# FLUSSO SU RETE

domenica 18 agosto 2024 11:15

## INTRODUZIONE

### Definizioni

Sia $G=(N,A)$ un grafo orientato con $\overset{\text{nodi}}{|N|}=n$ $\overset{\text{archi}}{|A|}=m$

$x_{ij}$ :: variabili decisionali :: quantità di bene (flusso) inviata lungo $(i,j)$ $\forall (i,j)\in A$

$b_i \in \mathbb{R}$ :: bilancio del nodo $i$, $\forall i \in N$. Per convenzione
- $b_i<0$ se $i$ è un nodo di origine (sorgente)
- $b_i>0$ se $i$ è un nodo di destinazione (pozzo)
- $b_i=0$ se $i$ è un nodo di transito

$u_{ij}\geq 0$ :: capacità di $(i,j)$ $\forall (i,j)\in A$

$c_{ij}\in\mathbb{R}$ :: costo unitario di invio lungo $(i,j)$, $\forall (i,j)\in A$

Il problema di flusso di costo minimo consiste nel decidere come inviare il bene lungo la rete di flusso, a costo minimo, rispettando i bilanci dei nodi e le capacità dei collegamenti.

### Modello matematico di flusso su rete

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

$$\sum_{(j,i)\in BS(i)} x_{ji} - \sum_{(i,j)\in FS(i)} x_{ij} = b_i \quad \forall i\in N$$ vincoli di bilancio o di conservazione del flusso

$$0\leq x_{ij}\leq u_{ij} \quad \forall (i,j)\in A$$ vincoli di capacità

$BS(i)$ := Backward Star di $i$
$FS(i)$ := Forward Star di $i$


In forma compatta

| | |
|---|---|
| $\min\ c^T x$ | $c=[c_{ij}]$ :: vettore dei costi |
| $Ex=b$ | $b=[b_i]$ :: vettore dei bilanci |
| $0\leq x\leq u$ | $u=[u_{ij}]$ :: vettore delle capacità |
| | $x=[x_{ij}]$ :: vettore delle v. di flusso |
| | $E$ : matrice di incidenza della rete |

Siano $D=\{i\in N \text{ t.c. } b_i>0\}$ :: insieme dei nodi di destinazione
$O=\{i\in N \text{ t.c. } b_i<0\}$ :: insieme dei nodi origine

### condizione necessaria di ammissibilità

$$\boxed{\sum_{i\in D} b_i = -\sum_{i\in O} b_i}$$

# Cammini cicli e alberi.

## Definizioni

Dato $G=(N,A)$ orientato un cammino $P$ da $r\in N$ a $t\in N$ e' una sequenza di nodi $P=(i_0,i_1,\dots,i_q)$ tali che $(i_{h-1},i_h)\in A$ $h=1,\dots,q$

Un cammino e' un ciclo se $i_0=i_q$

Un cammino e' semplice se non contiene ripetizioni di nodi

Un albero e' un grafo connesso e privo di cicli.

proprieta' → Quindi se $T=(N,A)$ e' un albero e $|N|=n \Rightarrow |A|=n-1$

def eq. → Un albero e' un grafo connesso tale che $|A|=n-1$

def. eq. → Un albero e' un grafo aciclico tale che $|A|=n-1$

Un albero e' radicato quando un suo nodo e' selezionato come nodo radice
e' orientato se i suoi archi sono orientati da padre a figlio

proprieta' → In un albero orientato e radicato $\exists!$ cammino dalla radice a ogni nodo dell'albero e la sua lunghezza (numero di archi) e' uguale al livello del nodo dell'albero

Dato $G=(N,A)$, un albero $T=(N_T, A_T)$, $A_T \subseteq A$ e' detto albero di copertura di $G$

## IL PROBLEMA DEI CAMMINI MINIMI

Caso speciale del problema di flusso di costo minimo

$|O|=|D|=1$ $\quad O=\{r\}$ $\quad D=\{t\}$

$c_{ij}\in\mathbb{R}$ $\quad \forall (i,j)\in A$

$u_{ij}=+\infty$ $\quad \forall (i,j)\in A$ ⟶ non capacitato

$b_r=-1$, $b_t=1$ ⟶ $r$ invia a $t$ un'unita' di flusso

## Modello matematico

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

$$\sum_{(j,i)\in BS(i)} x_{ji} - \sum_{(i,j)\in FS(i)} x_{ij} = \begin{cases} 1 & \text{se } i=t \\ -1 & \text{se } i=r \\ 0 & \text{altrimenti} \end{cases}$$

$$x_{ij}\geq 0 \quad \forall (i,j)\in A$$

## Definizioni

dato un cammino $P$ $\quad c(P)=\sum_{(i,j)\in P} c_{ij}$ e' detto costo di $P$

$\mathcal{P}_{rt}$ e' l'insieme dei cammini da $r$ a $t$ in $G$

Problema del cammino minimo ⟶ Dati $r$ e $t$, trova $P\in\mathcal{P}_{rt}$ che sia di costo minimo

Problema dell'albero dei cammini minimi ⟶ Dato $G=(N,A)$ dato $r$, trova un cammino di costo minimo da $r$ a $i$ $\forall i\neq r$

## Oss:

- proprieta' di concatenazione: se $(i\,j\,k)$ e' un cammino minimo da $i$ a $k \Rightarrow (i\,j)$ e' un cammino minimo da $i$ a $j$
- Gli alberi di copertura radicati a $r$ e orientati sono le sol di base ammissibili del modello di PL precedente
- $G$ e' fortemente connesso, ovvero in $G$ $\exists$ un cammino orientato da $r$ a $i$ $\forall i\neq r$.
  se non esiste basta aggiungere un arco fittizio $(r,i)\notin A$ di costo $M\geq \underbrace{(n-1)c_{max}+1}_{\text{max costo in } G}$
- Se $G$ non ha cicli negativi ogni cammino min e' semplice

# Visita di un grafo

$G=(N,A)$ grafo orientato

$r \in N$ nodo radice

Problema ⟶ det. l'insieme di nodi raggiungibili da r in G mediante un cammino orientato restituendo $T_r=(N_r, A_r)$ albero della visita con $N_r \subseteq N$ $A_r \subseteq A$

**Formalmente**

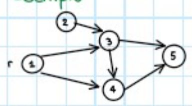
```
Procedura visita(G,r,p)

begin
for  i=1  to n   do p(i)=0;
p(r)=nil;
Q={r};
repeat
i=SELECT(Q);
Q=Q\{i};
for each (i,j) ∈ FS(i)  do
        if  p(j)=0  then  begin
                              p(j)=1;
                              Q=Q U {j};
                              end
until Q=ϕ
end
```

Complessità in tempo : $O(m)$ se select richiede tempo costante

Correttezza: non dipende da ordine di esame degli archi in FS(i) e neanche dal criterio di selezione in select

**Esempio**



**inizializzazione**

$Q=\{r\}=\{1\}$
$p(1)=nil$
$p(2)=p(3)=p(4)=p(5)=0$

**it 1)**

$i=1$ $Q=\phi$
$FS(1)$ $(1\,3)$ : $p(3)=1$
$(1\,4)$ : $p(4)=1$
$Q=\{3,4\}$

**it 2)**

$i=3$ $Q=\{4\}$
$FS(3)$ $(3\,5)$ : $p(5)=3$
$(3\,4)$ : —
$Q=\{4,5\}$

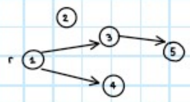
**it 3)**

$i=4$ $Q=\{5\}$
$FS(4)$ $(4\,5)$ : —
$Q=\{5\}$

**it 4)**

$i=5$ $Q=\phi$ STOP

$T_r$

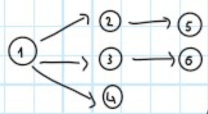


definito tramite i predecessori

## Implementazioni diverse

Q coda ⟶ strategia FIFO = First in First out
("i primi a entrare sono i primi a uscire")

$T_r$

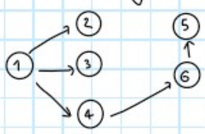


$Q=(\underset{r}{1}\ 2\ 3\ 4\ 5\ 6)$

cammini corti nel senso di n° di archi

Q stak ⟶ strategia LIFO Last in First out

$T_r$



5
6
4
3
2
1

"gli ultimi a entrare sono i primi a uscire"

## Teorema 1

Se Q e' implementato come una coda, la procedura visita individua un cammino con il minimo numero di archi da r a i per ogni i raggiungibile da r

## Corollario 1

Se $c_{ij} = 1 \quad \forall (i,j) \in A$, il problema dei cammini minimi e' risolubile in $O(m)$

# Alberi, etichette e condizioni di ottimo

**condizione necessaria per l'ottimalita' di T è** $\boxed{d(i) + c_{ij} \geq d(j) \quad \forall (i,j) \in A}$ $\quad d(i) =:$ etichetta

Condizione di Bellman

## Teorema 2

Dato $G = (N, A)$, sia $T = (N, A_T)$ albero di copertura radicato a r e orientato
Sia $d(i)$ l'etichetta di i $\forall i \in N$ (ovvero $d(r) = 0$ e $d(i)$ costo dell'unico cammino da r a i in T con $r \neq i$)
T e' un albero di cammini minimi di radice r $\Leftrightarrow d(i) + c_{ij} \geq d(j) \quad \forall (i,j) \in A$

## Lemma 1 (serve per dim Teo 2)

Sia $d \in \mathbb{R}^n$ t.c. 1) $d(r) = 0$
2) $d(i) + c_{ij} \geq d(j) \quad \forall (i,j) \in A$

Allora $d(i)$ e' una valutazione inferiore del costo del cammino minimo da r a i $\forall i \in N$

dim
$i = r$ caso banale
$i \neq r$, P cammino da r a i in G

$$P = \underset{\substack{\| \\ r}}{v_1} \to v_2 \to v_3 \to \dots \to v_{k-2} \to v_{k-1} \to \underset{\substack{\| \\ i}}{v_k}$$

Le cond. di Bellman valgono per ogni arco in A per hp $\Rightarrow$ valgono per gli archi in P

$$d(i) = d(v_k) \leq d(v_{k-1}) + c_{v_{k-1} v_k}$$
$$d(v_{k-1}) \leq d(v_{k-2}) + c_{v_{k-2} v_{k-1}}$$
$$\vdots$$
$$d(v_3) \leq d(v_2) + c_{v_2 v_3}$$
$$d(v_2) \leq \underset{\substack{\| \\ r}}{d(v_1)} + c_{v_1 v_2}$$

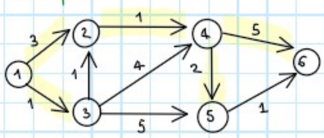
Sommo membro a membro $\longrightarrow$ $\underset{\substack{\| \\ d(i)}}{d(v_k)} + \cancel{d(v_{k-1})} + \dots + \cancel{d(v_3)} + \cancel{d(v_2)} \leq \cancel{d(v_{k-1})} + \cancel{d(v_{k-2})} + \dots + \cancel{d(v_2)} + \underset{\substack{\| \\ d(r)}}{d(v_1)} + \overbrace{c_{v_{k-1} v_k} + c_{v_{k-2} v_{k-1}} + \dots + c_{v_2 v_3} + c_{v_1 v_2}}^{C(P)}$

$\Rightarrow d(i) \leq \underset{\substack{\| \\ 0}}{d(r)} + c(P) \quad \Rightarrow \quad \boxed{d(i) \leq c(P)} \quad \forall$ P cammino da r a i

dim Teo 2

$\Rightarrow$ cond. necessaria gia' dim
$\Leftarrow$ Sia $d(r) = 0$ e $d(i)$ il costo dell'unico cammino in T da r a i $\forall i \neq r$
Per hp valgono le c. di Bellman cioè $d(i) + c_{ij} \geq d(j) \quad \forall (i,j) \in A$
Per il lemma $d(i) \leq c(P)$
$\hookrightarrow$ costo del cammino min da r a i
poiche' $d(i) =:$ e' il costo dell'unico cammino min in T da r a i $\forall i \neq r$
segue che T e' un albero di cammini min di radice r.

## Esempio



T e' una sol ottima?

Calcolo l'insieme di etichette

$d(1) = 0$
$d(2) = d(1) + 3 = 3$
$d(3) = d(1) + 1 = 1$
$d(4) = d(2) + 1 = 4$
$d(5) = d(4) + 2 = 6$
$d(6) = d(4) + 5 = 9$

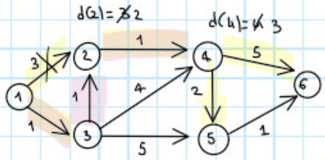
Considero un arco non in T es: (3 2)

$d(3) + c_{32} \overset{?}{\geq} d(2)$  NO $\Rightarrow$ T non e' un albero di cam. min di radice 1

$1 + 1 < 2$

il cammino da 1 a 2 può essere migliorato

$d(2) = \cancel{3}\,2$ $d(4) = \cancel{4}\,3$



$d(6) = \cancel{9}\;8$

## Schema algoritmo SPT (Shortest Processing time)

**Inizializzazione**

T: albero di copertura radicato in r e orientato
$d(i), \forall i \in N$ etichette
$p(i), \forall i \in N$ predecessori

**tipica iterazione**

se $\exists\ (i,j) \in A$ tale che $d(i) + c_{ij} < d(j)$:
- $p(j) = i$ (aggiornamento di T)
- aggiornamento di $d(i)\ \forall i \in N$

altrimenti STOP: T e' ottimo

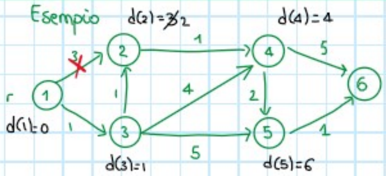
**Formalmente**

```
                         output
Procedura SPT(G,c,r,p,d)

begin
for  i=1 to n   do
Begin
p(i)=r;
d(i)=M;
end
p(r)=nil;
d(r)=0;
Q={r};
repeat
SELECT i from Q ;
Q=Q\{i};
for each (i,j) ∈ FS(i)  do
        if  d(i)+c_ij <d(j) then
                begin
                 d(j)=d(i)+c_ij ;
                        p(j)=i;
                        if  j ∉ Q then Q=Q U {j};
                        end
until Q=ϕ
end
```

correttezza di SPT $\longrightarrow$ Quando $Q = \phi$ le c. di Bellman sono soddisfatte per ogni arco in G.
L'ultimo albero determinato e' quindi ottimo

Esempio



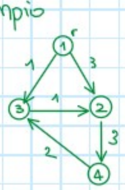

T: $p(1) = nil$;
$p(2)=1$, $p(3)=1$, $p(4)=2$, $p(5)=4$ ($p(i) = j$ indica il nodo $j$ da cui deriva il nodo $i$)

Poiché $\underset{1}{d(3)} + 1 < \underset{3}{d(2)}$ • $p(2)=3$ aggiorno T
• $d(2)=2$ aggiorno d

Per motivi di efficienza l'aggiornamento di $d(4)$, $d(5)$, $d(6)$ viene differito
Conseguenze

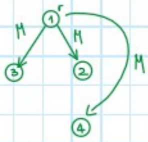
1) $d(i)$ è un'appross. superiore del costo del cammino da r a i in T
2) Viene introdotta una struttura dati Q, insieme dei nodi candidati $Q=\{2\}$ *
3) * segnala che vanno controllate le con. di Bellman per gli archi in FS(2)

Esempio



Inizializzazione

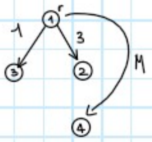
$p(1) = nil$
$p(2)=p(3)=p(4)=1$
$d(1)=0$
$d(2)=d(3)=d(4)=M$
$Q=\{1\}$



it 1)

$i=1$
FS(1): (1 2) : $d(1)+3 \leq \overset{M}{d(2)} \longrightarrow d(2)=3$
(1 3) $d(1)+1 \leq \underset{M}{d(3)} \longrightarrow d(3)=1$

$Q=\{2,3\}$

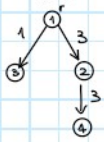


$d(4)=M$

it 2)

$i=2$
FS(2): (2 4) $\overset{3}{d(2)}+3 \leq \overset{M}{d(4)} \longrightarrow d(4)=6$
$p(4)=2$

$Q=\{3,4\}$

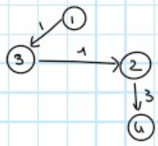


$d(2)=3$
$d(3)=1$

it 3)

$i=3$
FS(3) (3,2) : $\overset{1}{d(3)} + 1 < \overset{3}{d(2)} \Rightarrow d(2)=2$
$p(2)=3$

$Q=\{2,4\}$

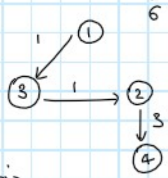


$d(3)=1$
$d(4)=6$

it 4)

$i=2$
FS(2) (2 4) $\overset{2}{d(2)}+3 < \underset{6}{d(4)} \Rightarrow d(4)=5$

$Q=\{4\}$



$d(2)=2$
$d(3)=1$

T non cambia

it 5)

$i=4$
FS(4)= 3 (4, 3) : $\overset{5}{d(4)}+2 \leq \overset{1}{d(3)}$ NO $\Rightarrow$ (4, 3) : —

$Q=\emptyset$ STOP

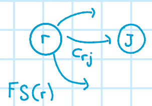
## Teorema 3

A ogni iterazione di SPT $d(i)$ è il costo di un cammino da $r$ a $i$ in $G$, oppure $d(i)=M$, $\forall i \in N$

dim

Per ind. sul numero di iterazioni

Al termine della prima iterazione




$d(j) = c_{rj}$ se $(r,j) \in FS(r)$

$d(j) = M$ altrimenti

P. Induttivo

Supp. che la proprietà sia vera al termine dell'it $K$ e consideriamo l'iterazione $K+1$

$i \xrightarrow{c_{ij}} j$, FS(i)

1) $d(j)$ non migliora $\rightarrow$ vero per hp. induttiva

2) $d(j)$ migliora cioè $d(j) = d(i) + c_{ij}$

per hp ind. $d(i)$ è il costo di un cammino $P_i$ da $r$ a $i$ in $G$:

$$r \underset{d(i)}{\overset{P_i}{\rightsquigarrow}} i \xrightarrow{c_{ij}} j$$

Ovvero $d(j) = d(i) + c_{ij}$ è il costo del cammino ottenuto concatenando $P_i$ e $(i,j)$ $\square$

## Teorema 4 (Terminazione SPT)

SPT esegue un numero finito di iterazioni

dim

Poiché

1) Poiché per hp $G$ non ha cicli negativi, i cammini individuati da SPT sono semplici

2) il numero dei cammini semplici da $r$ a $i$ in $G$ è finito $\forall i \in N$

3) un nodo $i$ è inserito in $Q$ solo se $d(i)$ diminuisce

4) Per il Teo 3 $d(i)$ è il costo di un cammino da $r$ a $i$ in $G$ (oppure $d(i) = M$)

segue che:

1) $d(i)$ può diminuire solo un numero finito di volte

2) un nodo $i$ può essere inserito in $Q$ solo un numero finito di volte

Dopo un numero finito di iterazioni quindi $Q = \emptyset \Rightarrow$ SPT termina.

SPT =: famiglia di algoritmi
(SELECT)

- **SPT-S**: Q è una coda di priorità. Si estrae $i$ con minima etichetta
- **SPT-L**: Q è una lista. La selezione di $i$ dipende dalla sua posizione nella lista

## SPT-S (shortest first)

## Teorema 5 (Dijkstra)

(estrae da)

Se $c_{ij} \geq 0$ $\forall (i,j) \in A$, SPT-S inserisce in $Q$ ogni nodo $i$ al più una volta

## Lemma 2 (di supporto)

notazione

$i_K$ = nodo estratto da $Q$ all'inizio della $K$-esima it.

$d_K(i)$ =: etichetta di $i$ all'inizio della $K$-esima it. $\forall i \in N$

$i_1 = r$, $d_1(r) = 0$, $d_1(i) = r$ $i \neq r$

Se $c_{ij} \geq 0$ $\forall (i,j) \in A$: $d_{K+1}(i_{K+1}) \geq d_K(i_K)$ $\forall K \geq 1$

ovvero la sequenza delle etichette dei nodi estratti da $Q$ in SPT-S è non decrescente dim

$\forall K \geq 1$ si possono verificare 2 casi

1) $d_{K+1}(i_{K+1}) = d_K(i_{K+1})$ ovvero l'etichetta di $i_{K+1}$ non cambia

$\Rightarrow i_{K+1} \in Q$ all'inizio della K-esima iterazione

$\Rightarrow d_{K+1}(i_{K+1}) = d_K(i_{K+1}) \geq d_K(i_K)$ essendo $i_K$ un nodo con min. etichetta all'it. K.

2) $d_{K+1}(i_{K+1}) < d_K(i_{K+1})$, ovvero l'etichetta di $i_{K+1}$ è diminuita durante l'it. K.

$\Rightarrow$ $i_K \xrightarrow{c_{i_K, i_{K+1}}} i_{K+1}$ $\quad d_{K+1}(i_{K+1}) = d_K(i_K) + \underbrace{c_{i_K, i_{K+1}}}_{\geq 0} \geq d_K(i_K)$

dim Teo 5

Dal lemma precedente la sequenza delle etichette dei nodi estratti da Q è non decrescente. Poiché un nodo è inserito in Q solo quando la sua etichetta diminuisce, un nodo può essere inserito in Q al più una volta

## Corollario

Se $c_{ij} \geq 0$ $\forall (i,j) \in A$ SPT-S esegue $O(n)$ iterazioni

nel caso di costi negativi SPT-S può essere esponenziale

## Algoritmo Dijkstra

Q lista non ordinata

Q vettore di puntatore ai nodi Q: [ ]

- inserzione in Q $\rightarrow O(1)$ per it.

  Q: [ ] ricerca della min etichetta

- estrazione da Q (SELECT) $\rightarrow O(n)$ per it.

- analisi delle stelle uscenti $\rightarrow O(m)$

Se $c_{ij} \geq 0$ l'alg esegue $O(n)$ it. quindi $O(n^2 m)$ cioè $\boxed{O(n^2)}$

complessità in tempo

# SPT - L

## Algoritmo di Bellman

Q = coda (FIFO) Q: [ ← inserzione $O(1)$ (se non già presente in Q)

estrazione $O(1)$

analisi delle stelle uscenti $O(mn)$

1) Se Q è una coda ogni nodo è inserito in Q $\Rightarrow$ estratto da Q al più n-1 volte

2) Ogni arco è esaminato al più n-1 volte

Quindi $O(mn + n^2)$ cioè $\boxed{O(mn)}$

complessità in tempo

3) L'alg permette di rilevare la presenza di cicli negativi in G, controllando se un nodo viene inserito in Q almeno n volte

Quale alg. è più conveniente?

- Dijkstra se i costi sono positivi
- Bellman se i costi sono arbitrari

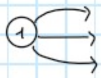
# Cammini min su reti acicliche

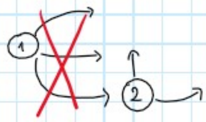
- Un grafo e' aciclico $\Leftrightarrow$ $\overbrace{\text{e' possibile enumerare i suoi nodi in modo che } (i,j) \in A \Rightarrow i < j}^{\text{grafo ben enumerato}}$
- Intuitivamente non essendoci cicli, posso numerare in modo crescente i nodi lungo qualsiasi P
- Se G è aciclico (e ben enumerato) il problema dei c.m. si risolve in $O(m)$ esaminando i nodi in ordine crescente

```
Procedura Acyclic (G,p,d)

begin
p(1)=nil;
d(1)=0;
for  i=2  to n   do
        begin
        p(i)=1;
        d(i)=M;
        end
for  i=1  to n-1  do
        for each (i,j) ∈ FS(i)   do
                if  d(i)+c_ij <d(j)  then
                        begin
                        d(j)=d(i)+c_ij ;
                        p(j)=i;
                        end
end
```

## Oss

- Se il grafo è aciclico ma non è ben enumerato, posso ben enumerarlo con VISITA in $O(m)$
- Se G è aciclico, $\exists$ i nodo con $BS(i) = \emptyset$ $\Rightarrow$ $i = 1$ 
- $\exists$ almeno un nodo $i$, nel sottografo ottenuto da G rimuovendo 1 con $BS(i) = \emptyset$ (perchè il sottografo è aciclico $\Rightarrow$ $i = 2$



# IL PROBLEMA DI FLUSSO MASSIMO

## Setting

Dati:
$G = (N, A)$ grafo orientato
$s \in N$ nodo sorgente
$t \in N$ nodo pozzo
$u_{ij} > 0 \quad \forall (i,j) \in A$ : capacità di $(i,j)$

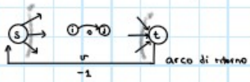
Il problema di flusso massimo consiste nell'inviare la massima quantità di flusso da s a t in G, rispettando le capacità degli archi

## Modello Matematico

V. decisionali : $x_{ij} :=$ flusso lungo $(i,j) \quad \forall (i,j) \in A$
$v :=$ valore del flusso

$$\max \ v$$
$$\sum_{(j,s) \in BS(s)} x_{js} - \sum_{(s,j) \in FS(s)} x_{sj} = -v$$
$$\sum_{(j,t) \in BS(t)} x_{jt} - \sum_{(t,j) \in FS(t)} x_{tj} = +v$$
$$\sum_{(j,i) \in BS(i)} x_{ji} - \sum_{(i,j) \in FS(i)} x_{ij} = 0 \quad i \neq s,t$$
$$0 \leq x_{ij} \leq u_{ij} \quad \forall (i,j) \in A$$

Si tratta di un caso speciale del problema di flusso di costo minimo




## Definizioni

- Un cammino P da s a t in G, non necessariamente orientato, e' un cammino aumentante se:
  1) $x_{ij} < u_{ij} \quad \forall (i,j) \in P^+$ archi concordi non saturi
  2) $x_{ij} > 0 \quad \forall (i,j) \in P^-$ archi discordi non vuoti

- La capacita' di un cammino aumentante P rispetto a un flusso x è

$$\theta(P, x) = \min \{ \min \{ u_{ij} - x_{ij} \text{ t.c. } (i,j) \in P^+ \}, \min \{ x_{ij} \text{ t.c. } (i,j) \in P^- \} \} > 0$$

Rappresenta la massima quantità di flusso inviabile lungo P senza violare la capacità superiore $u_{ij}$ degli archi concordi e la capacità inferiore (0) degli archi discordi

## Come determinare se $\exists$ un cammino aumentate rispetto a un flusso $x$?

$G_x = (N, A_x)$ dove $A_x = \{ (i,j) \in A \text{ t.c. } x_{ij} < u_{ij} \} \cup \{ (i,j) \text{ t.c. } (j,i) \in A \ x_{ji} > 0 \}$

$\hookrightarrow$ grafo residuo rispetto a x (contiene tutti e soli gli archi utilizzabili in un cammino aumentante rispetto a x, orientati da s a t)

## Proprieta'

I cammini aumentanti da s a t sono in corr. biunivoca con i cammini orientati da s a t in $G_x$
Quindi: determinare se $\exists$ un cammino aumentante da s a t e' equivalente a determinare se t e' raggiungibile da s in $G_x$ (Visita in $O(m)$)

**cond. necessaria per l'ottimalita' di un flusso**

x: non esistono cammini aumentanti da s a t

- Un taglio $(N_s, N_t)$ e' una partizione di $N$ tale che $s \in N_s, t \in N_t$
$\boxed{\begin{matrix} N_s \cup N_t = N \\ N_s \cap N_t = \phi \end{matrix}}$

- Dato un taglio $(N_s, N_t)$ : $A^+(N_s, N_t) = \{(i,j) \in A \text{ t.c. } i \in N_s, j \in N_t\} =:$ archi diretti del taglio
$A^-(N_s, N_t) = \{(i,j) \in A \text{ t.c. } i \in N_t, j \in N_s\} =:$ archi inversi del taglio

## Oss

1) $\boxed{x_{ij} = u_{ij}}$ $\forall (i,j) \in A^+(N_s, N_t)$ ovvero tutti gli archi diretti del taglio sono saturi

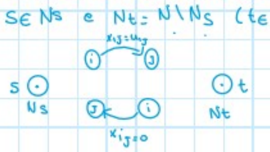
$x_{ij} = 0$ $\forall (i,j) \in A^-(N_s, N_t)$ ovvero tutti gli archi inversi del taglio hanno flusso nullo

## Proprietà

Dato un flusso ammissibile $x$, se $\nexists$ cammini aumentanti da s a t $\Rightarrow \exists$ un taglio $(N_s, N_t)$ tale che:

1) $x_{ij} = u_{ij}$ $\forall (i,j) \in A^+(N_s, N_t)$
2) $x_{ij} = 0$ $\forall (i,j) \in A^-(N_s, N_t)$

dim
Sia $N_s$ l'insieme dei nodi raggiunti da $s$ durante la ricerca di un cammino aumentante in $G_x$
$s \in N_s$ e $N_t = N \setminus N_s$ $(t \in N_t)$




$\forall (i,j) \in A^+(N_s, N_t)$ si ha che $x_{ij} = u_{ij}$, altrimenti $j$ sarebbe raggiungibile da s
$\forall (i,j) \in A^-(N_s, N_t)$ si ha che $x_{ij} = 0$, altrimenti $i$ sarebbe raggiungibile da s

- Dato un taglio $(N_s, N_t)$ e un flusso ammissibile $x$:

$$x(N_s, N_t) = \sum_{(i,j) \in A^+(N_s,N_t)} x_{ij} - \sum_{(i,j) \in A^-(N_s,N_t)} x_{ij}$$ = flusso che attraversa il taglio

$$u(N_s, N_t) = \sum_{(i,j) \in A^+(N_s,N_t)} u_{ij}$$ =: capacità del taglio

## Teorema 1

Per ogni flusso ammissibile $x$ di valore $v$ e per ogni taglio $(N_s, N_t)$ si ha che $v = x(N_s, N_t) \leq u(N_s, N_t)$

dim

$$x(N_s, N_t) = \sum_{(i,j) \in A^+(N_s,N_t)} x_{ij} - \sum_{(i,j) \in A^-(N_s,N_t)} x_{ij} \leq \sum_{(i,j) \in A^+(N_s,N_t)} u_{ij} = u(N_s, N_t)$$

Sommo membro a membro i vincoli di conservazione di flusso relativi a $i \in N_t$

$$\sum_{i \in N_t} \Big( \sum_{(j,i) \in BS(i)} x_{ji} - \sum_{(i,j) \in FS(i)} x_{ij} \Big) = +v$$

$$\sum_{i \in N_t} \Big( \sum_{\substack{(j,i) \in BS(i) \\ j \in N_s}} x_{ji} + \sum_{\substack{(j,i) \in BS(i) \\ j \in N_t}} x_{ji} - \sum_{\substack{(i,j) \in FS(i) \\ j \in N_t}} x_{ij} - \sum_{\substack{(i,j) \in FS(i) \\ j \in N_s}} x_{ij} \Big) = +v$$

$$\underbrace{\sum_{\substack{(j,i) \in A \\ i \in N_t \\ j \in N_s}} x_{ji}}_{A^+(N_s,N_t)} + \cancel{\sum_{\substack{(j,i) \in A \\ i \in N_t \\ j \in N_t}} x_{ji}} - \cancel{\sum_{\substack{(i,j) \in A \\ i \in N_t \\ j \in N_t}} x_{ij}} - \underbrace{\sum_{\substack{(i,j) \in A \\ i \in N_t \\ j \in N_s}} x_{ij}}_{A^-(N_s,N_t)} = v$$

$\Rightarrow x(N_s, N_t) = +v$

## Teorema 2 "Condizioni di ottimalita'"

Un flusso ammissibile $x$ e' massimo $\iff \nexists$ cammini aumentanti

dim

$\Rightarrow$ gia' visto

$\Leftarrow$ Per la proprieta' prec. $\exists$ un taglio $(N_s, N_t)$ tale che
1) $x_{ij} = u_{ij} \quad \forall (i,j) \in A^+(N_s, N_t)$
2) $x_{ij} = 0 \quad \forall (i,j) \in A^-(N_s, N_t)$

Per Teo 1 $v = x(N_s, N_t) = u(N_s, N_t)$

$\Rightarrow$ Non e' possibile spedire altro flusso da s a t, perche' la capacita' di $(N_s, N_t)$ sarebbe violata
Quindi $x$ e' un flusso massimo

## Corollario

Un taglio $(N_s, N_t)$ tale che $x_{ij} = u_{ij} \ \forall (i,j) \in A^+(N_s, N_t)$ e $x_{ij} = 0 \ \forall (i,j) \in A^-(N_s, N_t)$ e' un taglio di capacita' minima (taglio minimo)

## Teorema 3 "Flusso max - taglio minimo"

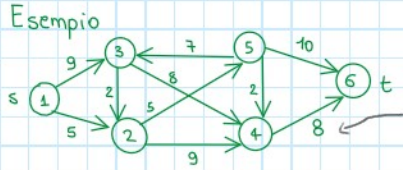
Il valore del massimo flusso da s a t e' uguale alla minima capacita' dei tagli $(N_s, N_t)$

dim

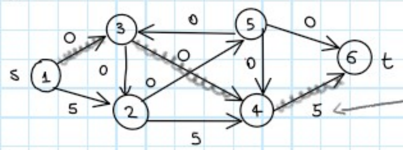
Segue dai risultati precedenti

Dato un flusso ammissibile $x$ di valore $v$, come decidere se $x$ è ottimo?
come migliorarlo se non lo è?

### Esempio



x:



$v = 5$

la sol è amm in quanto in ogni nodo di transito entra quanto esce

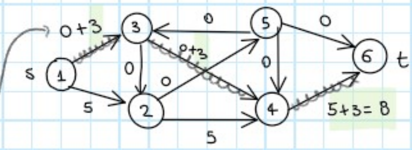
1) Inviamo il flusso lungo il cammino $P = (1\ 3\ 4\ 6)$

quantita' di flusso inviabile $\theta(P,x) = \min\{\underbrace{9-0,\ 8-0\ \ 8-5}_{u_{ij}-x_{ij}}\} = 3$

$\theta(P,x) > 0$ in quanto $u_{ij} - x_{ij} > 0 \quad \forall (i,j) \in A$ ovvero $x_{ij} < u_{ij} \ \forall (i,j) \in P$ archi non saturi

Aggiorniamo $x$ inviando $\theta(P,x)$ unita' di flusso lungo P: $x = x(\theta) = x \oplus \theta P$, con $\theta = \theta(P,x) = 3$
oper. di composizione tra x e P

$\Rightarrow$ la sol non è ottima ma migliorabile

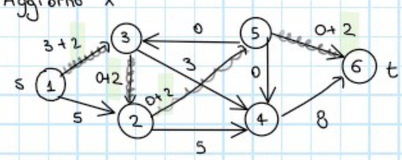


$v = 5 + \underset{\theta}{\underset{\shortparallel}{3}} = 8$

2) Inviamo il flusso lungo il c ino $P = (1\ 3\ 2\ 5\ 6)$

$\theta = \theta(P,x) = \min\{\underset{\text{init}}{9} - \underset{\text{quello prec.}}{3},\ 2-0,\ 5-0,\ 10-0\} = 2$

Aggiorno $\bar{x}$



$v = 8 + 2 = 10$

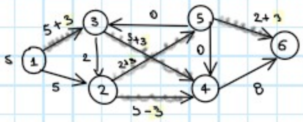
la sol non è ottima ma migliorabile

3) Inviamo il flusso lungo il cammino non orientato $P = (1\ 3\ 4\ 2\ 5\ 6)$
in cui $P^+ = \{(1,3), (3,4), (2,5), (5,6)\}$ archi concordi
$P^- = \{(2,4)\}$ archi discordi

3) Inviamo il flusso lungo il cammino non orientato $P = (1\ 3\ 4\ 2\ 5\ 6)$
in cui $P^+ = \{(1,3), (3,4), (2,5), (5,6)\}$ archi concordi
$P^- = \{(2,4)\}$ archi discordi

Lungo gli archi concordi il flusso è incrementato
Lungo gli archi discordi è decrementato

$\theta(P,x) = \min\{ \min\{9-5, 8-3, 5-2, 10-2\}, \min\{5\} \} = 3$

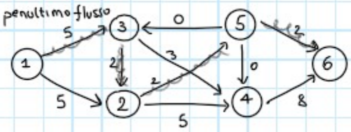



$v = 10 + 3 = 13$

## Esempio $G_x$

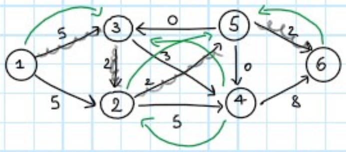
penultimo flusso

x:




$v = 10$

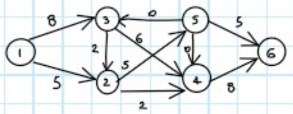
$G_x =$




$(1\ 3\ 4\ 2\ 5\ 6)$
e' ora rappresentato in $G_x$
come un cammino orientato da s a t

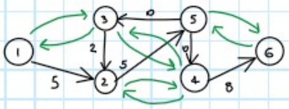
## Es cond. necessaria per l'ottimalità

X (ultimo flusso)

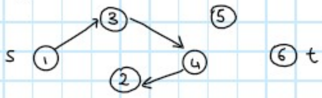



$v = 13$

$G_x$




eseguo la procedura visita a partire da s in $G_x$




t non è ragg. da s in $G_x$
$\Rightarrow \nexists$ cammini aumentanti rispd x

## esempio di $N_s$, $N_t$, $A^+$, $A^-$

$N_s = \{1,2,3,4\} =:$ insieme dei nodi ragg. da s in $G_x$
$N_t = N \setminus N_s = \{5,6\} =:$ insieme dei nodi non ragg. da s in $G_x$

$A^+(N_s, N_t) = \{(2,5), (4,6)\}$
$(1,5), (1,6), (2,6), (3,5), (3,6), (4,5) \notin A \quad i \in N_s,\ j \in N_t$
$A^-(N_s, N_t) = \{(5,4), (5,3)\}$

## es flusso max taglio min

$v = 13 = x(N_s, N_t) = \underset{u_{2,5}}{5} + \underset{u_{4,6}}{8} = u(N_s, N_t)$

# Legame con la Teoria della dualità PL

max v

$$\sum_{(j,i) \in BS(i)} x_{ji} - \sum_{(i,j) \in FS(i)} x_{ij} = \begin{cases} -v & \text{se } i = s \\ +v & \text{se } i = t \\ 0 & \text{altrimenti} \end{cases} \quad \forall i \in N$$

$0 \le x_{ij} \le u_{ij} \quad \forall (i,j) \in A$
v. duali

Matrice dei vincoli

$$\begin{array}{c|cc|c}
 & x_{ij} & v & \\
s & & -1 & \pi_s \\
i & 1 & & \pi_i \\
j & -1 & & \pi_j \\
t & & 1 & \pi_t \\
 & 1 & & \mu_{ij}
\end{array} \quad \begin{array}{l} \\ \\ = 0 \\ \\ \\ \le u_{ij} \end{array}$$

DUALE PL

$$\left.\begin{array}{l} \min \sum u_{ij} \mu_{ij} \\ \pi_i - \pi_j + \mu_{ij} \ge 0 \\ \pi_t - \pi_s = 1 \\ \mu_{ij} \ge 0 \end{array}\right\} \text{Prob. del taglio min.}$$

Ogni taglio $(N_s, N_t)$ e' una sol ammissibile di (D), e il valore della f. obiettivo duale e' $u(N_s, N_t)$

$\pi_i = \begin{cases} 0 & i \in N_s \\ 1 & i \in N_t \end{cases} \qquad \mu_{ij} = \begin{cases} 1 & \text{se } (i,j) \in A^+(N_s, N_t) \\ 0 & \text{else} \end{cases}$

Il risultato $v \le u(N_s, N_t)$ segue dal Teo debole della dualità
Teo flusso max - taglio min è la specializzazione del Teo forte.

## Schema algoritmico basato sui cammini aumentanti

Procedura CAMMINI AUMENTANTI $(G,u,s,t,x,\underbrace{Ns,Nt}_{\text{output}})$

**begin**
x=0;     %si può inizializzare da qualsiasi flusso ammissibile%
**while** TROVA CAMMINO $(G,u,s,t,x,\underbrace{P,\theta}_{\text{output}},Ns,Nt)$ **do**
    AUMENTA FLUSSO$(X,P,\theta)$
**end**

$X = X \oplus \theta P$

## Teorema

Se $u_{ij} \in \mathbb{Z}^+$ $\forall (i,j) \in A$ allora CAMMINI AUMENTANTI individua un flusso max intero

dim

Ad ogni iter. $\theta = \theta(P,x)$ e' un intero positivo e $x$ e' un flusso intero

$$\theta(P,x) = \min\left\{ \min\{u_{ij} - x_{ij} \text{ tc } (i,j) \in P^+\}, \min\{x_{ij} \text{ tc. } (i,j) \in P^-\} \right\}$$

e' vero alla prima iter. $u_{ij} \in \mathbb{Z}^+$ $\forall (i,j) \in A$
$x_{ij} = 0$ $\forall (i,j) \in A$

$X = X \oplus \theta P$ e' quindi intero

Per ind. se la proprietà vale alla K it. vale anche alla K+1
Quindi vale $\forall$ iter.

Poiche' $\theta \geq 1$ CAMMINI AUMENTANTI esegue un numero finito di iter. e il flusso finale (max) e' intero

CAMMINI AUMENTANTI

Alg. di Edmonds e Karp.

TROVA CAMMINO: si esegue visita in Gx, a partire da s, usando un criterio di selezione dei nodi da Q di tipo FIFO, ovvero Q è una coda.

Quindi: ad ogni iterazione (tranne l'ultima è individuato un cammino aumentante con il minimo numero di archi)

Complessità in tempo :

- numero di iterazioni: O(mn)
- TROVA CAMMINO : O(m) per it.
- AUMENTA FLUSSO :O(n) per it.

Quindi $O(m^2 n)$

# IL PROBLEMA DI FLUSSO DI COSTO MINIMO

$G = (N, A)$ rete di flusso
$c_{ij}$ $\forall (i,j) \in A$ costo di $(i,j)$
$u_{ij} \geq 0$ $\forall (i,j) \in A$ capacità superiore di $(i,j)$
$b_i$ $\forall i \in N$ bilancio del nodo $i$

$$\boxed{\begin{aligned} &\min \sum_{(i,j)\in A} c_{ij} x_{ij} \\ &\sum_{(j,i)\in BS(i)} x_{ji} - \sum_{(i,j)\in FS(i)} x_{ij} = b_i \quad i \in N \\ &0 \leq x_{ij} \leq u_{ij} \quad (i,j) \in A \end{aligned}}$$

Studiamo condizioni di ottimalità per il problema:
dato un flusso ammissibile x, è ottimo?(ovvero è di costo minimo?)

Sia $G_x = (N, A_x)$ il grafo residuo rispetto a $x$, def come il prob. di flusso max

$\forall (i,j) \in A$ tc $x_{ij} < u_{ij} \Rightarrow (i,j) \in A_x$ $\quad c'_{ij} = c_{ij}$
$\forall (i,j) \in A$ tc $x_{ij} > 0 \Rightarrow (j,i) \in A_x$ $\quad c'_{ji} = -c_{ij}$

$G_x$ è ora pesato in quanto ai suoi archi è associato un costo di percorrenza

Un ciclo aumentante è un ciclo orientato in $G_x$
La max quantità di flusso inviabile lungo un ciclo aumentante $C$ è

$$\theta(C,x) = \min \{ (u_{ij} - x_{ij}) \text{ tc. } (i,j) \in C^+, \ x_{ij} \text{ tc. } (i,j) \in C^- \}$$

con $C^+$ = archi diretti del ciclo (in $G$)
$C^-$ = archi inversi del ciclo (in $G$)
considerando l'orientamento dato dal verso di percorrenza in $G_x$

## Proprietà

Se $x$ è un flusso ammissibile e $C$ è un ciclo in $G_x$ (aumentante), $x(\theta) = x \oplus \theta C$ con $0 \leq \theta \leq \theta(C,x)$ è ancora un flusso ammissibile

Quindi se $c(C) < 0$ ovvero $C$ è un ciclo aumentante negativo $\Rightarrow cx(\theta) < cx \Rightarrow x$ non è un flusso di costo min.

**Cond necessaria per l'ottimalità di un flusso ammissibile x**
Non devono esistere cicli aumentanti negativi in $G_x$

La cond è anche sufficiente?

## Teo: "decomposizione di flussi"

Siano $x$ e $x'$ flussi amm. in $G$
Allora $\exists$ $K$ cicli aumentanti in $G_x$ $C_1, \dots, C_K$ con $K \leq m$ tali che $x' = x \oplus \theta_1 C_1 \oplus \theta_2 C_2 \oplus \dots \oplus \theta_K C_K$

con $0 < \theta_i \leq \theta(C_i, x)$ $i = 1, \dots, K$

Ovvero: un qualsiasi flusso ammissibile x' è ottenibile da un qualsiasi altro flusso ammissibile x inviando flusso lungo al più m cicli aumentanti rispetto a x

dim

Sia $A_x^+ = \{ (i,j) \text{ tc. } x'_{ij} > x_{ij} \}$ $\quad d_x(i,j) = x'_{ij} - x_{ij}$ $\quad$ se $i \xrightarrow[x'_{ij}=7]{x_{ij}=4} j$ $\quad$ in $A_x^+$ $\quad i \xrightarrow{d_x(i,j) = 7-4=3} j$

$A_x^- = \{ (j,i) \text{ tc. } x'_{ij} < x_{ij} \}$ $\quad d_x(j,i) = \underbrace{x_{ij} - x'_{ij}}_{>0}$ $\quad$ se $i \xrightarrow[x'_{ij}=2]{x_{ij}=3} j$ $\quad$ in $A_x^-$ $\quad i \xleftarrow{d_x(j,i) = 3-2=1} j$

Vale una proprietà: se un nodo in $\bar{G}_x$ ha un arco entrante, deve avere almeno un arco uscente

Quindi (procedura per convertire $x$ in $x'$)

1) Scelgo un nodo $i$ avente almeno un arco uscente in $\bar{G}_x$
(se non esistono $\Rightarrow$ STOP $x = x'$)

2) Visito $\bar{G}_x$ a partire da $i$ fino a visitare un nodo già visitato: ciclo aumentante rispetto a $x$ $(C)$

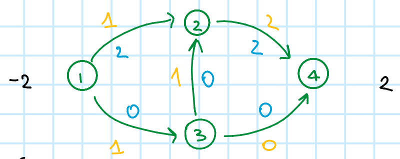
3) Invio la quantità di flusso $\theta = \min\{d_x(i,j) \text{ tc } (i,j)\in C\}$ lungo C

4) elimino da $\bar{G}_x$ ogni arco $(i,j)$ tc $x_{ij} = x'_{ij}$ dopo l'invio (almeno uno)

5) aggiorno le capacità $d_x(i,j)$ e itero finché x è stato convertito in x'

$O(m)$ invii lungo cicli aumentanti rispetto a x !

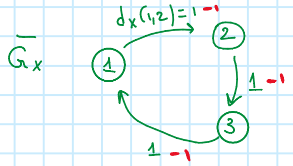
6) Ho eliminato ogni arco da $\bar{G}_x$




$u_{ij} = 3$

$x'$, $x$

Convertiamo x in x'




• invio $\theta = \min\{1,1,1\} = 1$ lungo C

Così facendo converto x in x' → STOP
rimuovo i 3 archi da $\bar{G}_x$

## Teorema 2

Un flusso ammissibile x è di costo minimo $\iff$ ∄ cicli aumentanti negativi rispetto a x.

dim

$\Rightarrow$ già visto

$\Leftarrow$ (sufficienza).

Supponiamo ∄ cicli aumentanti rispetto a x

Sia x' un qualsiasi altro flusso ammissibile in G

Per il Teo di decomposizione di flussi $x' = x \oplus \theta_1 C_1 \oplus \dots \oplus \theta_k C_k$

Quindi $cx' = cx + \underbrace{\theta_1 c(C_1)}_{\geq 0} + \dots + \underbrace{\theta_k c(C_k)}_{\geq 0}$ per hp. (cicli non neg)

Quindi $cx' \geq cx$ cioè x è un flusso di costo minimo

ALGORITMO BASATO SU CANCELLAZIONE DI CICLI

- Determina se esiste, un flusso ammissibile x
- Finché esistono cicli negativi:
  - Determina un ciclo negativo C in Gx
  - Sia $\theta=\theta(x)$ massima quantità di flusso inviabile lungo C
  - Aggiorna $x := x \oplus \theta C$

### Dettagli implementativi

A livello di svolgimento degli esercizi, i cicli negativi possono essere cercati per ispezione, salvo che all'ultima iterazione

I cicli negativi possono essere cercati direttamente in G, come per il problema di flusso massimo

Come determinare se esiste un flusso ammissibile iniziale? Mediante un algoritmo di flusso massimo