Il problema del flusso massimo su reti

Introduzione

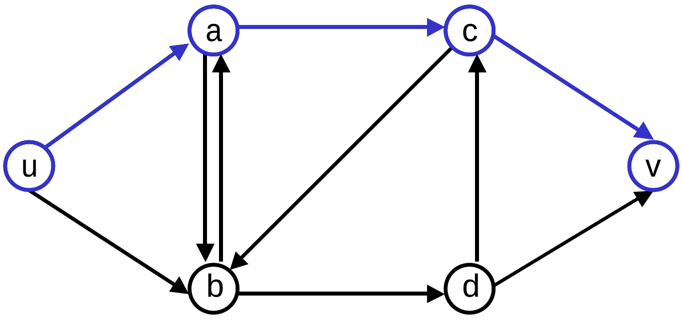
Modelli di flusso hanno applicazioni:

- Nelle telecomunicazioni
- In informatica (multiprocessori, protocolli internet)
- Nei trasporti (aereo, stradale, ferroviario, merci)

Cammino orientato

- Un grafo orientato è una coppia G=(N,A), in cui
  - N è un insieme finito di nodi
  - A è una famiglia di coppie ordinate di nodi, $A \subseteq N \times N$ (a=(i,j)), dette archi
- Un cammino orientato (da v_1 a v_{k+1}) è una sequenza, $a_1=(v_1, v_2)$, $a_2=(v_2, v_3)$,..., $a_k=(v_k, v_{k+1})$, di archi consecutivi

Esempio di cammino



Il cammino da u a v, ovvero (u,a), (a,c), (c,v)

Taglio orientato

Dato $S \subset N$, un taglio orientato è un sottoinsieme di archi del tipo

– $\delta^+_G(S) := \{(i,j) \in A: i \in S, j \notin S\}$

archi uscenti da S

oppure del tipo

– $\delta^-_G(S) := \{(i,j) \in A: i \notin S, j \in S\}$

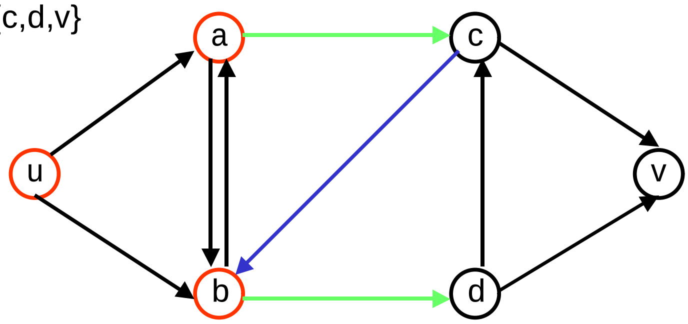
archi entranti in S

- Useremo $\delta^+(v)=\delta^+_G(\{v\})$ e $\delta^-(v)=\delta^-_G(\{v\})$

Esempio di taglio orientato

S ={u,a,b}

N\S={c,d,v}

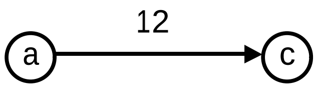


$\delta^+_G(S) := \{(a,c), (b,d)\}$

$\delta^-_G(S) := \{(c,b)\}$

Rete di flusso

•Una rete di flusso è un grafo orientato G=(N,A) in cui ad ogni arco (i,j) $\in$ A è associata una capacità $k_{ij} \geq 0$ es:

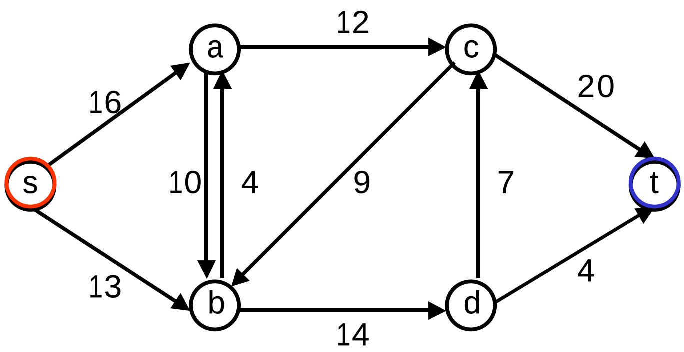


- Vengono inoltre specificati due vertici s e t, detti sorgente e destinazione
- Tipicamente (ma non sempre) si ha

$$\delta^-(s) = \delta^+(t) = 0$$

(non ci sono archi entranti in s e uscenti da t)

Esempio di rete di flusso



Flusso

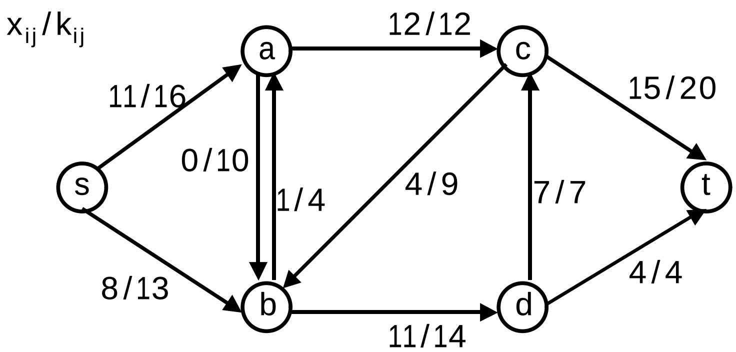
- Data una rete di flusso, si dice flusso ammissibile da s a t una funzione
  x: $A \rightarrow \Re$ tale che:
  – per ogni arco $(i,j) \in A \quad 0 \leq x_{ij} \leq k_{ij}$
  – $\sum_{(h,j) \in \delta^+(h)} x_{hj} - \sum_{(j,h) \in \delta^-(h)} x_{jh} = 0$, per ogni $h \in N \setminus \{s,t\}$

Flusso uscente da h

Flusso entrante in h

Conservazione del flusso

Esempio di flusso ammissibile



Il problema del flusso massimo

- Il problema di ottimizzazione su rete più comune consiste nell'inviare il massimo flusso (Max flow) da s a t,
- $\max\{\varphi = \sum_{(s,j) \in \delta^+(s)} x_{sj} - \sum_{(j,s) \in \delta^-(s)} x_{js}$:
  x flusso ammissibile$\}$
- $\varphi_0 = \varphi(s)$ si dice valore del flusso

Formulazione matematica

- Sia G=(N,A) la rete dove ad ogni arco $(i,j) \in A$ è associata una capacità $k_{ij} \geq 0$ e (s,t) siano la coppia sorgente-destinazione
- Variabili decisionali:
  X_{ij}= flusso che attraversa l'arco $(i,j) \in A$
- Obbiettivo: massimizzare il valore del flusso φ da s a t

Formulazione del problema del flusso massimo

$$\text{Max } \varphi$$

Soggetto a

$$\sum_{\forall (i,j) \in \delta^+(i)} X_{ij} - \sum_{\forall (j,i) \in \delta^-(i)} X_{ji} = \varphi \text{, per } i=s$$

$$\sum_{\forall (i,j) \in \delta^+(i)} X_{ij} - \sum_{\forall (j,i) \in \delta^-(i)} X_{ji} = 0 \text{, per } i \in N \backslash \{s,t\}$$

Conservazione del flusso

$$\sum_{\forall (i,j) \in \delta^+(i)} X_{ij} - \sum_{\forall (j,i) \in \delta^-(i)} X_{ji} = -\varphi \text{, per } i=t$$

$$0 \leq X_{ij} \leq K_{ij} \text{, per } (i,j) \in A$$ ammissibilità

Formulazione matematica

- Esempio di modello di programmazione lineare in quanto sia la funzione obbiettivo che i vincoli sono funzioni lineari

Sezione

- Si dice sezione una partizione (S,N\S) dell'insieme dei nodi tale che $s \in S$ e $t \in N \setminus S$
- Dato un flusso ammissibile x, si dice flusso attraverso una sezione (S,N\S) la quantità

$$\varphi(S) = \underbrace{\sum_{(i,j) \in \delta^+(S)} x_{ij}}_{\text{Flusso uscente da S}} - \underbrace{\sum_{(j,i) \in \delta^-(S)} x_{ji}}_{\text{Flusso entrante in S}}$$

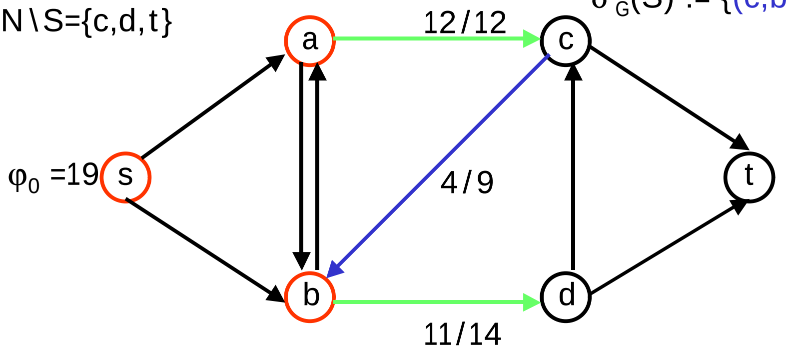
- Si dice capacità della sezione la quantità

$$k(S) = \sum_{(i,j) \in \delta^+(S)} k_{ij}$$

Esempio di Sezione

S ={s,a,b}
N\S={c,d,t}

$\delta^+_G(S)$:= {(a,c), (b,d)}
$\delta^-_G(S)$:= {(c,b)}



Flusso attraverso S è 12+ 11 -4 = 19
Capacità della sezione S è 12+ 14 = 26

Enunciato di due primi risultati

- Teorema 1: Sia x un flusso ammissibile. Per ogni sezione (S, N\S) si ha che $\varphi_0 = \varphi(S)$

  (il flusso attraverso ogni sezione è costante)
- Teorema 2: Per ogni flusso ammissibile x e per ogni sezione (S, N\S) si ha che $\varphi(S) \leq k(S)$

Osservazione: il valore del massimo flusso non può superare la capacità della sezione minima

Dimostrazione Teorema 1

- $\varphi_0 = \varphi(s) = \sum\limits_{(s,j)\in\delta^+(s)} x_{sj} - \sum\limits_{(j,\, s)\in\delta^-(s)} x_{js}$

$$+ \sum_{h\in S}\Big[\underbrace{\sum_{(h,j)\in\delta^+(h)} x_{hj} - \sum_{(j,\, h)\in\delta^-(h)} x_{jh}}_{=0 \text{ per ogni } h\neq s}\Big] = \sum_{h\in S}\Big[\sum_{(h,j)\in\delta^+(h)} x_{hj} - \sum_{(j,\, h)\in\delta^-(h)} x_{jh}\Big]$$

$$= \Big[\sum_{(i,j)\in A(S)} x_{ij} + \sum_{(j,\, i)\in\delta^+(S)} x_{ji}\Big] - \Big[\sum_{(i,j)\in A(S)} x_{ij} + \sum_{(j,\, i)\in\delta^-(S)} x_{ji}\Big] = \varphi(S)$$

Dimostrazione Teorema 2

$$\varphi(S) = \sum_{(i,j) \in \delta^+(S)} x_{ij} - \sum_{(j,\, i) \in \delta^-(S)} x_{ji} \le \sum_{(i,j) \in \delta^+(S)} k_{ij} = k\,(S)$$

$$x_{ij} \le k_{ij}$$

Il teorema del Max-flow Min-cut

- Un flusso ammissibile x è ottimo per il problema del Max-flow se e solo se esiste una sezione $(S^*, N \setminus S^*)$ con $\varphi(S^*) = k(S^*)$. In questo caso $(S^*, N \setminus S^*)$ è una sezione di capacità minima nella rete considerata (Min-cut).

$$\varphi_0^*$$

$$\mathrm{Max}_x \{\varphi_0\} = \mathrm{Min}_S \{K(S)\}$$

Osservazione sul teorema del Max-flow Min-cut

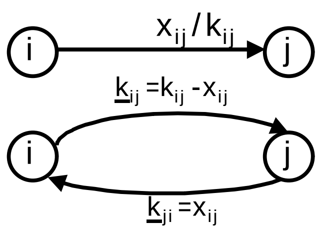
- Il problema del flusso massimo e quello della sezione minima sono fortemente legati l'uno all'altro, tanto che il valore della soluzione ottima di uno corrisponde a quello dell'altro.
- Diciamo che il problema del Min-cut è il duale del problema del Max-flow.
- E' una relazione che lega due problemi tra loro e che studieremo piu' avanti quanto parleremo di programmazione lineare
- Il teorema del Max-flow Min-cut è un risultato di dualità forte ($\varphi(S^*) = k(S^*)$), mentre il teorema 2 è un esempio di dualità debole ($\varphi(S^*) \leq k(S^*)$).

Rete incrementale

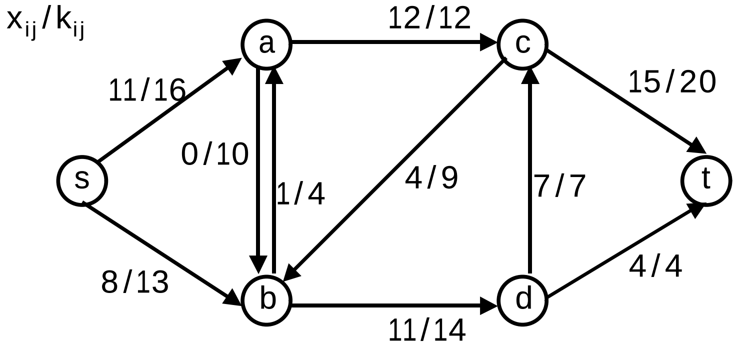
Sia x un flusso ammissibile.

- La rete incrementale $\underline{G}=(N,\underline{A})$ associata ad x è ottenuta dalla rete originale G=(N,A) sostituendo ogni arco $(i,j) \in A$ con due archi
  - un arco diretto (i,j) di capacità $\underline{k}_{ij} = k_{ij} - x_{ij} \geq 0$
  - un arco inverso (j,i) di capacità $\underline{k}_{ji} = x_{ij} \geq 0$

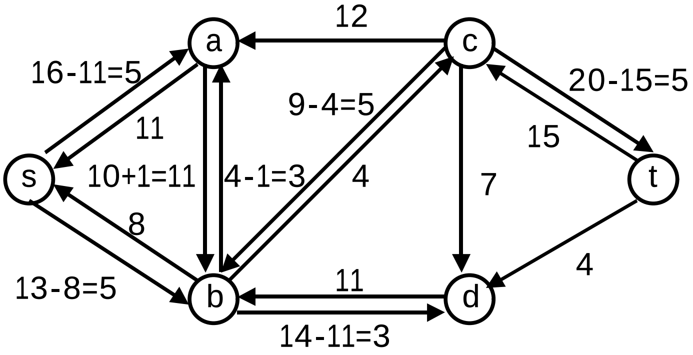
- Un arco (i,j) si dice
  - saturo se $x_{ij} = k_{ij}$
  - scarico $x_{ij} = 0$



Esempio di flusso ammissibile



Esempio di rete incrementale



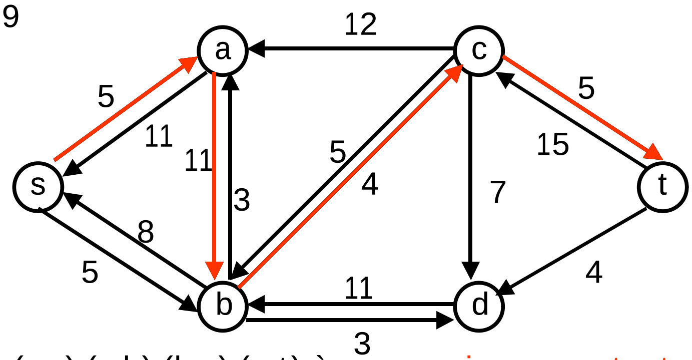
Un flusso ammissibile con valore φ_0 =19

Proprietà

- La rete incrementale offre la possibilità di aumentare il flusso sugli archi non saturi e di diminuirlo sugli archi non scarichi
- Un cammino P da s a t nella rete incrementale consente di aumentare di Δ il valore del flusso corrente (cammino aumentante)
- $\Delta = \min\{\underline{k}_{ij} : (i,j) \in P\}$

Esempio di cammino aumentante

$\varphi_0 = 19$



(s,a),(a,b),(b,c),(c,t) è un cammino aumentante

$\Delta = \min\{5,11,4,5\}=4$

Teorema 3

- Un flusso ammissibile x è ottimo per il problema del flusso massimo se e solo se il vertice t non è raggiungibile dal vertice s nella rete incrementale $\underline{G}=(N, \underline{A})$ associata ad x

Dimostrazione di 3 (I)

- Sia φ_0 il valore del flusso.
- Se t è raggiungibile da s in $\underline{G}$ allora esiste un cammino aumentante da s a t in $\underline{G}$.
- Posto $\Delta = \min\{\underline{k}_{ij}: (i,j) \in P\}$,

  per ogni $(i,j) \in P$ è possibile aggiornare
  - $x_{ij} = x_{ij} + \Delta$ se (i,j) è diretto
  - $x_{ij} = x_{ij} - \Delta$ se (i,j) è inverso
- Il nuovo vettore x è un flusso ammissibile di valore $\varphi_0 + \Delta$, il che contraddice l'ipotesi.

Dimostrazione di 3 (II)

- Supponiamo che t non sia raggiungibile da s in $\underline{G}$.
- Esiste quindi una sezione $(S^*, N \setminus S^*)$ nella rete incrementale $\underline{G}$ tale che $\delta^+_{\underline{G}}(S^*) = \emptyset$.
- Per def. di rete incrementale, si ha che in G:
  - ogni arco $(i,j) \in \delta^+_{\underline{G}}(S^*)$ è saturo
  - ogni arco $(i,j) \in \delta^-_{\underline{G}}(S^*)$ è scarico
- Allora $\varphi(S) = \underset{\text{Tutti saturi}}{\sum_{(i,j) \in \delta^+(S)} x_{ij}} - \underset{\text{Tutti scarichi}}{\sum_{(j,i) \in \delta^-(S)} x_{ii}} = \sum_{(i,j) \in \delta^+(S)} k_{ij} = k(S)$
- L'ottimalità di x deriva dal teorema 2 che garantisce anche che $(S^*, N \setminus S^*)$ sia una sezione di capacità minima nella rete G.

Considerazioni

- Il teorema 3 è dovuto a Ford-Fulkerson
- Il teorema 3 ci offrirà lo spunto per costruire un algoritmo per risolvere il problema del massimo flusso, dove
  - il flusso nullo serve per inizializzare l'algoritmo
  - il passo iterativo è dato dalla ricerca del cammino aumentante
  - il criterio di arresto dalla mancanza di cammini aumentanti in $\underline{G}$
- La ricerca del cammino aumentante caratterizza l'efficienza dell'algoritmo

Algoritmo di Ford-Fulkerson

begin

x:= 0 e φ_0 := 0 ottimo:=false (Inizializzazione)

repeat (Passo iterativo)

costruisci la rete incrementale $\underline{G}$ associata a x;

individua, se esiste, un cammino P da s a t in $\underline{G}$;

if P non esiste **then** ottimo:= true; (Condizione di ottimalità)

else

begin

$\Delta = \min\{\underline{k}_{ij}: (i,j) \in P\}$;

$\varphi_0 := \varphi_0 + \Delta$

for each $(i,j) \in P$ do

$x_{ij} = x_{ij} + \Delta$ se (i,j) è diretto

$x_{ij} = x_{ij} - \Delta$ se (i,j) è inverso

end

until ottimo=true

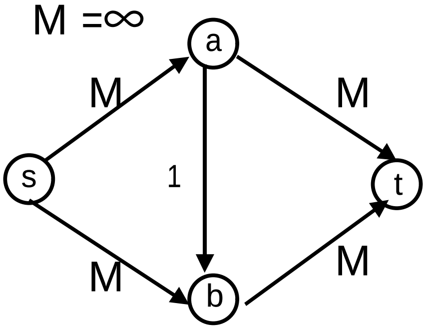
end

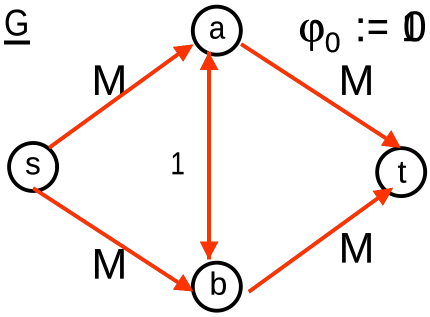
Osservazioni

- Poichè $\Delta > 0$, ad ogni iterazione del ciclo il valore del flusso φ_0 aumenta in modo monotono, il che assicura la convergenza al φ^* ottimo.
- Nel caso di capacità intere, i vettori x e $\underline{k}$ sono interi in ogni iterazione.
- Nel caso di capacità intere la complessità dell'algoritmo è $O(m^2 k_{max})$, cioè non polinomiale nella dimensione dell'istanza che è $O(m \log k_{max})$, dove
  $$k_{max} = \max \{k_{ij}: (i,j) \in A\}.$$
- Esistono situazioni in cui l'algoritmo è particolarmente inefficiente.
- Esistono tuttavia semplici modifiche che ne assicurano una complessità polinomiale.

Un esempio in cui l'algoritmo è particolarmente inefficiente

M =∞





Caso peggiore 2M iterazioni

(s,a)(a,b)(b,t) è un cammino aumentante

Δ = min{M,1,M}=1

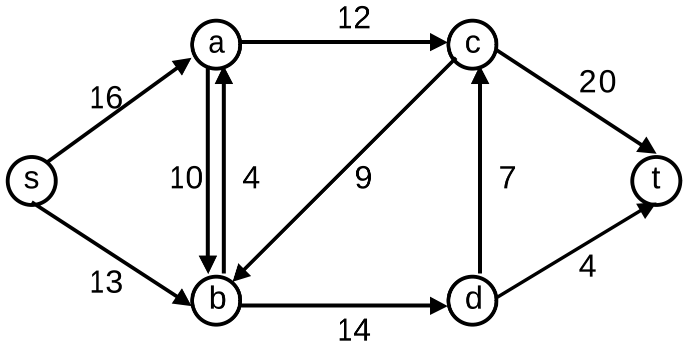
Idea: Cammino aumentante con il numero minimo di archi

Algoritmi polinomiali

- Etichette
- Cammino aumentante minimo
- Preflow-Push (potenziali)
- Scaling
- Dynamic Tree

Esempio

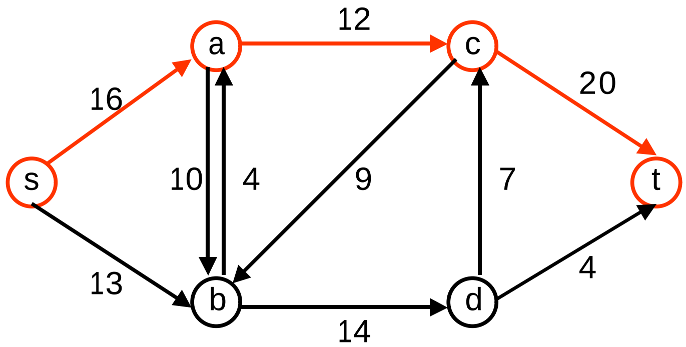
x:= 0 e φ_0 := 0 ottimo:=false



$\underline{G}$:= G

Esempio (2)

$\underline{G}$:= G

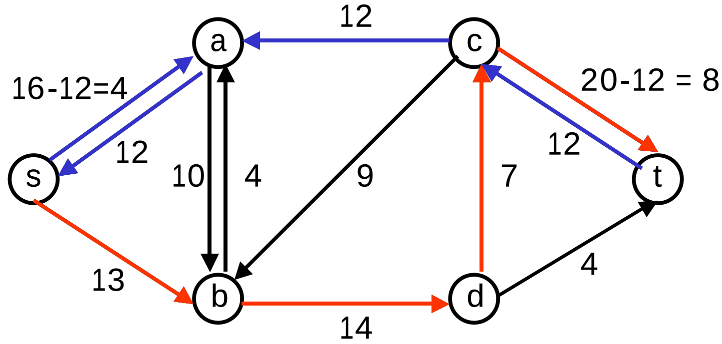


(s,a),(a,c),(c,t) è un cammino aumentante

Δ = min{16,12,20} = 12

Esempio (3)

G diventa

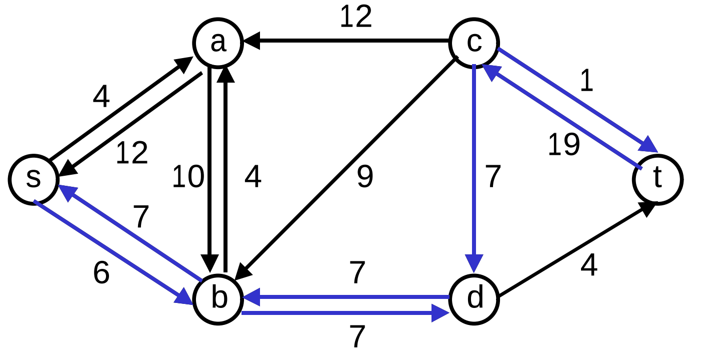


$\varphi_0 := \varphi_0 + 12 = 12$

Nel nuovo G: (s,b),(b,d),(d,c),(c,t) è un cammino aumentante e $\Delta = 7$

Esempio (4)

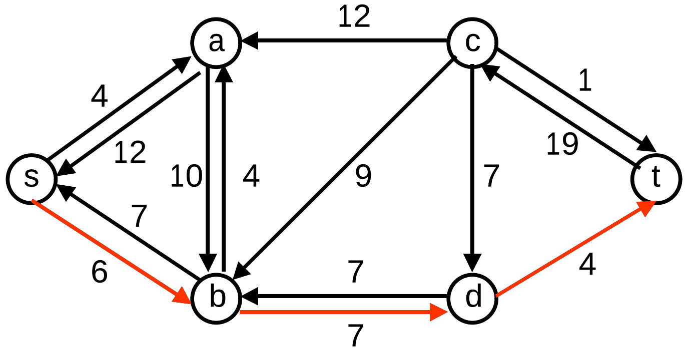
G diventa



$\varphi_0 := \varphi_0 + 7 = 19$

Esempio (5)

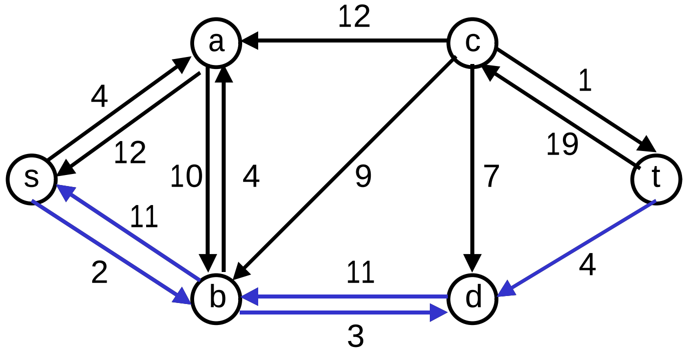
φ_0 := 19 e G diventa



Nel nuovo G: (s,b),(b,d),(d,t) è un cammino aumentante e $\Delta = 4$

Esempio (6)

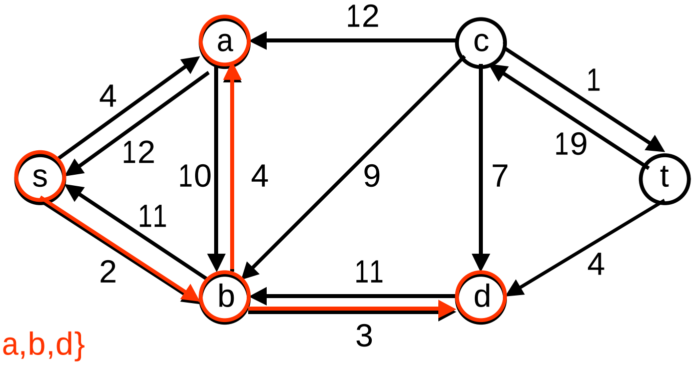
φ_0 := 23 e G diventa



Nel nuovo G non c'è un cammino aumentante e ottimo:=true

Esempio (7)

φ_0 := 23 e G diventa

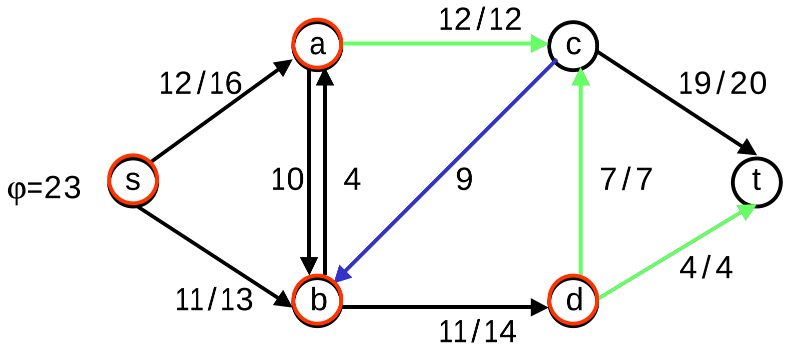


S^* ={s,a,b,d}
N\S^*={c,t}

Esempio (8)

Dati G, x,
S^* ={s,a,b,d} e N\S^*={c,t}

$\delta^+_G(S^*)$:= {(a,c), (d,c), (d,t)}
$\delta^-_G(S^*)$:= {(c,b)}



Flusso attraverso S^* è 12+ 7+4 = 23
Capacità della sezione è 12+ 7+ 4 = 23

$\varphi(S^*) = k(S^*)$

Esempi di applicazione del problema del massimo flusso

- Spedizioni con autobotti
- Collegamenti tra processori
- Assegnamento di compiti
- Traffico autostradale
- Traffico aereo
- Informazioni via Internet
- Telefonate

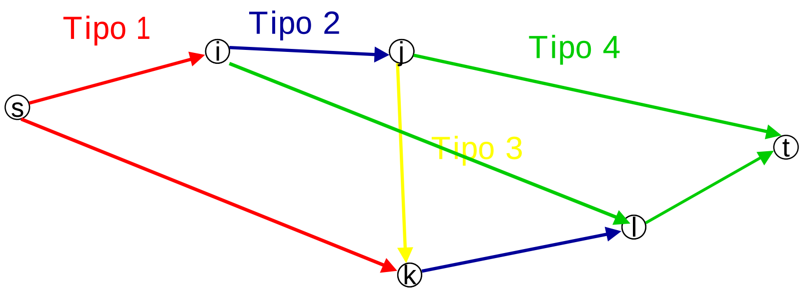
Spedizioni con autobotti

- Il problema del trasporto di petrolio su autobotti in una rete di terminal
- Le autobotti sono differenti in capacità
- Un pozzo sorgente e una raffineria destinazione
- Qual'è la quantità di petrolio trasportato da una flotta di autobotti di tipo diverso in una rete di terminal tra il pozzo sorgente e la raffineria destinazione
- Problema del massimo flusso

Spedizioni con autobotti

- Ogni arco rappresenta il collegamento tra un terminal i e un altro j
- Il valore della capacità di ogni arco è uguale a quella della capacità dell'autobotte
- Nel prossimo disegno, colori diversi, capacità diverse

Spedizioni con autobotti come problema di MF



- φ è la quantità di petrolio trasportato dalla flotta di autobotti

Calcolo distribuito tra processori

- Il problema di assegnare i moduli di un programma tra due processori in parallelo
- I moduli hanno costi differenti sui due processori
- I moduli hanno costi di comunicazione tra loro
- Qual'è l'assegnamento che minimizza i costi di calcolo e comunicazione
- Problema del taglio di capacità minimo

Modello

- α_i costo modulo i sul processore 1
- β_j costo modulo j sul processore 2
- γ_{ij} costo di comunicazione se i moduli i e j sono assegnati a processori diversi