LABORATORIO DI RICERCA OPERATIVA 2020-2021

Laboratorio OPL - LEZIONE 3

Grafi orientati e pesati in OPL

Implementazione 1

```
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] = ...;
```

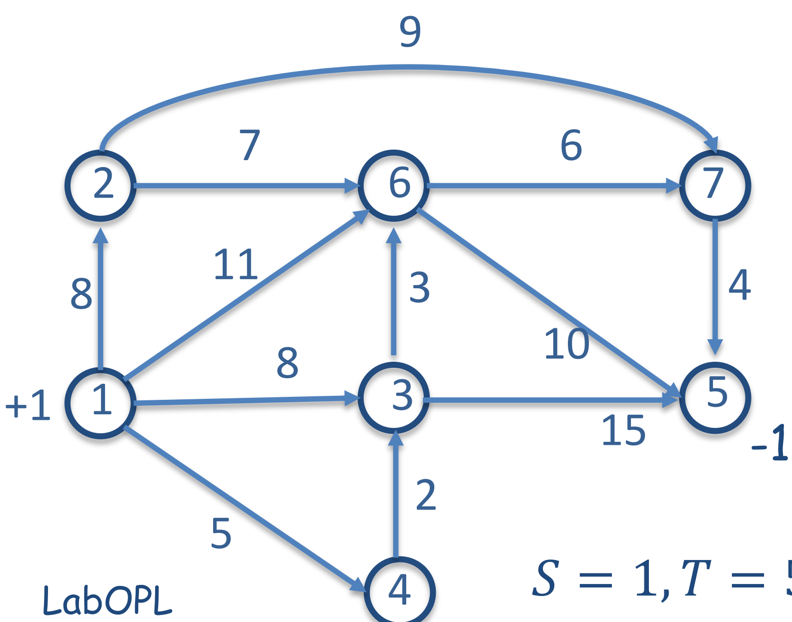
```
tuple arco {
        int NodoOut;
        int NodoIn;
};
{arco} InsiemeArchi = ...;
int costoA[InsiemeArchi] = ...;
int capA[InsiemeArchi] = ...;
```

Cammino di Costo Minimo dal nodo S al nodo T

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

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

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



$S = 1, T = 5$

E' un caso particolare del problema di flusso di costo minimo

1) Nel grafo è presente un solo nodo sorgente S con divergenza +1 un solo nodo pozzo T con divergenza -1 Tutti i restanti nodi hanno divergenza nulla.

2) Costi sugli archi >= 0

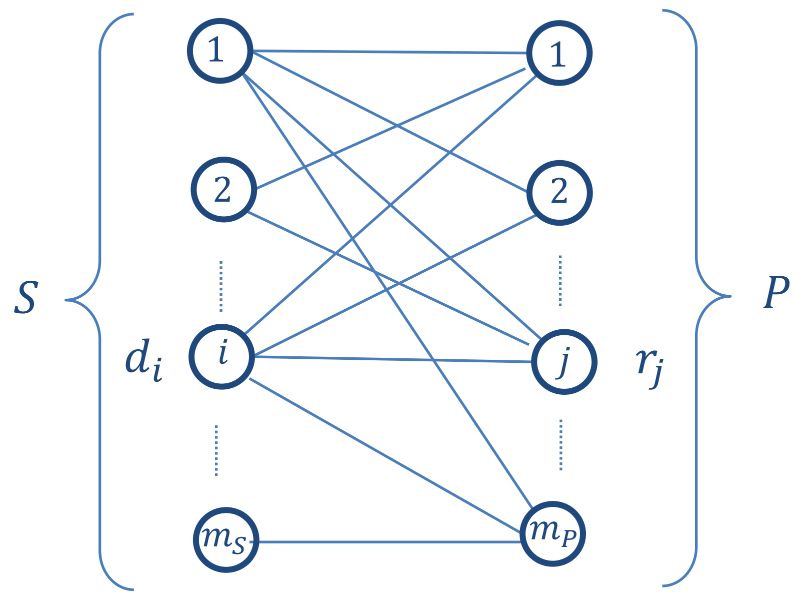
In tutte le soluzioni ammissibili di base l'unica unica unità di flusso che la sorgente eroga raggiungerà il pozzo lungo un "cammino".

Poiché i costi sugli archi sono costi unitari di flusso, la soluzione ottima rappresenta il cammino di costo minimo da S a T.

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);
- Le divergenze dei nodi in S e P sono sempre positive (chiamiamo d_i le divergenze dei nodi in $i \in S$; r_j quelle dei nodi in $j \in P$);
- Esiste un arco tra ogni coppia di nodi (grafo completo);
- Non esistono archi tra nodi dello stesso insieme;



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

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

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

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

Il Problema dei Trasporti: formulazione

$|S| = m_S, \ |P| = m_P$

Variabile di flusso f_{ij}= quantità di merce trasportata dall'origine "i" alla destinazione "j"

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

$$\boxed{\begin{array}{ll} \min \sum_{i=1}^{m_S} \sum_{j=1}^{m_P} c_{ij} f_{ij} & \\ \sum_{j=1}^{m_P} f_{ij} = d_i & \forall i = 1,\ldots, m_S \\ \sum_{i=1}^{m_S} f_{ij} = r_j & \forall j = 1,\ldots, m_P \\ f_{ij} \geq 0 & \forall i = 1,\ldots m_S, j = 1,\ldots m_P \end{array}}$$

Se $\sum_{i=1}^{m_S} d_i > \sum_{j=1}^{m_P} r_j$: basta aggiungere un pozzo fittizio $m_P + 1$, con divergenza

$r_{m_P+1} = \sum_{i=1}^{m_S} d_i - \sum_{j=1}^{m_P} r_j$ e costi sugli archi $c_{i\, m_P+1} = 0 \ \ \forall i \in S$

Se $\sum_{i=1}^{m_S} d_i < \sum_{j=1}^{m_P} r_j$: basta aggiungere una sorgente fittizia $m_S + 1$, con divergenza

$d_{m_S+1} = \sum_{j=1}^{m_P} r_j - \sum_{i=1}^{m_S} d_i$ e costi sugli archi $c_{m_S+1\, j} = 0 \ \ \forall j \in P$

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];
```

costo*A*, cap*A* 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]
```

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 5 punti vendita (Roma, Cosenza, Bari, Milano, Genova). Gli impianti hanno una capacità produttiva pari [500, 300, 750] quintali, mentre le richieste da parte dei punti vendita sono [487, 207, 272, 295, 286]. 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

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

$$\sum_{i=1}^{m_S} d_i = 500 + 300 + 750 = 1550$$

$$\sum_{j=1}^{m_P} r_j = 487 + 207 + 272 + 295 + 286 = 1547$$

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

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

Il Problema di Assegnamento

E' un particolare problema dei trasporti, caratterizzato da $m_S = m_P = m$ e divergenze unitarie su tutti i nodi.

Variabile di flusso f_{ij} vale 0 o 1 in qualsiasi soluzione ammissibile di base

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

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

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

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

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.

CPU

TASKS

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

Problema di Massimo Flusso

$$\max \sum_{(s,j)\in E} f_{sj}$$

Tutti i nodi della rete, esclusi S e T, sono nodi di transito, il valore del flusso che circola nella rete può essere misurato come flusso uscente da S (o flusso entrante in T

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

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

$$0 \leq f_{ij} \leq u_{ij} \qquad (i,j) \in E$$

Vincoli di capacità sugli archi

$$\begin{array}{llllllll}
\max & & & & v & & & \\
\sum_{(s,j)\in E} f_{sj} & & & & -v & = & 0 & \\
\sum_{j|(i,j)\in E} f_{ij} & - & \sum_{j|(j,i)\in E} f_{ji} & & & = & 0 & i \in V, i \neq S,T \\
& - & \sum_{(i,T)\in E} f_{T} & & +v & = & 0 & \\
& 0 & \leq & & f_{ij} & \leq & u_{ij} & (i,j) \in E
\end{array}$$

Aggiungendo nella formulazione precedente i vincoli di continuità sui nodi S e T

Massimo Flusso comeProblema di Circolazione

Problema del Massimo Flusso (Esercizio Per casa)

Implementare in OPL il Problema del Massimo Flusso come Problema di Circolazione. Risolvere l'istanza del problema definita dal seguente digrafo, in cui i pesi sugli archi rappresentano le capacità degli archi stessi