nodi = 1..NNodi;
tuple arco {
      int NodoOut;
      int NodoIn;
};
{arco} InsiemeArchi = {<i,j> | i in Nodi, j in Nodi : i != j};
int divergenza[Nodi] = ...;
int costoA[InsiemeArchi] = ...;
int capA[InsiemeArchi] = ...;
dvar float+ f[InsiemeArchi];
```

costoA, capA e f sono vettori…………

Oppure……

Oppure……si può evitare la dichiarazione "esplicita" dell'insieme di archi (tanto è noto che il grafo è completo!), trasformando la definizione di PesoA…..

```
int NNodi = ...;
range Nodi = 1..NNodi;
int divergenza[Nodi] = ...;
int costoA[Nodi][Nodi] = ...;
int capA[Nodi][Nodi] = ...;
dvar float+ f[Nodi][Nodi];
```

costoA, capA e f sono matrici…………

Nel file dat lungo la diagonale principale delle matrici costoA, capA si può inserire un valore mnemonico per identificare l'assenza degli archi (i,i), ad esempio un valore "sufficientemente grande" o "sufficientemente piccolo" a seconda delle necessità.

```
minimize  sum(i in Nodi)sum(j in Nodi : j != i)costoA[i][j]*f[i][j];

forall(i in Nodi)
sum(j in Nodi : j != i)f[i][j]-sum(j in Nodi : j!=i)f[j][i]==divergenza[i]
```

gli autoanelli non fanno parte del grafo

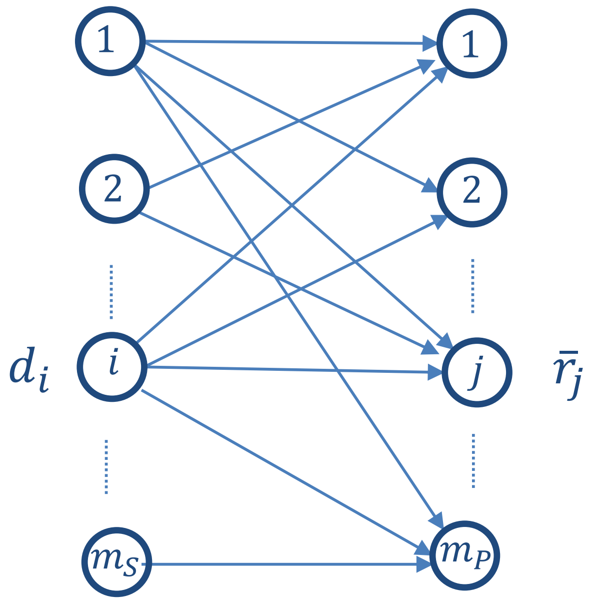
Il Problema dei Trasporti

Il problema dei Trasporti è un caso speciale del problema di flusso di costo minimo. E' definito su un grafo bipartito completo

- I nodi sono partizionati in due insiemi S e P (non ci sono nodi di transito);
- Esiste un arco tra ogni coppia di nodi (grafo completo);
- Non esistono archi tra nodi dello stesso insieme;

E' facile verificare che il grafo $G =< V, E >$ con le caratteristiche appena descritte è un Grafo Orientato Bipartito Completo con $V = S \cup P,\ \ S \cap P = \emptyset$. S è l'insieme delle sorgenti; P è l'insieme dei pozzi; $|S| = m_S$, $|P| = m_P$, $|V| = m = m_S + m_P$.

Insieme delle sorgenti
d_i le divergenze dei nodi in S



Insieme dei terminali
$\bar{r}_j$ divergenze dei nodi in P

Il Problema dei Trasporti: Formulazione

Variabili decisionali f_{ij}= quantità trasportata dalla sorgente "i" pozzo "j"

Data la struttura del grafo su cui il Problema dei Trasporti è definito (Grafo Bipartito Completo), l'equazione di bilancio al nodo i

```
flussoUscente(i)-flussoEntrante(i)=divergenza(i)
```

assume la seguente forma

a) Ciascuna sorgente ha solo archi uscenti e diretti verso ciascun pozzo

$$\sum_{j \in P} f_{ij} = d_i \quad \forall i \in S$$

b) Ciascun pozzo ha solo archi entranti provenienti da ciascuna delle sorgenti

$$-\sum_{i \in S} f_{ij} = \bar{r}_j \quad \forall j \in P$$

Ricordando che $\bar{r}_j < 0$, si può cambiare segno alle equazioni di bilancio dei pozzi

$$\sum_{i \in S} f_{ij} = r_j \quad \forall j \in$$

dove $-\bar{r}_j = r_j > 0$

Il Problema dei Trasporti: formulazione

$$\min \sum_{i \in S} \sum_{j \in P} c_{ij} f_{ij}$$

$$\sum_{j \in P} f_{ij} = d_i \qquad \forall i \in S$$

$$\sum_{i \in S} f_{ij} = r_j \qquad \forall j \in P$$

$$f_{ij} \geq 0 \qquad \forall i \in S, j \in P$$

$$\min \sum_{i=1}^{m_S} \sum_{j=1}^{m_P} c_{ij} f_{ij}$$

$$\sum_{j=1}^{m_P} f_{ij} = d_i \qquad \forall i = 1, .., m_S$$

$$\sum_{i=1}^{m_S} f_{ij} = r_j \qquad \forall j = 1, .., m_P$$

$$f_{ij} \geq 0 \qquad \forall i = 1, .. m_S, j = 1, .. m_P$$

Problema in forma standard

$$\begin{array}{ll} min & c^T f \\ & Af = b \\ & f \geq 0 \end{array}$$

$$A \in \mathbb{R}^{(m_S+m_P)\times(m_S m_P)} \quad b \in \mathbb{R}^{m_S+m_P} \quad b = \begin{bmatrix} d \\ r \end{bmatrix}$$

1) A matrice di incidenza di un grafo non orientato bipartito completo. Ogni colonna che rappresenta un ramo del grafo bipartito (i, j) ha esattamente due elementi non nulli entrambi pari ad 1: nella riga i $(1 \leq i \leq m_S)$ e nella riga $m_S + j$ $(1 \leq j \leq m_P)$.

2) c_{ij} costo necessario a trasportare un'unità di merce dalla sorgente i al pozzo j.

Proprietà del Problema dei Trasporti

$$\min \sum_{i=1}^{m_S} \sum_{j=1}^{m_P} c_{ij} f_{ij}$$

$$\sum_{j=1}^{m_P} f_{ij} = d_i \qquad \forall i = 1, .., m_S$$

$$\sum_{i=1}^{m_S} f_{ij} = r_j \qquad \forall j = 1, .., m_P$$

$$f_{ij} \geq 0 \qquad \forall i = 1, .. m_S, j = 1, .. m_P$$

La condizione necessaria di ammissibilità per il problema di flusso di costo minimo

$$\sum_{i \in V} b_i = 0$$

per il problema dei trasporti diventa

$$\sum_{i=1}^{m_S} d_i = \sum_{j=1}^{m_P} r_j = D$$

relazione di congruenza

La relazione di congruenza è anche una condizione sufficiente di ammissibilità

Infatti ponendo

$$f_{ij} = \frac{d_i r_j}{D} \quad \forall i \in S, j \in P$$

a) nei vincoli relativi alle sorgenti

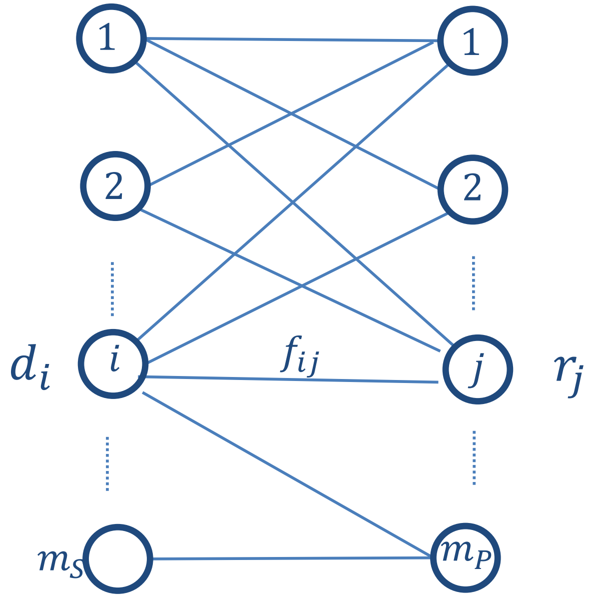
$$\sum_{j=1}^{m_P} f_{ij} = \sum_{j=1}^{m_P} \frac{d_i r_j}{D} = \frac{d_i}{D} \sum_{j=1}^{m_P} r_j = \frac{d_i}{D} D = d_i$$

b) nei vincoli relativi ai pozzi

$$\sum_{i=1}^{m_S} f_{ij} = \sum_{i=1}^{m_S} \frac{d_i r_j}{D} = \frac{r_j}{D} \sum_{i=1}^{m_S} d_i = \frac{r_j}{D} D = r_j$$

essi sono soddisfatti.

Proprietà del Problema dei Trasporti



1. Poiché vale

$$0 \leq f_{ij} \leq min\{d_i, r_j\} \; \forall i \in S, j \in P,$$

   le variabili sono limitate superiormente. Perciò il problema dei trasporti non è mai illimitato, anche in presenza di costi unitari di trasporto negativi.

2. Se vale la relazione di congruenza, il problema dei trasporti ammette sempre ottimo finito.

3. Come il problema di flusso di costo minimo, tutti i vertici di $\Omega(TP)$ hanno componenti intere (se le divergenze sono intere) e corrispondono ad alberi ricoprenti il grafo bipartito.

Osservazione

Qualora la relazione di congruenza non fosse soddisfatta, non si può scrivere il problema dei trasporti in forma standard. In particolare se

$$\sum_{i=1}^{m_S} d_i > \sum_{j=1}^{m_P} r_j$$

l'offerta totale eccede la domanda totale. Una frazione della merce $r_{m_P+1} = \sum_{i=1}^{m_S} d_i - \sum_{j=1}^{m_P} r_j$ non parte dalle sorgenti. Allora i vincoli di bilancio sulle sorgenti saranno

$$\sum_{j \in P} f_{ij} \leq d_i \qquad \forall i \in S$$

Ricordando che un vincolo di « $\leq$ » può essere trasformato in un vincolo di uguaglianza introducendo una variabile di slack, allora basta aggiungere a ciascuno dei vincoli precedenti la variabile $f_{i\,m_P+1}$ $\forall i \in S$

$$\sum_{j \in P} f_{ij} + f_{i\,m_P+1} = d_i \qquad \forall i \in S$$

ciascun arco $(i, m_P + 1)$ è un arco che incide sulla sorgente i e su un pozzo fittizio $m_P + 1$ con divergenza r_{m_P+1}. Il costo unitario di trasporto di tali archi è zero.

Il vincolo relativo al pozzo fittizio $m_P + 1$ può essere omesso, perché ridondante.

Osservazione

Se, invece, risulta

$$\sum_{i=1}^{m_S} d_i < \sum_{j=1}^{m_P} r_j$$

la domanda totale non può essere soddisfatta poiché eccede la domanda totale. Una frazione della merce $d_{m_S+1} = -\sum_{i=1}^{m_S} d_i + \sum_{j=1}^{m_P} r_j$ non giunge ai pozzi. Allora i vincoli di bilancio sui pozzi saranno

$$\sum_{i \in S} f_{ij} \leq r_j \qquad \forall j \in P$$

Ricordando che un vincolo di « $\leq$ » può essere trasformato in un vincolo di uguaglianza introducendo una variabile di slack, allora basta aggiungere a ciascuno dei vincoli precedenti la variabile $f_{S_0 j}$ $\forall j \in P$

$$\sum_{i \in S} f_{ij} + f_{m_S+1\ j} = r_j \qquad \forall j \in P$$

ciascun arco $(m_S + 1, j)$ è un arco che incide sulla sorgente fittizia $m_S + 1$ con divergenza d_{m_S+1} e sul pozzo j. Il costo unitario di trasporto di tali archi è zero.

Il vincolo relativo alla sorgente fittizia $m_S + 1$ può essere omesso perché ridondante.

Problema dei Trasporti

Un'azienda produttrice di acque minerali deve predisporre il piano settimanale di approvvigionamento dei propri punti vendita a partire dagli impianti di imbottigliamento. L'azienda dispone di 3 impianti di imbottigliamento localizzati a Verona, Pisa, Napoli e 3 punti vendita (Roma, Cosenza, Bari). Gli impianti hanno una capacità produttiva pari [12, 8, 10] quintali, mentre le richieste da parte dei punti vendita sono [8,10,9]. Nella Tabella seguente sono mostrati i costi di trasporto per quintali da ciascun impianto a ciascun punto vendita (i costi sono espressi in Euro). Si vuole aiutare l'azienda a determinare il piano di trasporto ottimale.

Punti Vendita

Impianti

200	500	200
300	400	600
100	500	300

$$\sum_{i=1}^{m_S} d_i = 12 + 8 + 10 = 30$$

$$\sum_{j=1}^{m_P} r_j = 8 + 10 + 9 = 27$$

$$\sum_{i=1}^{m_S} d_i > \sum_{j=1}^{m_P} r_j$$

abbiamo bisogno di un punto vendita fittizio con richiesta pari a 3 quintali a settimana

Formulazione in FS

$G =< S, P, E >$

$|S| = m_S = 3, |P| = m_P = 3 + 1 = 4;$ $\quad d = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 12 \\ 8 \\ 10 \end{bmatrix} \quad r = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} 8 \\ 10 \\ 9 \\ 3 \end{bmatrix} \quad c = \begin{bmatrix} 2 & 5 & 2 & 0 \\ 3 & 4 & 6 & 0 \\ 1 & 5 & 3 & 0 \end{bmatrix}$

$$\begin{cases} \begin{array}{lcccccccccccccc} min & 2f_{11} & +5f_{12} & +2f_{13} & +0f_{14} & +3f_{21} & +4f_{22} & +6f_{23} & +0f_{24} & +2f_{31} & +5f_{32} & +3f_{33} & +0f_{34} & & \\ & f_{11} & +f_{12} & +f_{13} & +f_{14} & & & & & & & & & = & 12 \\ & & & & & f_{21} & +f_{22} & +f_{23} & +f_{24} & & & & & = & 8 \\ & & & & & & & & & f_{31} & +f_{32} & +f_{33} & +f_{34} & = & 10 \\ & f_{11} & & & & +f_{21} & & & & +f_{31} & & & & = & 8 \\ & & f_{12} & & & & +f_{22} & & & & +f_{32} & & & = & 10 \\ & & & f_{13} & & & & +f_{23} & & & & +f_{33} & & = & 9 \\ & & & & f_{14} & & & & +f_{24} & & & & +f_{34} & = & 3 \\ & & & & & & & & & & & & f_{ij} & \geq & 0 \end{array} \end{cases}$$

$$\boxed{\begin{array}{ll} min & c^T f \\ & Af = b \\ & f \geq 0 \end{array}}$$

dove $b = \begin{bmatrix} d \\ r \end{bmatrix}$

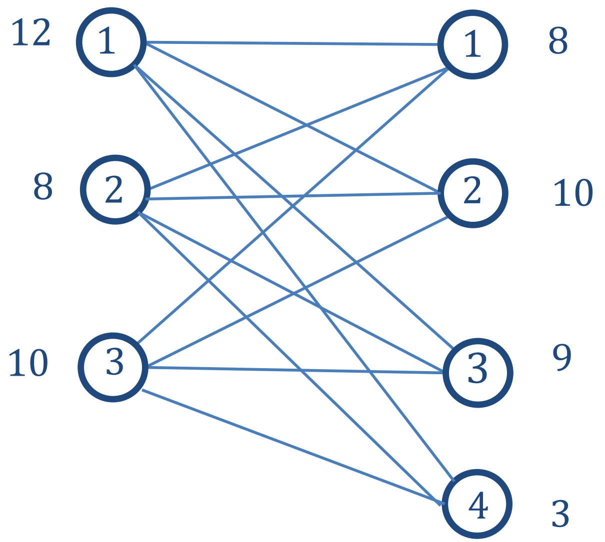
$$\begin{array}{c|cccccccccccc|} & (1,1) & (1,2) & (1,3) & (1,4) & (2,1) & (2,2) & (2,3) & (2,4) & (3,1) & (3,2) & (3,3) & (3,4) \\ \hline 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hline \end{array} = A$$

$\boxed{rank(A) = m_S + m_P - 1 \quad \text{un'equazione dipende linearmente dalle altre}}$

Esempio - Flussi -> Albero

$G =< S, P, E >$
$|S| = m_S = 3, |P| = m_P = 3 + 1 = 4;$

$d = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 12 \\ 8 \\ 10 \end{bmatrix}$ $\quad r = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} 8 \\ 10 \\ 9 \\ 3 \end{bmatrix}$ $\quad c = \begin{bmatrix} 2 & 5 & 2 & 0 \\ 3 & 4 & 6 & 0 \\ 1 & 5 & 3 & 0 \end{bmatrix}$



Consideriamo il seguente assegnamento di valori alle variabili

$f_{11} = 8, f_{12} = 4, f_{22} = 6, f_{23} = 2, f_{33} = 7, f_{34} = 3$

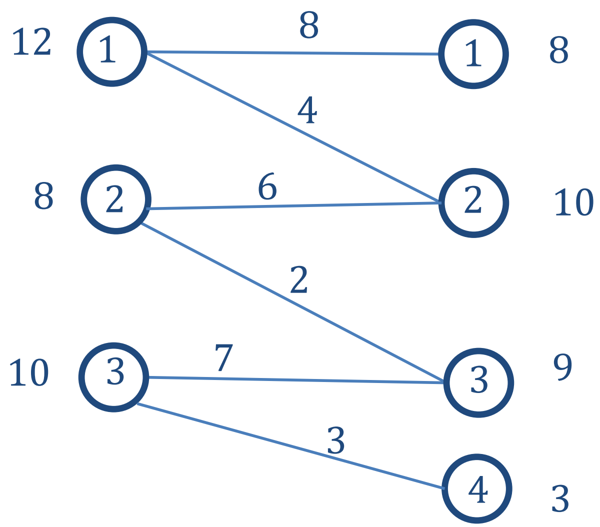
tutte le altre variabili pari a zero

Sostituendo tali valori nei vincoli, troviamo che $f \in \Omega(TP)$.

Costo della soluzione

$z(f) = 2\times8 + 5\times4 + 4\times6 + 6\times2 + 3\times7 + 0\times3 = 93$

Considerando solo i rami con flusso positivo



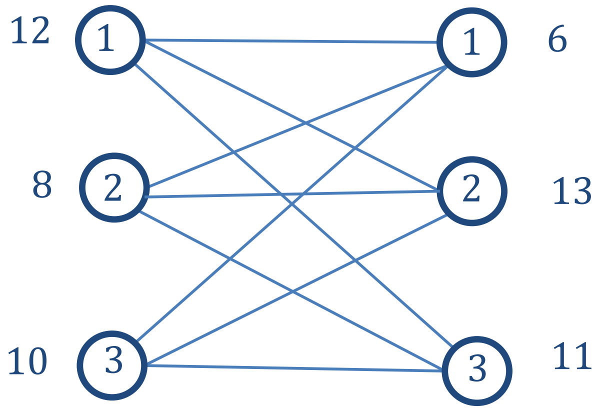
Albero ricoprente

$f \in \Omega(TP)$ è un vertice di

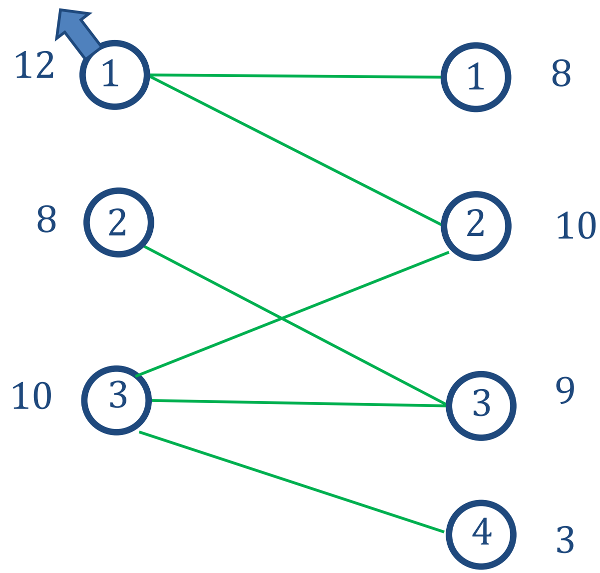
Esempio - Albero -> Flussi

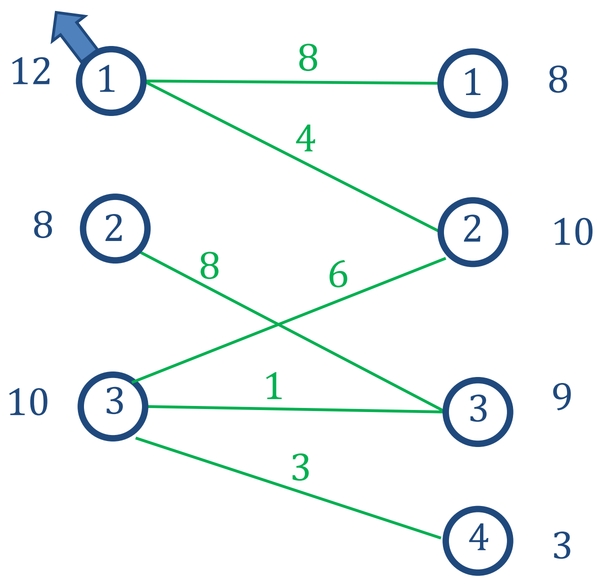
$G =< S, P, E >$
$|S| = m_S = 3, |P| = m_P = 3;$

$$d = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 12 \\ 8 \\ 10 \end{bmatrix} \qquad r = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} 8 \\ 10 \\ 9 \\ 3 \end{bmatrix} \qquad c = \begin{bmatrix} 2 & 5 & 2 & 0 \\ 3 & 4 & 6 & 0 \\ 1 & 5 & 3 & 0 \end{bmatrix}$$



Consideriamo il seguente Albero ricoprente ($m_S + m_P - 1$ rami).
Risolviamo il sistema di equazioni di continuità a partire dalle foglie verso la radice

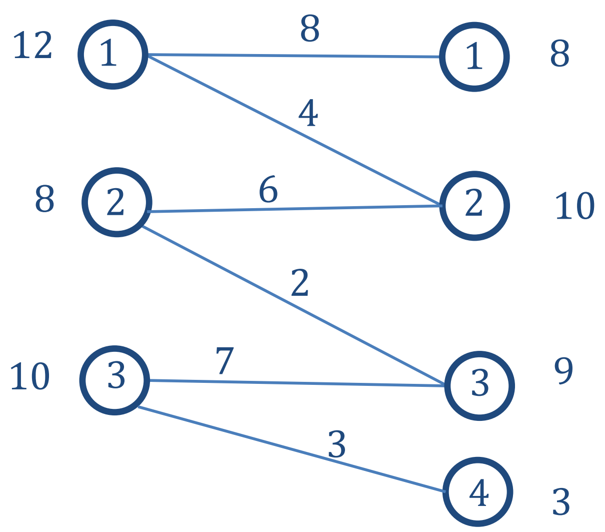


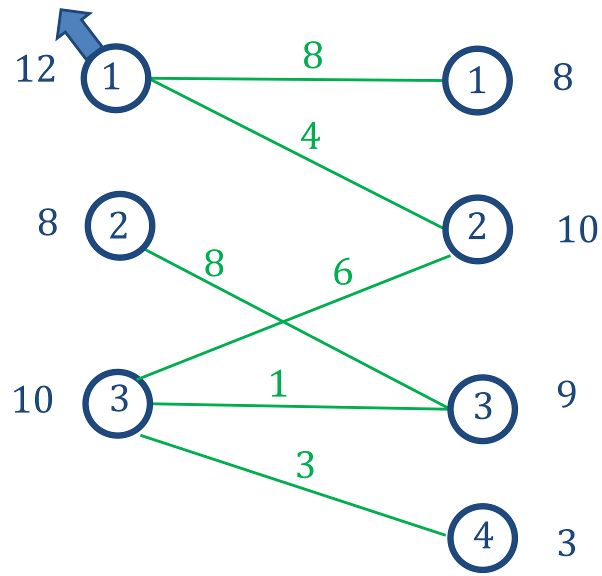


Costo della soluzione

$z(f) = 2\times 8 + 5\times 4 + 6\times 8 + 5\times 6 + 3\times 1 + 0\times 3 = 117$

Esempio





I due Alberi differiscono per un ramo --> i corrispondenti vertici sono adiacenti. Il cambio di base si effettua in modo analogo a quanto visto per il MCFP

Formulazione OPL del Problema dei Trasporti in forma non standard

```
int nOrigini=...;
int nDestinazioni=...;

range Origini = 1..nOrigini;
range Destinazioni = 1..nDestinazioni;

int D[Origini]=...;
int R[Destinazioni]=...;
int C[Origini][Destinazioni]=...;

dvar float+ F[Origini][Destinazioni];

minimize sum(i in Origini, j in
Destinazioni)C[i][j]*F[i][j];
subject to{
  forall( i in Origini)
  origine_i:  sum(j in Destinazioni)F[i][j]<=D[i];

  forall( j in Destinazioni)
  destinazione_j:  sum(i in Origini)F[i][j]==R[j];
}
```

3 origini
3 destinazioni

$$c = \begin{bmatrix} 2 & 5 & 2 \\ 3 & 4 & 6 \\ 1 & 5 & 3 \end{bmatrix}$$

$$d = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} 12 \\ 8 \\ 10 \end{bmatrix}$$

$$r = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 10 \\ 9 \end{bmatrix}$$

Il Problema dell'Assegnamento

Problema dei Trasporti

$$\min \sum_{i=1}^{m_S} \sum_{j=1}^{m_P} c_{ij} f_{ij}$$

$$\sum_{j=1}^{m_P} f_{ij} = d_i \qquad \forall i = 1,..,m_S$$

$$\sum_{i=1}^{m_S} f_{ij} = r_j \qquad \forall j = 1,..,m_P$$

$$f_{ij} \geq 0 \qquad \forall i = 1,..m_S\,, j = 1,..m_P$$

$$m_S = m_P = m$$
$$d_i = 1, \forall i \in S$$
$$r_j = 1, \forall j \in P$$

$\Longrightarrow$

Problema dell'Assegnamento (AP)

$$\min \sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij} f_{ij}$$

$$\sum_{j=1}^{m} f_{ij} = 1 \qquad \forall i = 1,..,m$$

$$\sum_{i=1}^{m} f_{ij} = 1 \qquad \forall j = 1,..,m$$

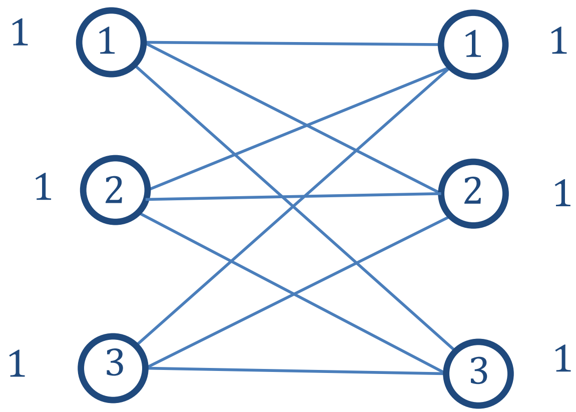
$$f_{ij} \geq 0 \qquad \forall i,j = 1,..m$$

AP eredita tutte le proprietà del Problema dei Trasporti
1) Ammette sempre ottimo finito.
2) $0 \leq f_{ij} \leq \min\{d_i, r_j\} = \min\{1,1\} = 1$
3) I vertici di $\Omega(AP)$ corrispondono ad alberi ricoprenti e sono a componenti intere.
4) 2) + 3) I vertici hanno componenti $\in \{0,1\}$

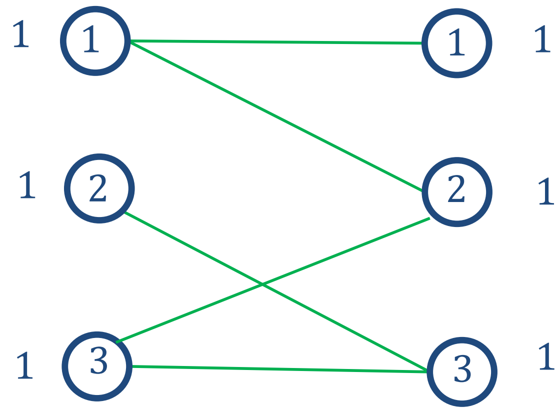
Esempio - Albero -> Flussi

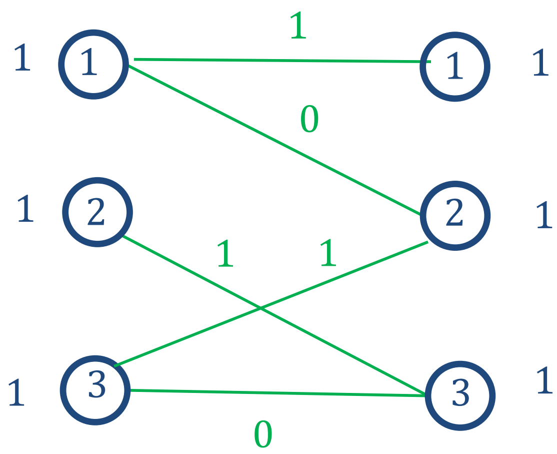
$G = < S, P, E >$
$|S| = |P| = m = 3;$

$$c = \begin{bmatrix} 2 & 5 & 2 \\ 3 & 4 & 6 \\ 1 & 5 & 3 \end{bmatrix}$$



Consideriamo il seguente Albero ricoprente ($2m - 1$ rami).
Nel sistema di vincoli eliminiamo un'equazione e poniamo a zero le variabili di flusso associate a rami non presenti nell'albero

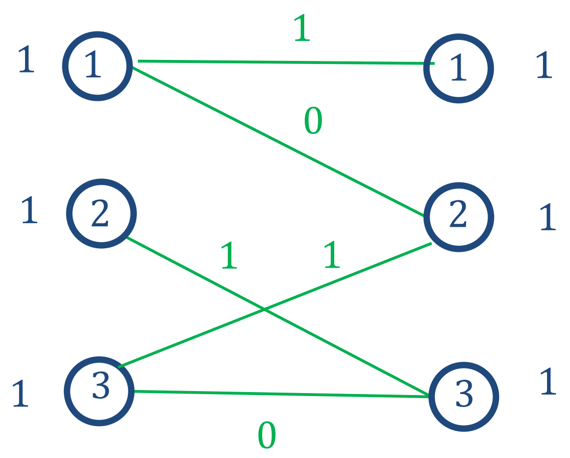




La soluzione è ammissibile (>=0) e corrisponde ad un vertice

Costo della soluzione
$z(f) = 2\times1 + 6\times1 + 5\times1 = 13$

Esempio – Osservazioni sulla soluzione



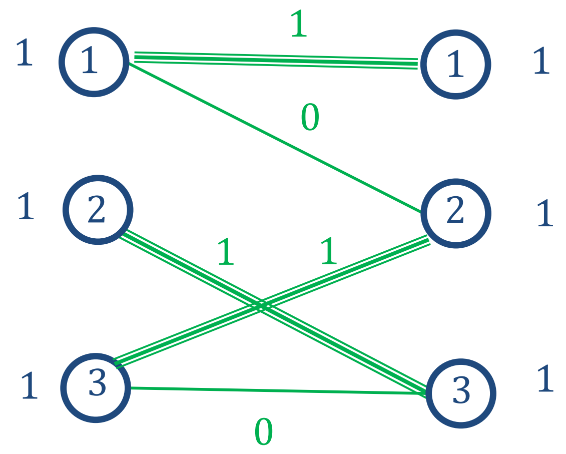
1) Nel Problema dell'Assegnamento tra le 2m-1 variabili di flusso associate ad altrettanti rami dell'albero ricoprente, esattamente m sono non nulle (e valgono 1). Le rimanenti m-1 sono nulle pur essendo associate a rami dell'albero.

2) La proprietà 1 vale per tutti i vertici di $\Omega(AP)$. Perciò tutti i vertici (o tutte le sab) di *AP* sono degeneri.

Interpretando gli archi con flusso non nullo come «coppie di nodi», i vertici di AP rappresentano m accoppiamenti tra i nodi di S e quelli di P.

Ogni nodo di S è accoppiato con un nodo di P (ed uno solo)

$$\sum_{j=1}^{m} f_{ij} = 1 \qquad \forall i = 1,..,m$$



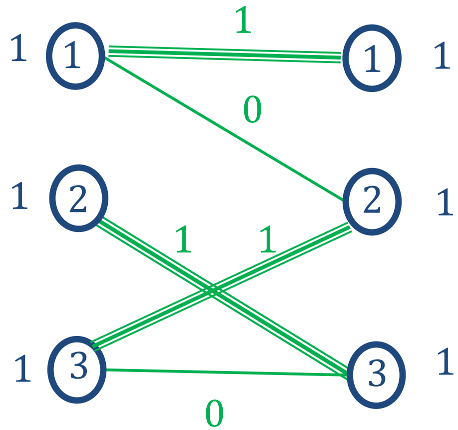
Ogni nodo di P è accoppiato con un nodo di S (ed uno solo)

$$\sum_{i=1}^{m} f_{ij} = 1 \qquad \forall j = 1,..,m$$

Il problema dell'Assegnamento è, quindi, quello di formare, in modo ottimale, m coppie di oggetti non omogenei (insiemi disgiunti S e P), come ad esemptio (compito, lavoratore), (task, cpu), (arbitro, partita), in modo tale che gli oggetti di S e P compaiano tutti in una ed una sola coppia. In maniera equivalente si può dire che dati due insiemi disgiunti S, P , di uguale cardinalità, si vuole accoppiare ogni elemento di S con uno ed uno solo elemento di P (e viceversa)

1. $\Omega(AP)$ è un ipercubo unitario con $m!$ vertici.

2. Un vertice può essere rappresentato (oltre che da un albero ricoprente) da una matrice $f \in \mathbb{R}^{m^2}$ con elementi $\{0,1\}$



$$f = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

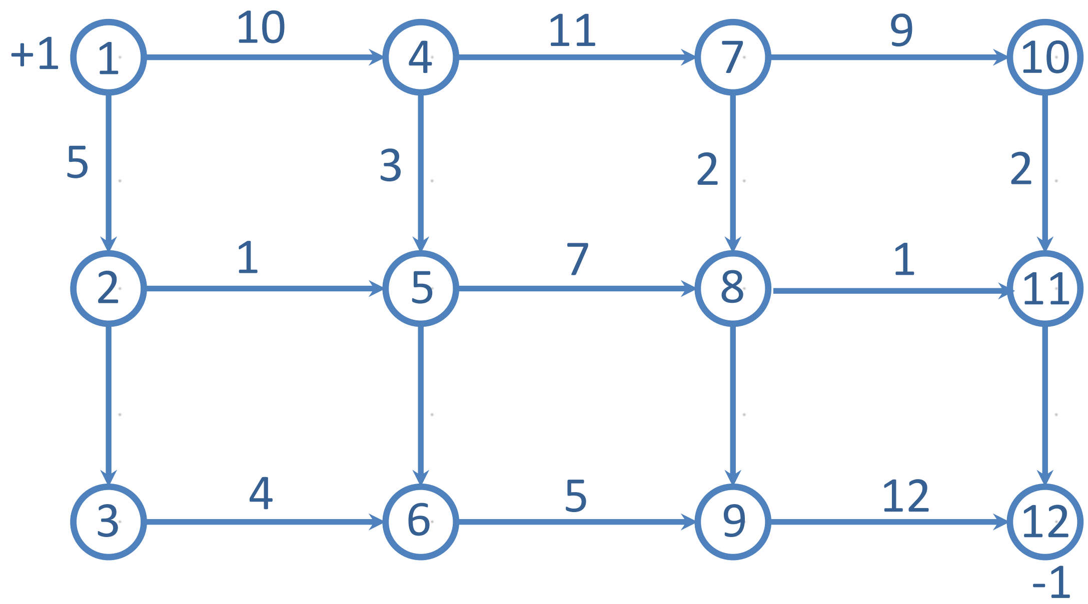
Esattamente un «1» su ogni riga e colonna

3. Tutti i vertici si possono ottenere «permutando» le colonne della matrice f

$$f^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} f^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} f^3 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} f^4 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} f^5 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} f^6 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Esercizio per casa 1

Implementare in OPL il problema del cammino di costo minimo per il grafo in figura con S=1 e T=12 (Esercizio per casa)



Esercizio per casa 2

Come si modifica la formulazione OPL del Problema dei Trasporti (slide 25) nel caso in cui volessimo formulare il problema in forma standard?

Esercizio per casa 3

Si considerino 5 processi (TASKS) e altrettante CPU. Per ogni coppia (TASK, CPU) è noto il tempo che la CPU impiega a processare il TASK. Si vuole determinare l'accoppiamento ottimale TASK-CPU che minimizzi il tempo totale di calcolo.

TASKS \ CPU					
	7	2	1	9	4
	9	6	9	5	5
	3	8	3	1	8
	7	9	4	2	2
	8	4	7	4	8

Esercizio per casa 4

Provare ad implementare in OPL il problema del Massimo Flusso ricordando che:

1) Nella rete ci sono due nodi speciali (S e T)
2) Gli archi sono pesati con le capacità

Testare infine il modello OPL con la seguente rete in cui S=1, T=8