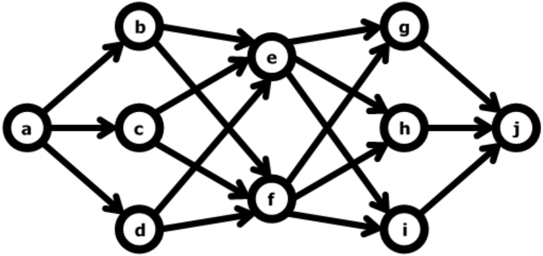
Tutoraggio Ricerca Operativa 2020/2021
8. Teoria dei Grafi:
Il massimo flusso in una rete

Aurora Rossi, Alice Raffaele, Romeo Rizzi

Università degli Studi di Verona

14 aprile 2021

Reti di flusso



- Consideriamo una **rete diretta e pesata**, i.e., un grafo connesso $G = (V, A)$ dove a ogni (u, v) è associato un valore;
- La freccia di ogni *arco* indica quindi il verso di percorrenza del *flusso*;
- Il valore/peso di un arco (u, v) rappresenta la **capacità** dell'arco, i.e., la massima quantità di flusso che l'arco può sopportare (indicato con $c(u, v)$);
- Un vertice con solo archi uscenti è detto **sorgente** (indicato con s);
- Un vertice con solo archi entranti è detto **pozzo** (indicato con t).

Flusso

Un **flusso** in G è una funzione a valori reali $f : V \times V \to R$ che soddisfa due proprietà:

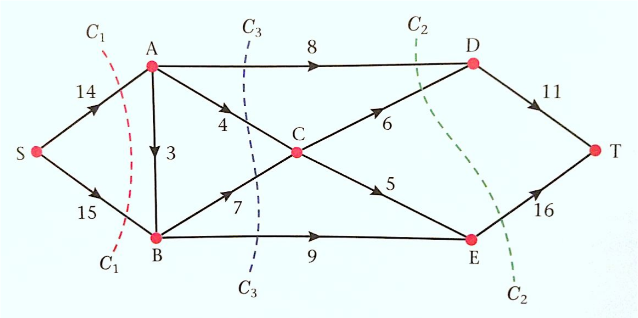
- **Ammissibilità del flusso**: $\forall(u, v), 0 \leq f(u, v) \leq c(u, v)$ (il flusso passante in un arco non deve eccedere la capacità dell'arco);
- **Conservazione del flusso** $\forall u \in V \setminus \{s, t\}$:
$$\sum_{v \in V} x_{uv} = \sum_{v \in V} x_{vu}$$

Se il flusso passante in un arco è uguale alla capacità dell'arco stesso, allora l'arco si dice **saturo**.
Se il flusso passante in un arco è pari a 0, allora l'arco si dice **vuoto** o **non sfruttato**.

Reti e tagli

- Un **taglio**, in una rete avente sorgente S e pozzo T, corrisponde a un set di archi che, se rimossi, separerebbero la rete in due parti P_1 e P_2, dove P_1 include S e P_2 include T;
- La **capacità del taglio** è la somma delle capacità degli archi del taglio stesso;
- Un taglio è **minimo** quando la sua capacità è la più piccola possibile;
- **Nota**: si considerano gli archi che *fluiscono nel taglio*, i.e., quelli diretti da P_1 a P_2; gli archi che *fuoriescono dal taglio* non contribuiscono.

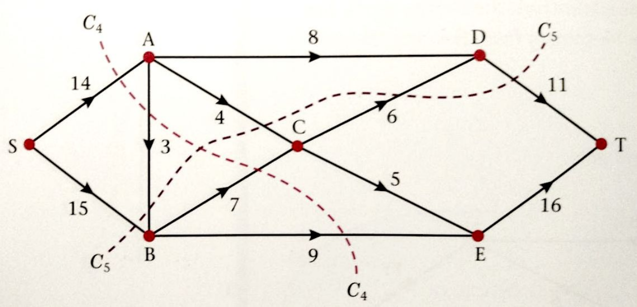
Esempio 1



Quanto valgono le capacità dei tagli C_1, C_2 e C_3?

- C_1: separa S e $\{V \setminus S\}$; il taglio è composto dagli archi SA (14) e SB (15), quindi la capacità totale è 29;
- C_2: composto dagli archi AD (8), CD (6) ed ET (16), quindi la capacità totale è 30;
- C_3: composto dagli archi AC (4), AD (8), BC (7) e BE (9), quindi la capacità totale è 28.

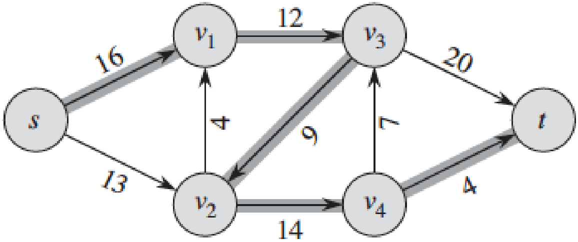
Esempio 2



Quanto valgono le capacità dei tagli C_4 e C_5?

- C_4: separa $\{S, B\}$ dal resto della rete; nel taglio consideriamo solo gli archi SA (14), BC (7) e BE (9), quindi la capacità totale è 30; escludiamo l'arco AB (entrante in B);
- C_5: separa $\{S, A, D\}$ dal resto della rete; nel taglio consideriamo solo gli archi SB (15), AB (3), AC (4) e DT (11), quindi la capacità totale è 33; escludiamo l'arco CD (entrante in D).

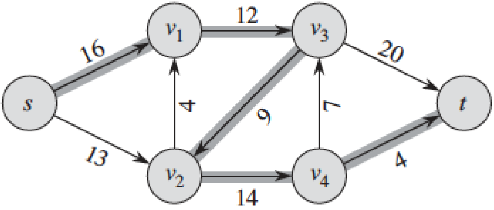
Il problema del massimo flusso



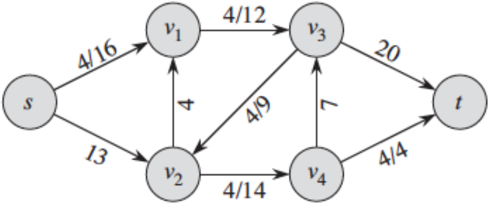
Data una rete $G = (V, A)$, avente sorgente s e pozzo t, si vuole determinare il flusso con massimo valore, i.e., si vuole stabilire la quantità di flusso da mandare su ogni arco, rispettando ammissibilità e conservazione del flusso, massimizzando il flusso totale.

Flusso iniziale

Si trova un flusso iniziale individuando un cammino da s a t:



In questo caso la quantità di flusso è limitata dalla capacità dell'arco (v_4, t); possiamo scrivere il flusso e la capacità su ogni arco usato:



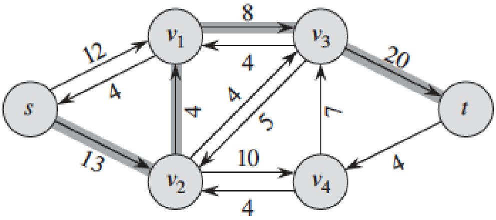
Rete residua

Il flusso che abbiamo trovato è massimo o no? Definiamo la **rete residua** G_f:

- Lasciando invariati gli archi non usati nel flusso;
- Sdoppiando ogni arco usato:
  - Aggiorniamo la capacità dell'arco con quella residua ancora sfruttabile;
  - Aggiungiamo un **arco di ripensamento** nella direzione opposta avente come capacità il flusso che è stato mandato.

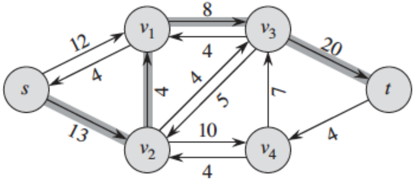
Esempio: sull'arco (s, v_1) con capacità 16 è mandato un flusso pari a 4:

- Aggiorniamo la capacità dell'arco (s, v_1) a $16 - 4 = 12$;
- Aggiungiamo l'arco di ripensamento (v_1, s) con capacità 4.

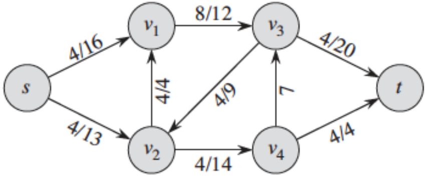


Cammino aumentante

Un **cammino aumentante** è un altro cammino da s a t individuato sulla rete residua, che può essere usato per aumentare il flusso.
Esempio: s-v_2-v_1-v_3-t.



Ciò ci consente di aggiornare la rete con il flusso mandato:



L'algoritmo di Ford-Fulkerson

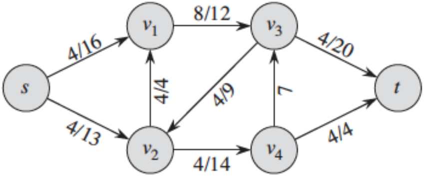
Input: una rete diretta e pesata $G = (V, A)$, due vertici s e t;
Output: valore del massimo flusso da s a t;

1. Inizializzare il flusso $f := 0$;
2. Finché esiste un cammino aumentante p da s a t nella rete residua G_f, incrementare il flusso f lungo il cammino p;
3. Return f^*.

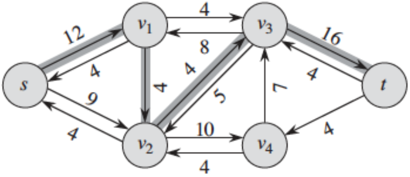
Nota: conviene tenere entrambi i disegni della rete residua G_f e della rete G dove teniamo nota del flusso inviato; li aggiorneremo entrambi a ogni iterazione, fino alla soluzione ottima.

Esercizio (I)

Riprendiamo l'esempio usato nelle slides precedenti. **Valore del flusso**: 8.



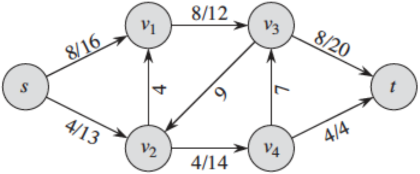
Proviamo a incrementare il flusso considerando la **rete residua** G_f:



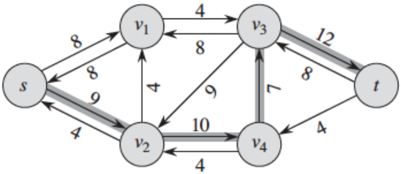
Cammino aumentante: s-v_1-v_2-v_3-t
Incremento del flusso: 4 (archi limitanti: (v_1, v_2) e (v_2, v_3)).

Esercizio (II)

Aggiornamento della rete G - **Valore del flusso**: 12.



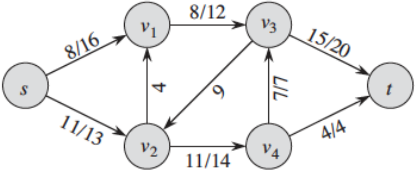
Aggiornamento della **rete residua** G_f:



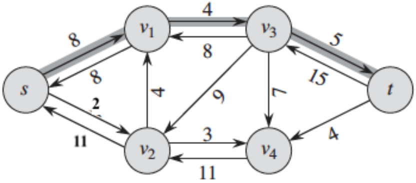
Cammino aumentante: s-v_2-v_4-v_3-t
Incremento del flusso: 7 (arco limitante: (v_4, v_3)).

Esercizio (III)

Aggiornamento della rete G - **Valore del flusso**: 19.



Aggiornamento della **rete residua** G_f:

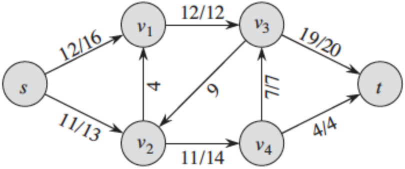


Cammino aumentante: s-v_1-v_3-t

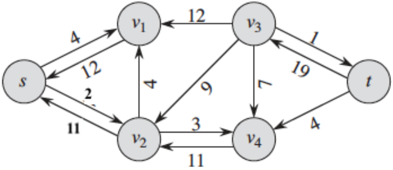
Incremento del flusso: 4 (arco limitante: (v_1, v_3)).

Esercizio (IV)

Aggiornamento della rete G - **Valore del flusso**: 23.



Aggiornamento della **rete residua** G_f:



$\nexists$ **cammino aumentante** $\rightarrow$ Valore massimo del flusso: 23.

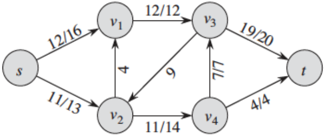
Note

- Se tra u e v esistono già i due archi in entrambe le direzioni, allora quando si aggiorna la rete residua non si aggiungono altri archi multipli ma si provvede solo ad aggiornare i pesi sugli archi esistenti;
- Cercando un cammino aumentante, si possono sfruttare tutti gli archi che non hanno capacità pari a 0 (anche gli archi di ripensamento);
- Non bisogna preoccuparsi di usare gli archi di ripensamento, perché si sta come *ridirezionando* il flusso, non lo si sta diminuendo.

Il teorema del massimo flusso - minimo taglio

Come essere certi che il flusso trovato sia proprio massimo? Sfruttando il concetto di minimo taglio che abbiamo visto prima:

- Se si riesce a trovare un taglio avente capacità pari al valore massimo del flusso, allora il flusso è massimo;
- Il minimo taglio passa attraverso:
  - Archi saturi, quando diretti dalla sorgente al pozzo;
  - Archi vuoti (non sfruttati), quando diretti dal pozzo alla sorgente;
- **Nota**: questo non vuol dire che tutti gli archi saturi o non sfruttati debbano essere per forza nel taglio!



Gli archi coinvolti nel taglio sono (v_1, v_3), (v_4, v_3) e (v_4, t).
Capacità del taglio: $12 + 7 + 4 = 23$.
NB: usare le capacità degli archi (e NON il valore del flusso degli archi).

Analisi dell'algoritmo di Ford-Fulkerson

Quanto costano i singoli step?

Input: una rete diretta e pesata $G = (V, A)$, due vertici s e t;
Output: valore del massimo flusso da s a t;

1. Inizializzare il flusso $f := 0$; $O(1)$
2. Finché esiste un cammino aumentante p da s a t nella rete residua G_f, incrementare il flusso f lungo il cammino p; $O(|E| \cdot f^*)$
3. Return f^*. $O(1)$.

Lo step 2 dipende da come si trovano i cammini aumentanti:

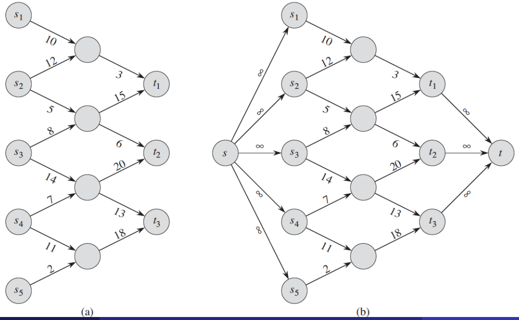
- Senza un criterio preciso, l'algoritmo potrebbe anche non terminare;
- Usando breadth-first search o depth-first search, si può trovare un cammino aumentante in tempo $O(E)$;
- Il *finché* corrisponde a un ciclo **while**, ripetuto al massimo f^* volte;

Complessità temporale: $O(|E| \cdot f^*)$.

Estensione dell'algoritmo a reti con più sorgenti e/o pozzi

- **Super-sorgente**: si aggiunge un macro-nodo all'inizio collegato a tutti i nodi sorgenti, in modo tale che la capacità dei suoi archi uscenti corrisponda alle capacità delle singole sorgenti;
- **Super-pozzo**: viceversa, si aggiunge un macro-nodo alla fine, raggiungibile da tutti i pozzi, in modo tale che la capacità dei suoi archi entranti corrisponda alle capacità dei singoli pozzi.

Esempio:



Bibliografia e sitografia

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, *Introduction to Algorithms, Third Edition*, The MIT Press, pag. 708-727.
- Susie G. Jameson, *Decision Mathematics 2, Edxecel AS and A-level Modular Mathematics*, Pearson Education Limited.