

Lezioni di Ricerca Operativa

Corso di Laurea in Informatica

Università di Salerno

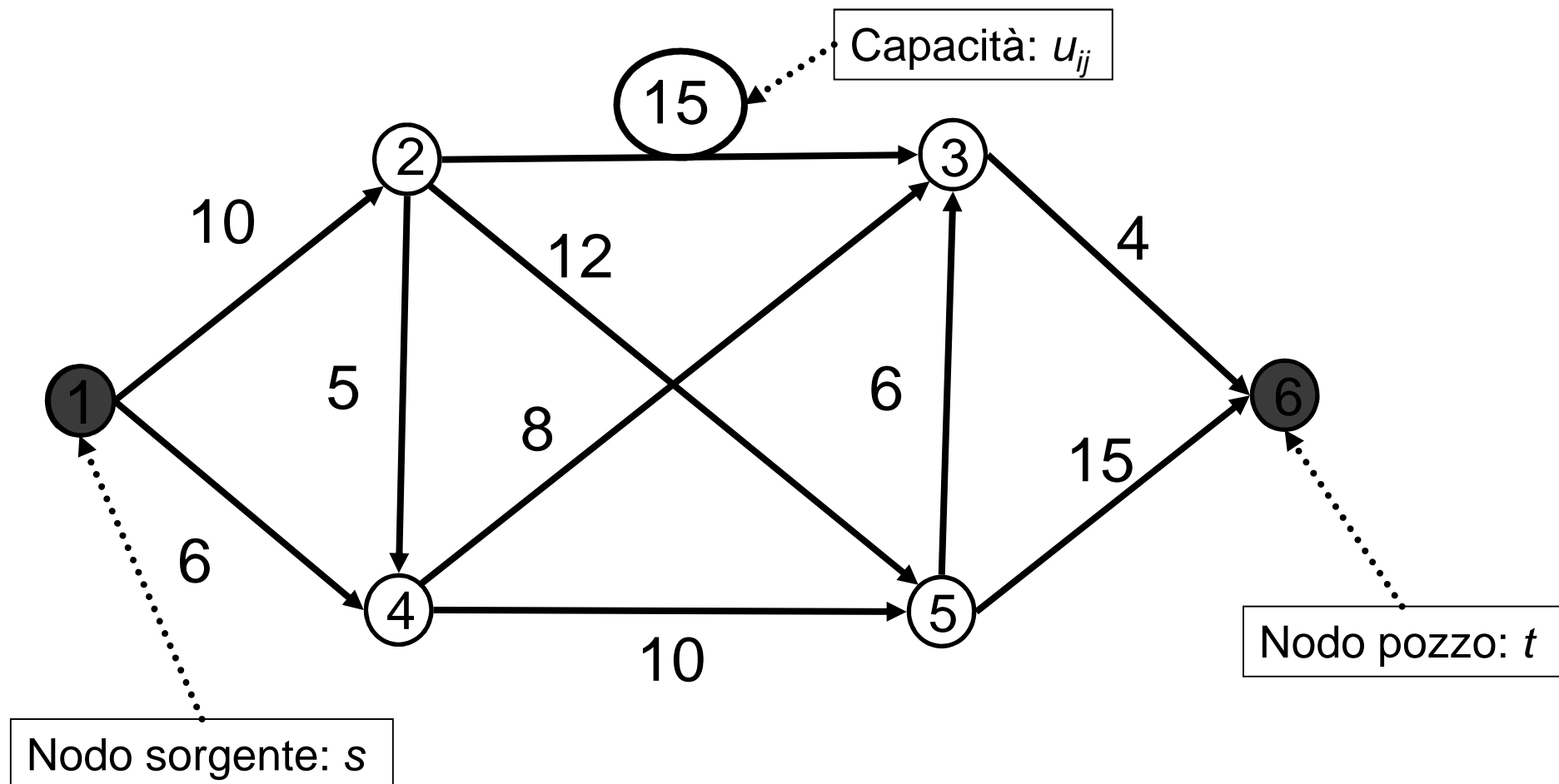
Problema del Massimo Flusso:

- Formulazione Matematica
- Teorema flusso massimo / taglio minimo
- Algoritmo del Grafo Ausiliario

Prof. Cerulli – Dott.ssa Gentili – Dott. Carrabs

Il Problema del Massimo Flusso

Sia $G = (V,A)$ un grafo orientato su cui sia definito un vettore $\underline{u} = [u_{ij}]$ delle capacità associate agli archi del grafo; inoltre, siano s e t due nodi distinti, detti rispettivamente *sorgente* (o *origine*) e *pozzo* (o *destinazione*). Il problema del flusso massimo consiste nel determinare la massima quantità di flusso che è possibile inviare da s a t attraverso G .



Il Problema del Massimo Flusso

Nodo sorgente fornisce flusso $\rightarrow f$

Nodo destinazione assorbe flusso $\rightarrow -f$

Tutti gli altri nodi sono nodi di transito



Voglio spedire dalla sorgente la massima quantità di flusso f fino al pozzo senza violare i vincoli di capacità

Il Problema del Massimo Flusso: formulazione

Parametri di input:

- Grafo orientato $G=(V,A)$
- Nodo sorgente s
- Nodo destinazione t

u_{ij} = capacità dell'arco (i, j)

Variabili decisionali:

x_{ij} = quantità di flusso che viaggia sull'arco (i, j)

f = flusso totale inviato dalla sorgente al pozzo

Il Problema del Massimo Flusso: formulazione

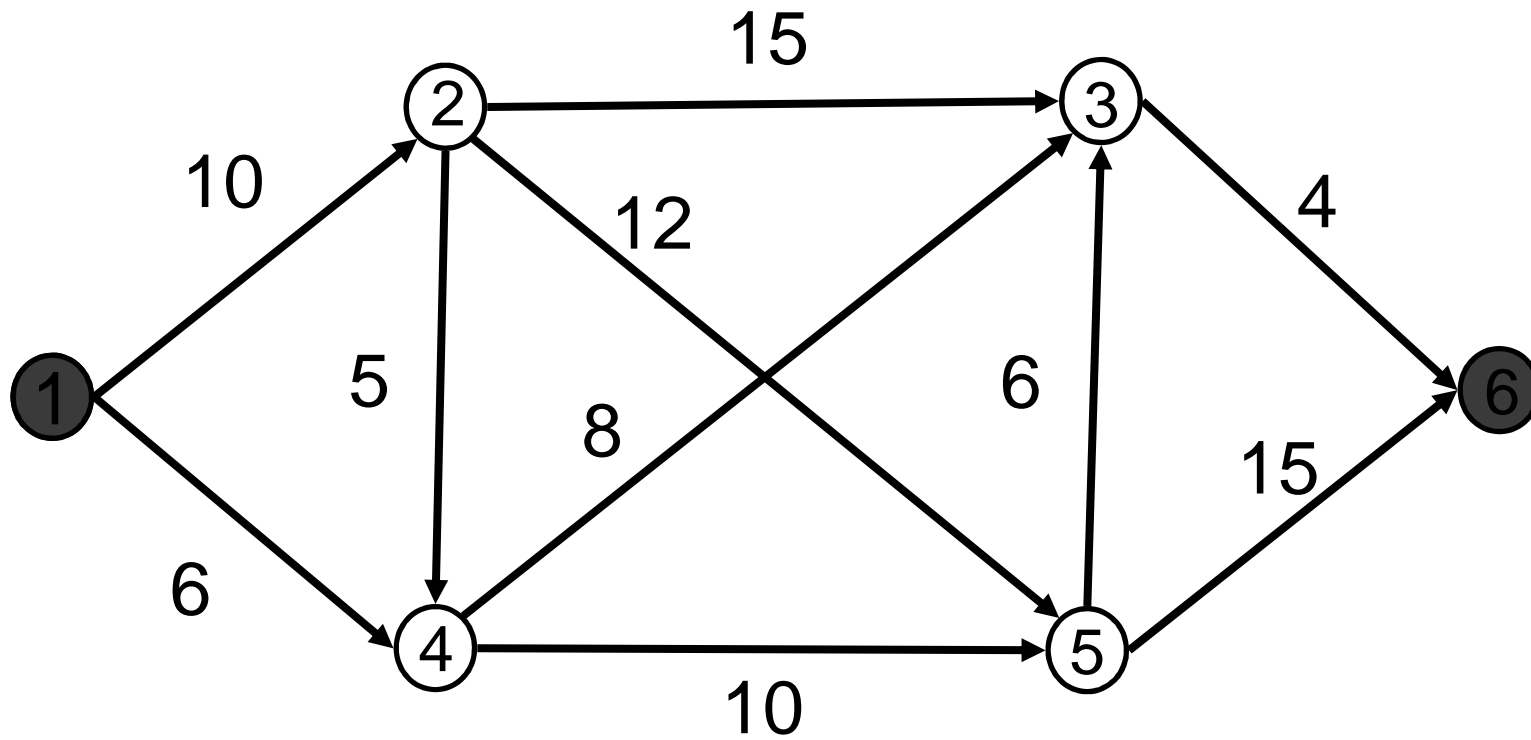
max f

con vincoli:

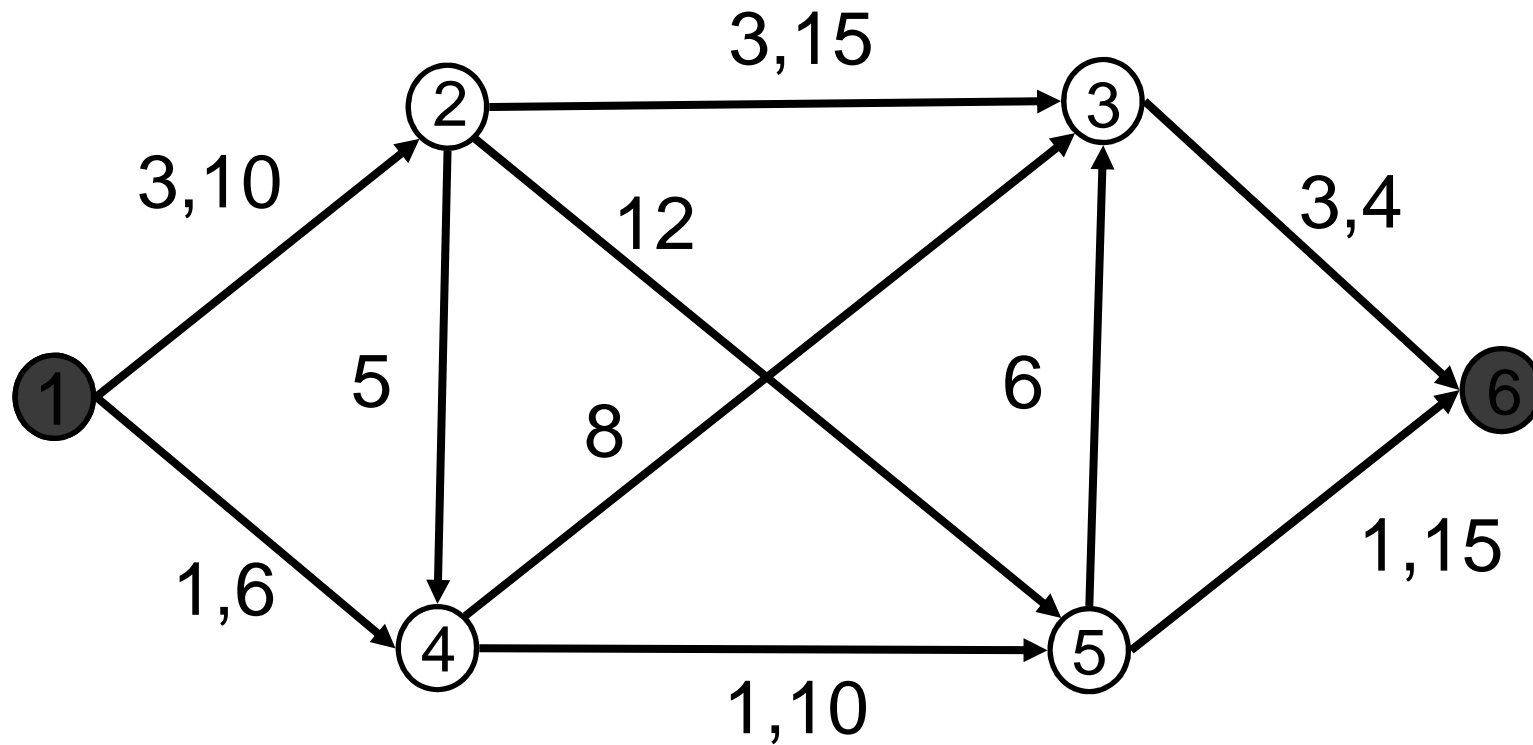
$$\sum_{j \in FS(i)} x_{ij} - \sum_{k \in BS(i)} x_{ki} = \begin{cases} 0 & \forall i \in V \ i \neq s, t \\ f & \text{se } i = s \\ -f & \text{se } i = t \end{cases} \quad (1)$$

$$0 \leq x_{ij} \leq u_{ij} \quad \forall (i, j) \in A \quad (2)$$

Il Problema del Massimo Flusso: esempio

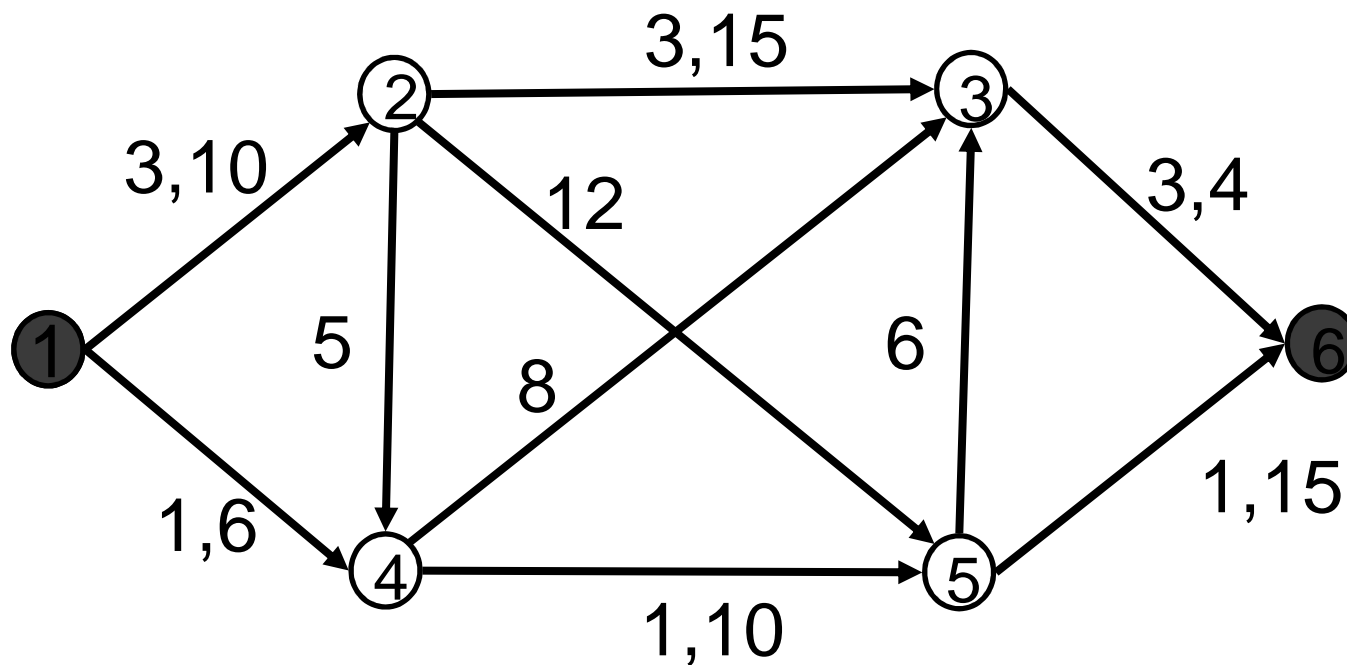


Il Problema del Massimo Flusso: esempio



$$x_{12} = 3 \quad x_{23} = 3 \quad x_{36} = 3 \quad x_{14} = 1 \quad x_{45} = 1 \quad x_{56} = 1$$

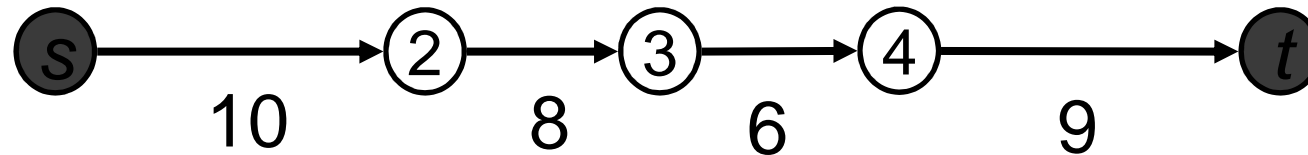
Il Problema del Massimo Flusso: esempio



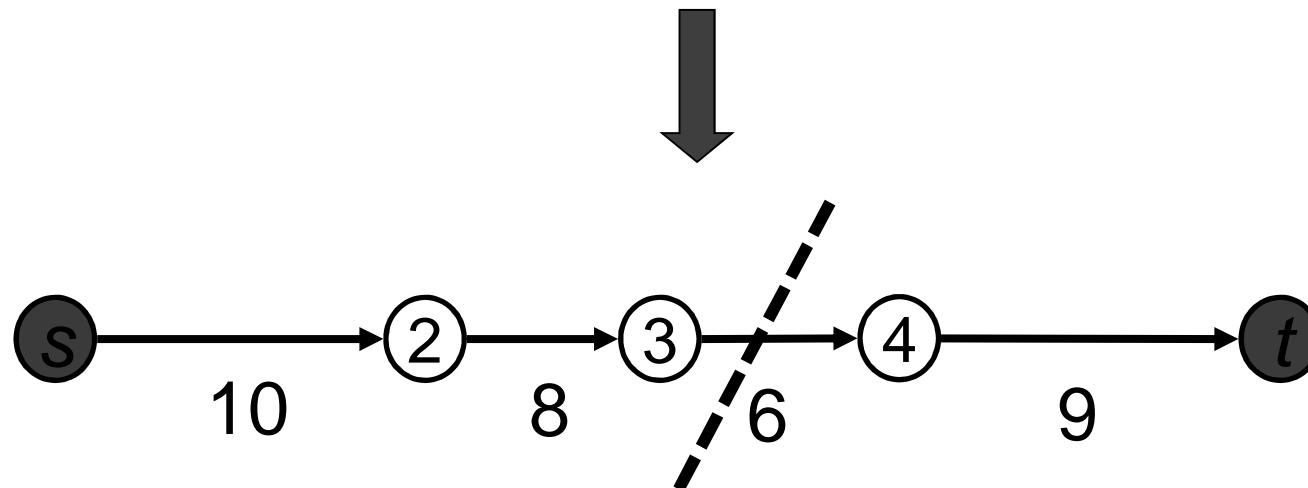
$$\underline{X} = \begin{bmatrix} x_{12} \\ x_{14} \\ x_{23} \\ x_{24} \\ x_{25} \\ x_{36} \\ x_{43} \\ x_{45} \\ x_{53} \\ x_{56} \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 3 \\ 0 \\ 0 \\ 3 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

\underline{X} rappresenta un **flusso ammissibile**, con valore $f=4$, per il grafo G .

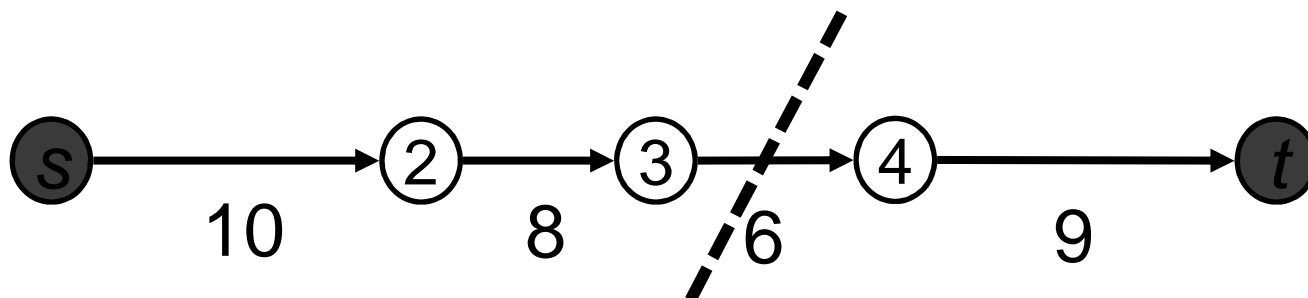
Il Problema del Massimo Flusso: concetti fondamentali



Il flusso massimo su questo grafo è pari a 6 (corrispondente alla capacità minima degli archi del cammino).



Il Problema del Massimo Flusso: concetti fondamentali



La retta in figura mostra un **taglio s-t** del grafo, ossia un partizionamento dei vertici del grafo in due sottinsiemi $V_1 = \{s, 2, 3\}$ e $V_2 = \{4, t\}$ tali che:

- Il nodo sorgente appartiene a V_1
- Il nodo pozzo appartiene a V_2
- $V_1 \cup V_2 = V$
- $V_1 \cap V_2 = \emptyset$

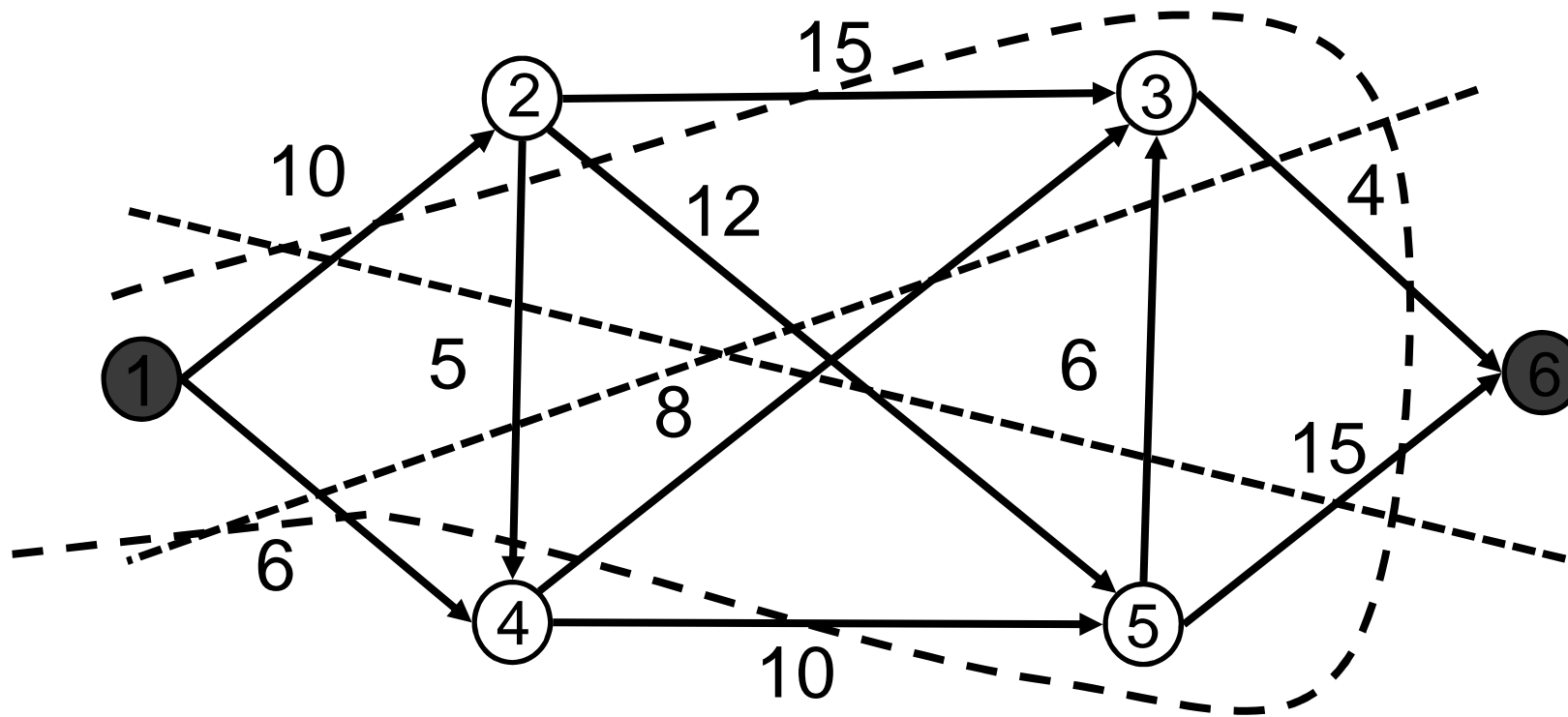
Definizione:

- **Archi diretti** del taglio $[V_1, V_2]$: $\{(i, j) : i \in V_1 \text{ e } j \in V_2\}$
- **Archi inversi** del taglio $[V_1, V_2]$: $\{(p, q) : p \in V_2 \text{ e } q \in V_1\}$

Estendiamo questo concetto di taglio s-t ad un grafo più complesso.

(nel resto della lezione ometteremo "l's-t" tenendo presente però che faremo riferimento sempre a tagli di questa tipologia)

Taglio di un grafo e archi del taglio

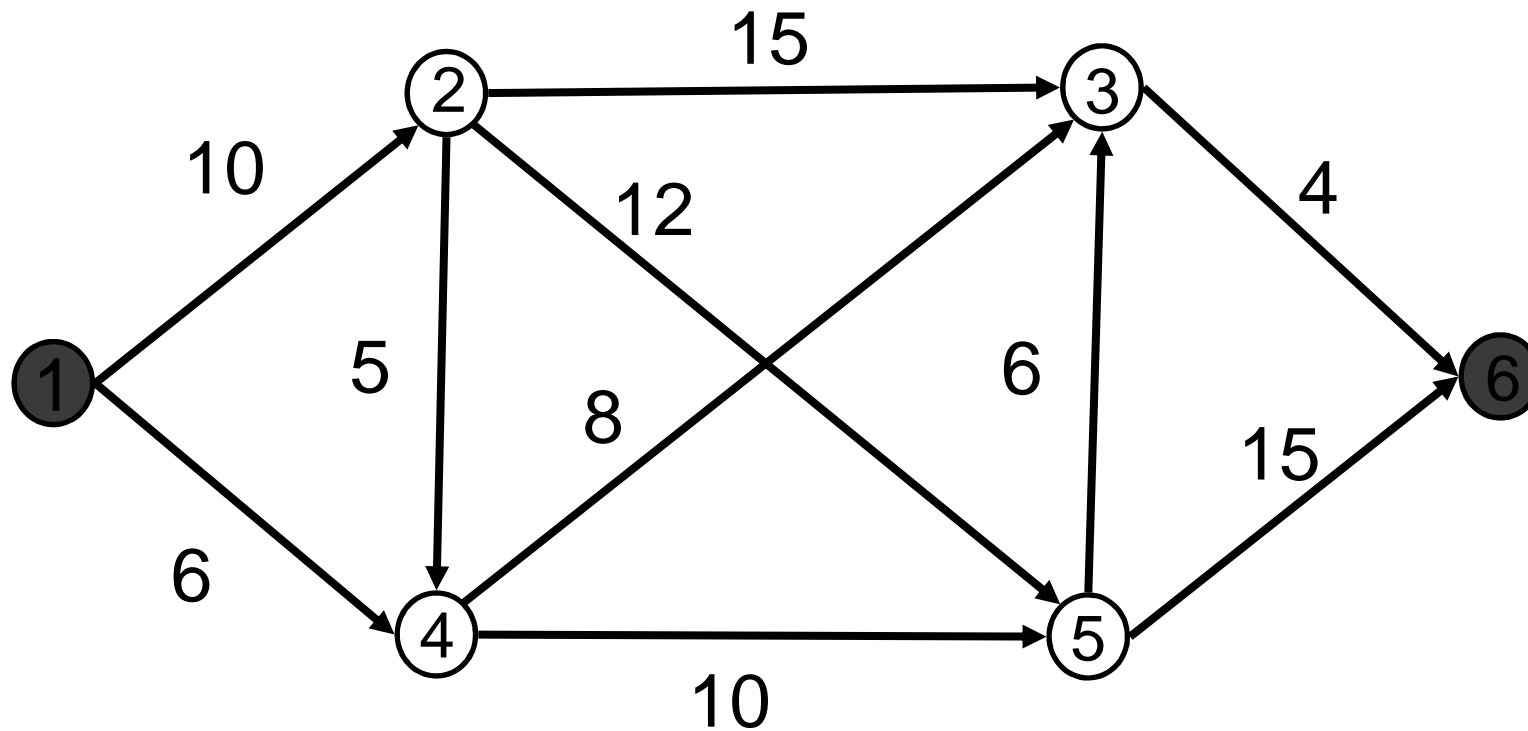


Taglio 1: $V_1 = \{1,2,3\}$ $V_2 = \{4,5,6\}$ \rightarrow archi "diretti" del taglio = $\{(1,4) (2,4) (2,5) (3,6)\}$

Taglio 2: $V_1 = \{1,3,5\}$ $V_2 = \{2,4,6\}$ \rightarrow archi "diretti" del taglio = $\{(1,2) (1,4) (3,6) (5,6)\}$

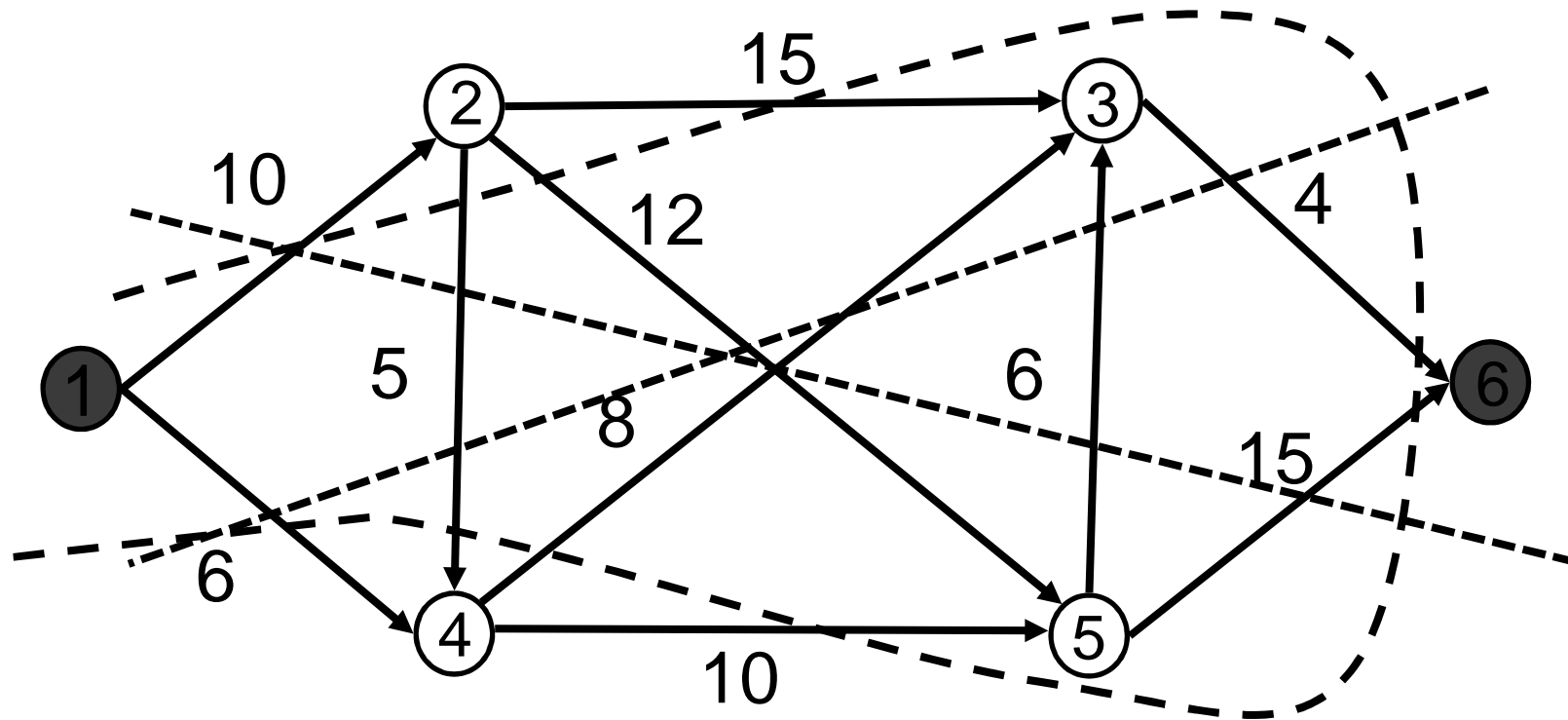
Taglio 3: $V_1 = \{1,4,5\}$ $V_2 = \{2,3,6\}$ \rightarrow archi "diretti" del taglio = $\{(1,2) (4,3) (5,3) (5,6)\}$

Capacità di un taglio



Dato il taglio $[V_1, V_2]$ la **capacità del taglio** è pari alla somma delle capacità degli archi diretti del taglio.

Capacità di un taglio



Taglio 1: $V_1 = \{1, 2, 3\}$ $V_2 = \{4, 5, 6\}$ \rightarrow archi del taglio = $\{(1, 4) (2, 4) (2, 5) (3, 6)\}$
Capacità $u[V_1, V_2] = 6 + 5 + 12 + 4 = 27$

Taglio 2: $V_1 = \{1, 3, 5\}$ $V_2 = \{2, 4, 6\}$ \rightarrow archi del taglio = $\{(1, 2) (1, 4) (3, 6) (5, 6)\}$
Capacità $u[V_1, V_2] = 10 + 6 + 4 + 15 = 35$

Taglio 3: $V_1 = \{1, 4, 5\}$ $V_2 = \{2, 3, 6\}$ \rightarrow archi del taglio = $\{(1, 2) (4, 3) (5, 3) (5, 6)\}$
Capacità $u[V_1, V_2] = 10 + 8 + 6 + 15 = 39$

Relazione tra il massimo flusso e la capacità di un taglio

Proprietà 1:

Il valore di un qualunque flusso ammissibile è minore o uguale alla capacità di un qualunque taglio.

Dim.

Sia X un flusso ammissibile e (V_1, V_2) un qualunque taglio del grafo. Sommando i vincoli di bilanciamento del flusso relativi ai nodi in V_1 otteniamo:

$$f = \sum_{i \in V_1} \left[\sum_{j \in FS(i)} x_{ij} - \sum_{k \in BS(i)} x_{ki} \right]$$

Relazione tra il massimo flusso e la capacità di un taglio

$$f = \sum_{i \in V_1} \left[\sum_{j \in FS(i)} x_{ij} - \sum_{k \in BS(i)} x_{ki} \right]$$



$$f = \sum_{(i,j) \in [V_1, V_2]} x_{ij} - \sum_{(p,q) \in [V_2, V_1]} x_{pq} \leq \sum_{(i,j) \in [V_1, V_2]} x_{ij} \leq \sum_{(i,j) \in [V_1, V_2]} u_{ij} = u[V_1, V_2]$$

Relazione tra il massimo flusso e la capacità di un taglio

La capacità di un taglio fornisce un limite superiore al valore del flusso f che posso spedire dalla sorgente al pozzo

Se ho un flusso ammissibile di valore f e riesco a trovare un taglio la cui capacità è uguale ad f allora posso concludere che il flusso che ho trovato è massimo.

Teorema (Max Flow- Min Cut)

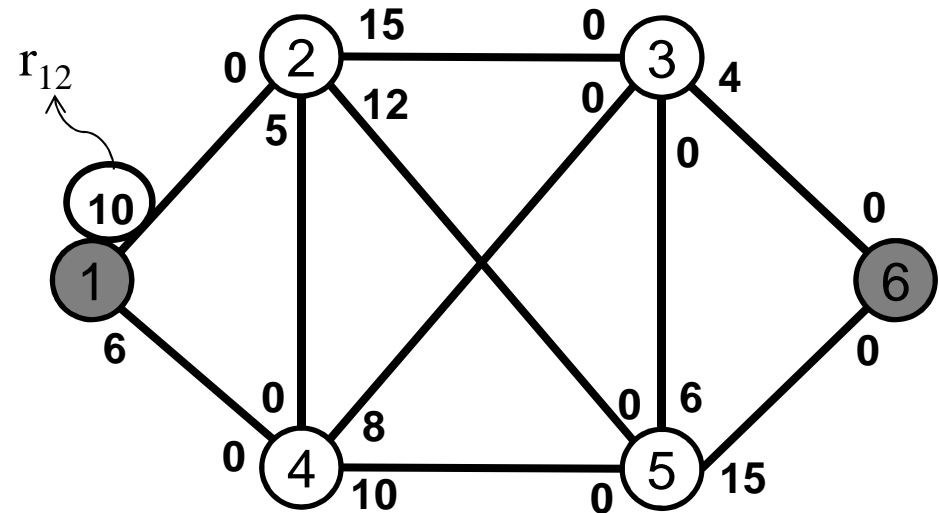
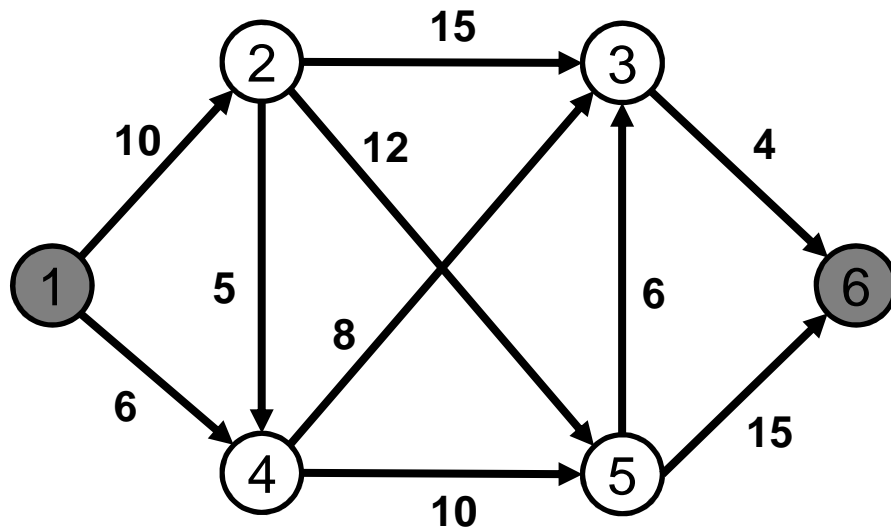
Il flusso massimo che può essere spedito dalla sorgente al pozzo su un grafo orientato G è uguale alla capacità del taglio minimo di G .

Grafo ausiliario e capacità residue

Dato un grafo $G=(V,A)$ ed un flusso ammissibile \underline{X} su G , il **grafo ausiliario** $G(\underline{x})=(V,A')$ è così costruito:

- ✓ $V'=V$
- ✓ A' definisce un grafo bi-orientato ai cui archi è associata una capacità residua $r_{ij}=u_{ij}-x_{ij}+x_{ji}$.

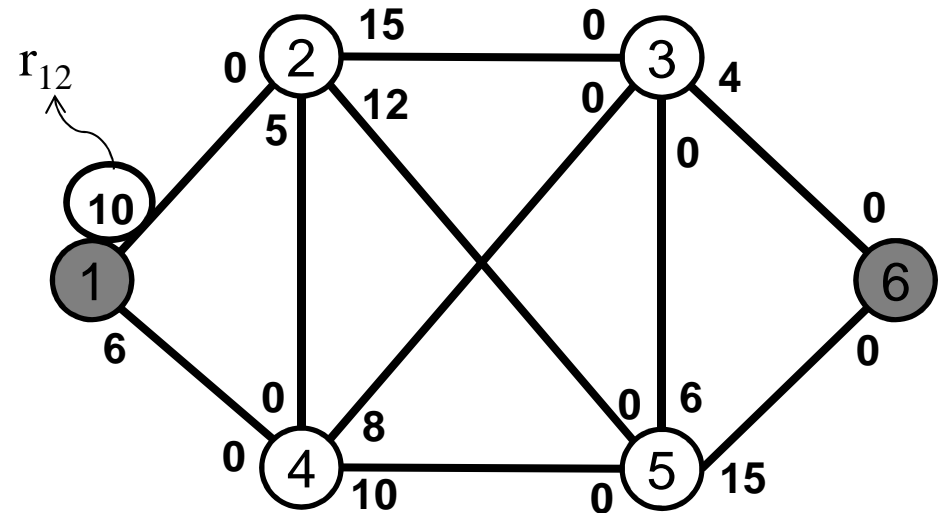
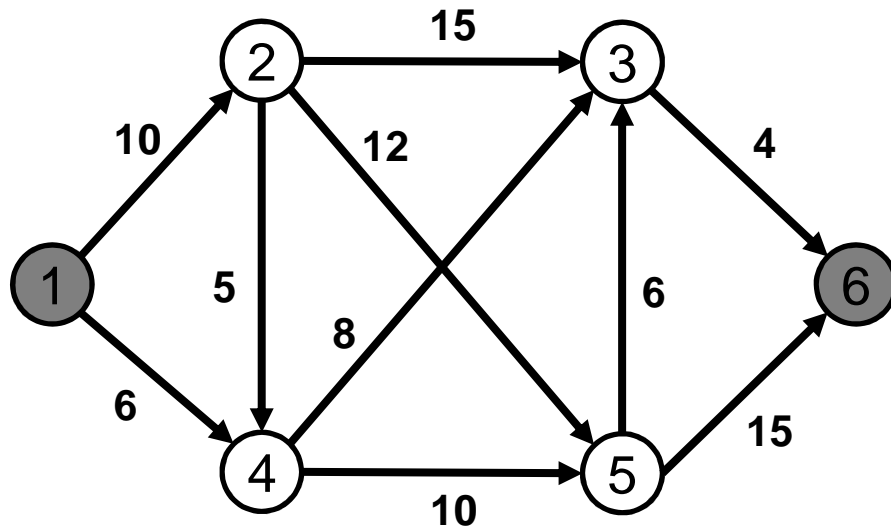
Grafo ausiliario e capacità residue



Calcoliamo la rete residua prodotta dal seguente flusso ammissibile \underline{X} :

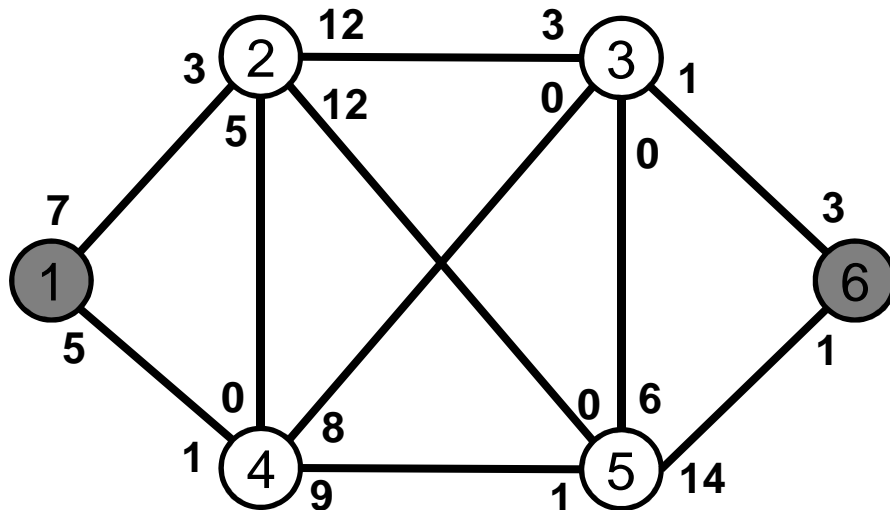
$$\underline{X}^T = [x_{12} \ x_{14} \ x_{23} \ x_{24} \ x_{25} \ x_{36} \ x_{43} \ x_{45} \ x_{53} \ x_{56}] = [3 \ 1 \ 3 \ 0 \ 0 \ 3 \ 0 \ 1 \ 0 \ 1]$$

Grafo ausiliario e capacità residue

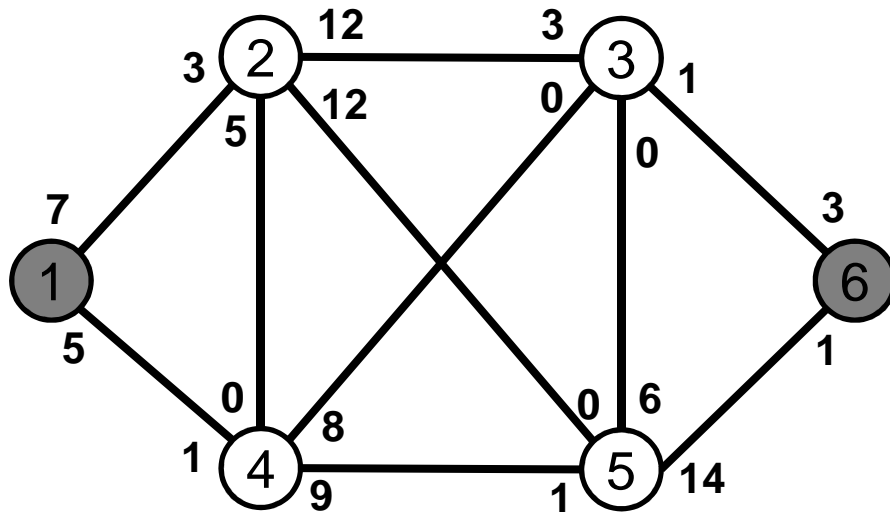


Calcoliamo la rete residua prodotta dal seguente flusso ammissibile \underline{X} :

$$\underline{X}^T = [x_{12} \ x_{14} \ x_{23} \ x_{24} \ x_{25} \ x_{36} \ x_{43} \ x_{45} \ x_{53} \ x_{56}] = [3 \ 1 \ 3 \ 0 \ 0 \ 3 \ 0 \ 1 \ 0 \ 1]$$



Grafo ausiliario e capacità residue



- Se un arco ha capacità residua maggiore di zero, significa che posso spedire ancora del flusso attraverso quell'arco.
- Se riesco ad individuare un cammino da s a t sul grafo ausiliario allora posso spedire del flusso addizionale dalla sorgente al pozzo.
- Un cammino da s a t sul grafo ausiliario viene definito **cammino aumentante**.
- Fino a quando nel grafo ausiliario sono presenti cammini aumentanti è sempre possibile incrementare il flusso da s a t .

Algoritmo dei Cammini Aumentanti

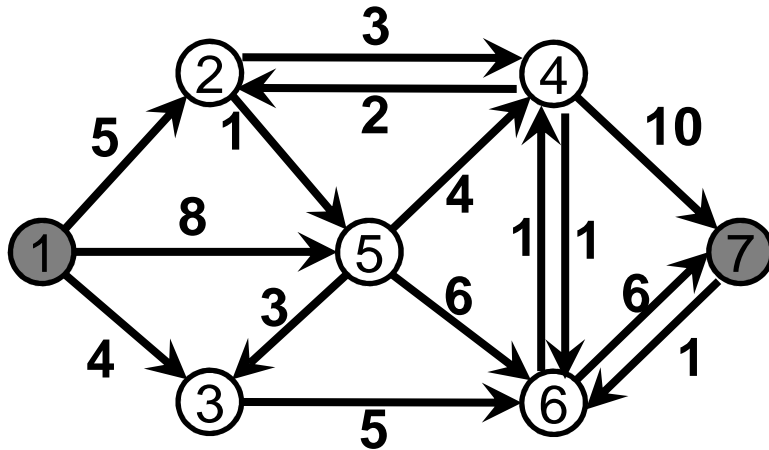
L'algoritmo dei cammini aumentanti risolve il problema del flusso massimo utilizzando il grafo grafo ausiliario (o delle capacità residue) per stabilire come instradare il flusso sulla rete.

Consideriamo un grafo $G=(V,A)$ ed un flusso ammissibile \underline{X} (inizialmente il metodo considera il flusso nullo ossia $x_{ij}=0 \forall (i,j) \in A$).

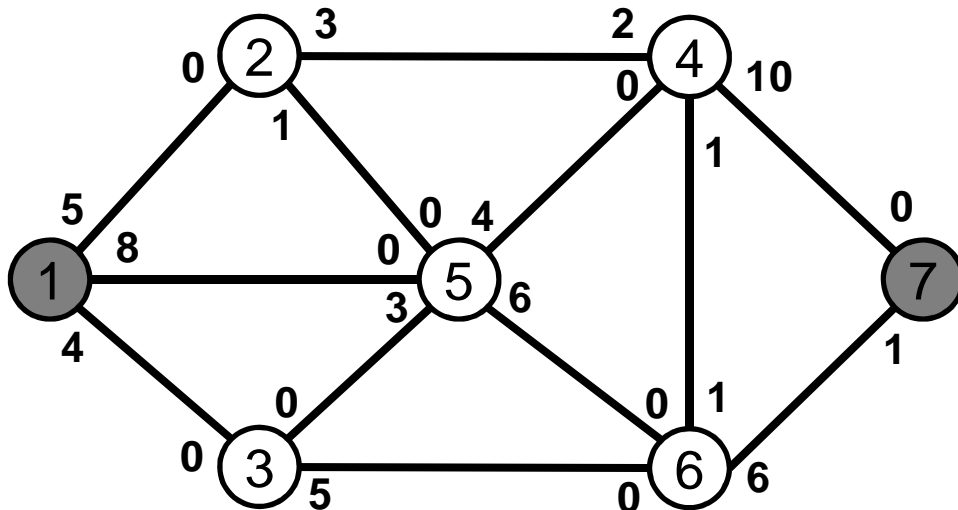
I passi principali dell'algoritmo dei cammini aumentanti sono:

1. Individuare nel grafo ausiliario un qualsiasi cammino p dal nodo sorgente al nodo pozzo su cui è possibile far transitare una quantità di flusso $\Delta > 0$ (cammino aumentante). Se non esiste tale cammino, l'algoritmo si arresta.
2. Il valore del flusso da inviare lungo il cammino p è pari alla capacità residua minima degli archi in p (i.e. $\Delta = \min\{r_{ij} : (i,j) \in p\}$)
3. Incrementare di Δ il valore del flusso f corrente ed aggiornare le capacità residue lungo il cammino p .

Algoritmo del Grafo Ausiliario

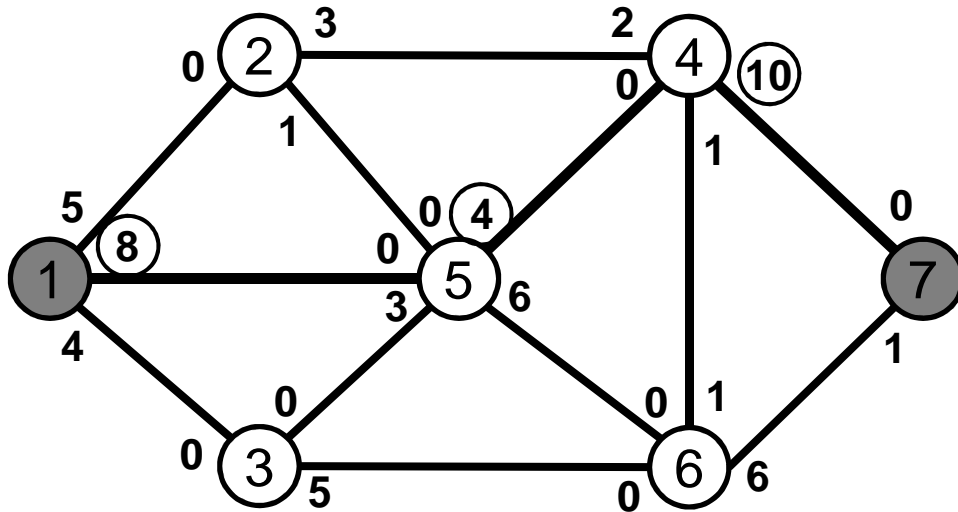


$G=(V,A)$ grafo di input
 $s=1$ nodo sorgente
 $t=7$ nodo pozzo
 $x_{ij} = 0, \forall (i,j) \in A$ flusso ammissibile
 $f=0$



f = 0

Algoritmo del Grafo Ausiliario



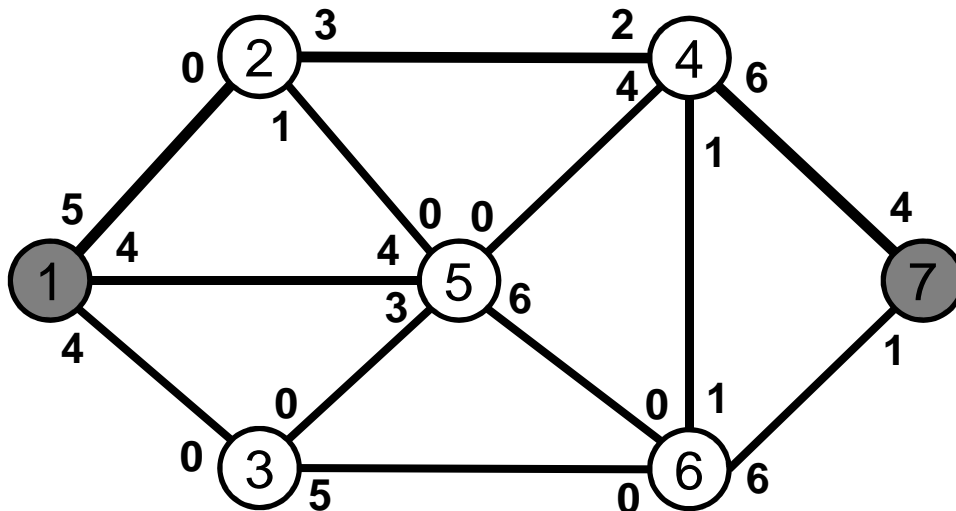
Un path aumentante è un path da s a t sul grafo ausiliario.

Viene chiamato "aumentante" perché permette di aumentare il flusso sul grafo da s a t utilizzando gli archi del path.

Il flusso che posso spedire è uguale alla minima capacità residua degli archi del path.

$$P = 1-5-4-7$$

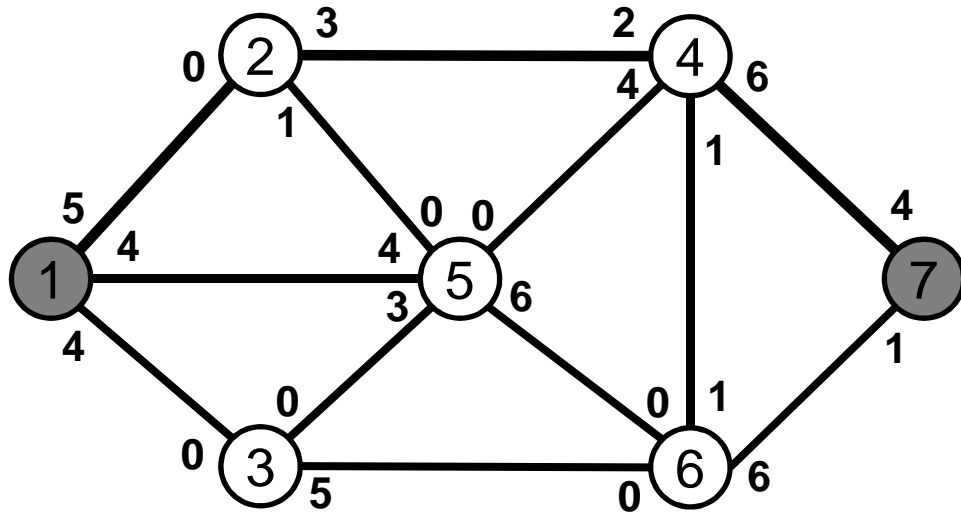
$$\Delta = 4 \quad f = 4$$



$$P = 1-2-4-7$$

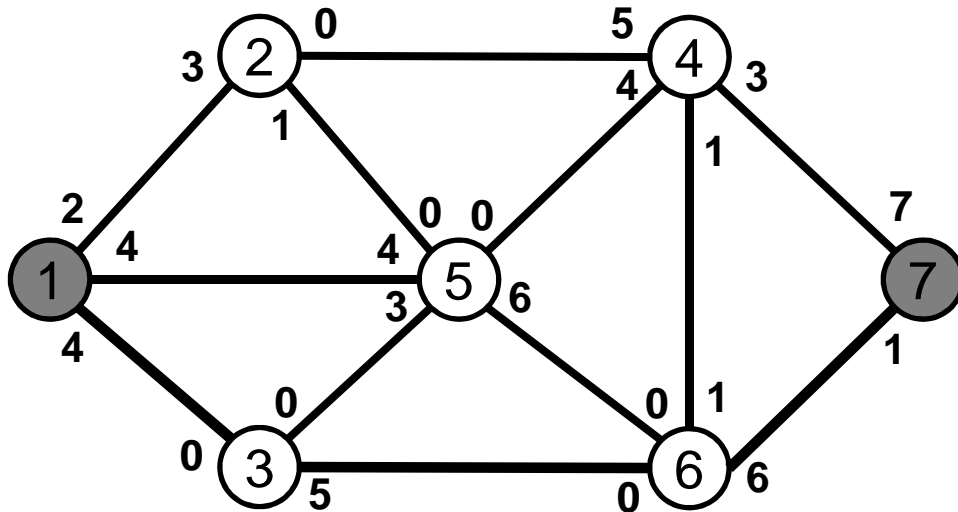
$$\Delta = 3 \quad f = f + \Delta = 7$$

Algoritmo del Grafo Auxiliar



$$P = 1-2-4-7$$

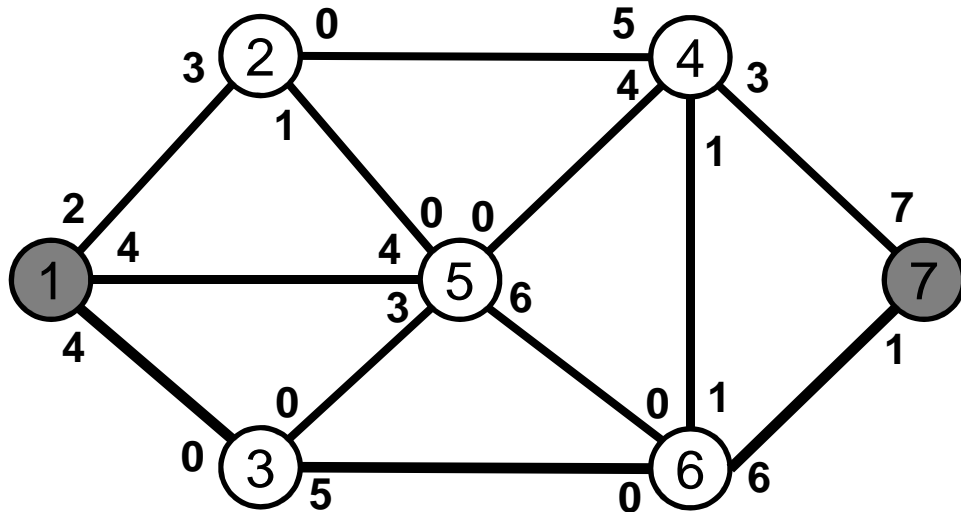
$$\Delta=3 \quad f = f+\Delta = 7$$



$$P = 1-3-6-7$$

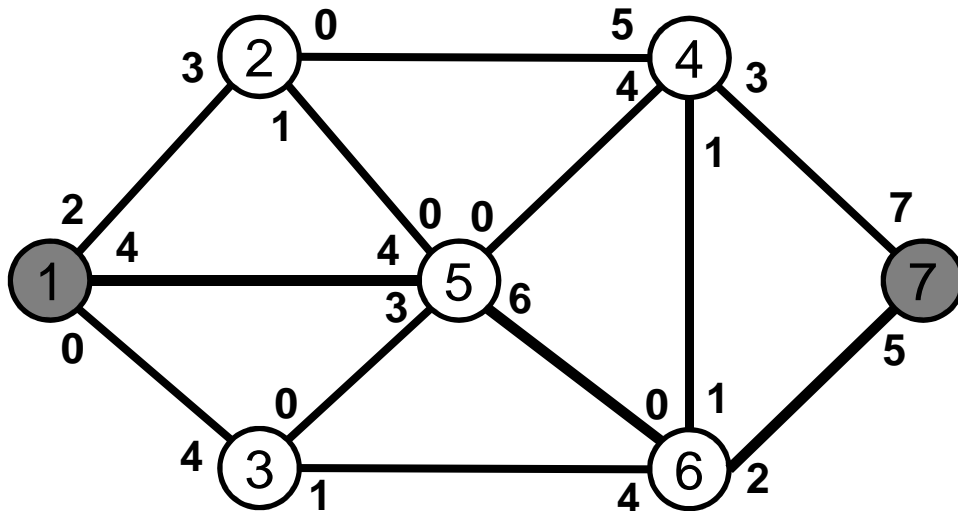
$$\Delta=4 \quad f = f+\Delta = 11$$

Algoritmo del Grafo Auxiliar



$$P = 1-3-6-7$$

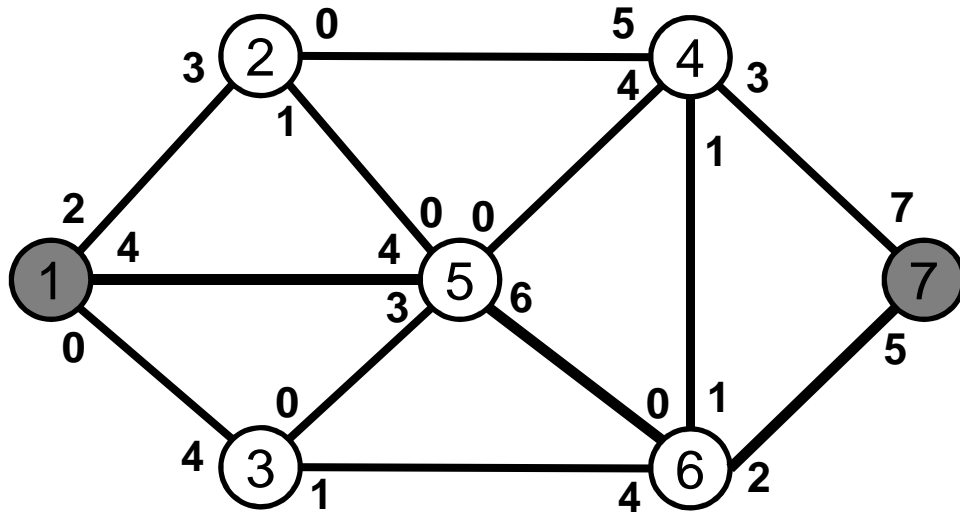
$$\Delta=4 \quad f = f+\Delta = 11$$



$$P = 1-5-6-7$$

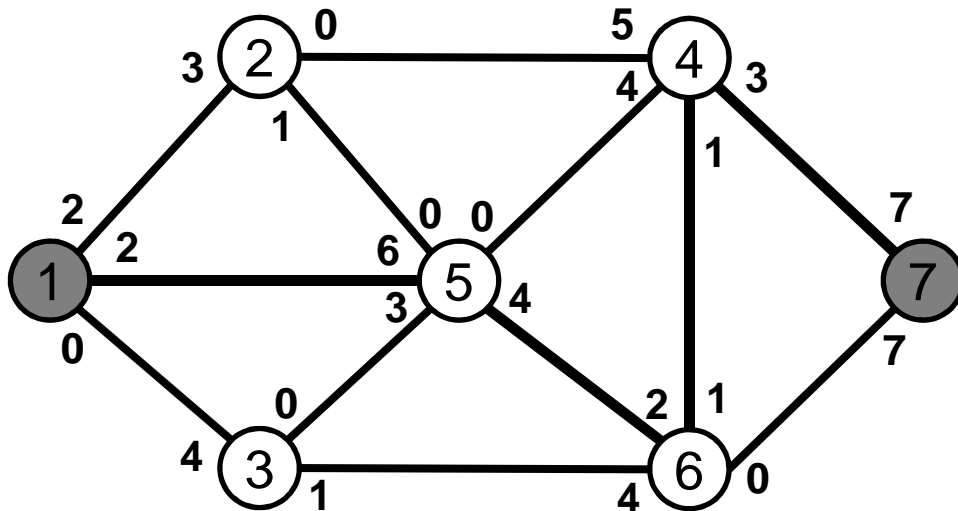
$$\Delta=2 \quad f = f+\Delta = 13$$

Algoritmo del Grafo Auxiliar



$$P = 1-5-6-7$$

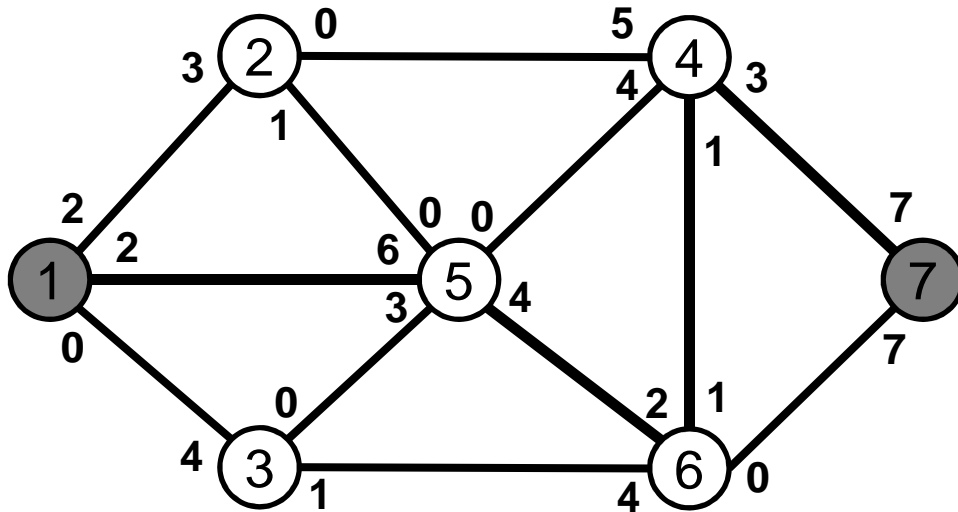
$$\Delta=2 \quad f = f+\Delta = 13$$



$$P = 1-5-6-4-7$$

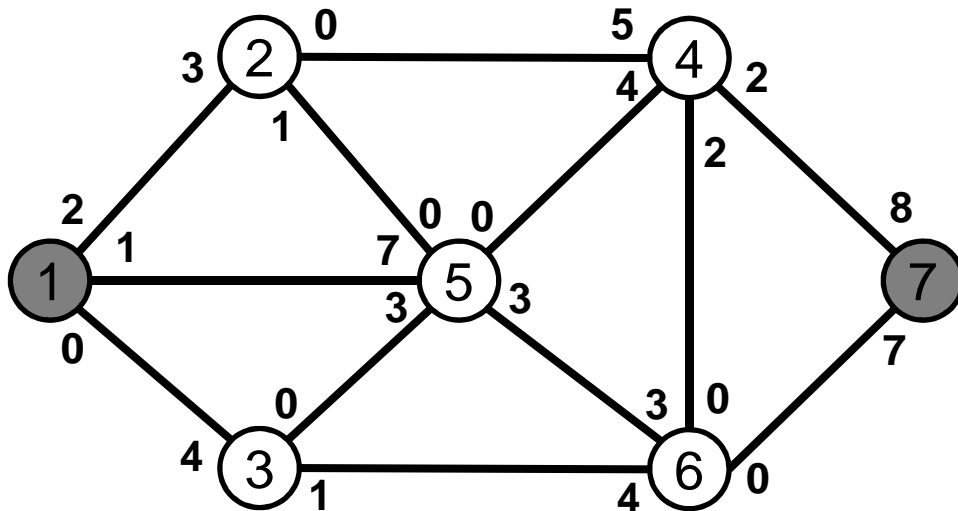
$$\Delta=1 \quad f = f+\Delta = 14$$

Algoritmo del Grafo Ausiliario



$$P = 1-5-6-4-7$$

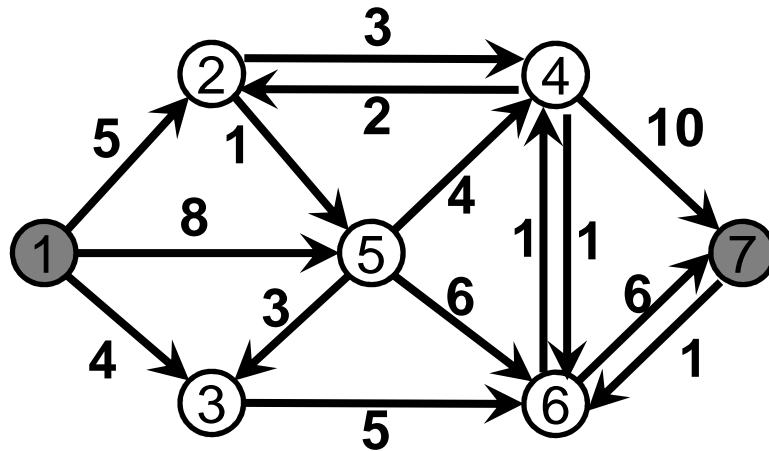
$$\Delta = 1 \quad f = f + \Delta = 14$$



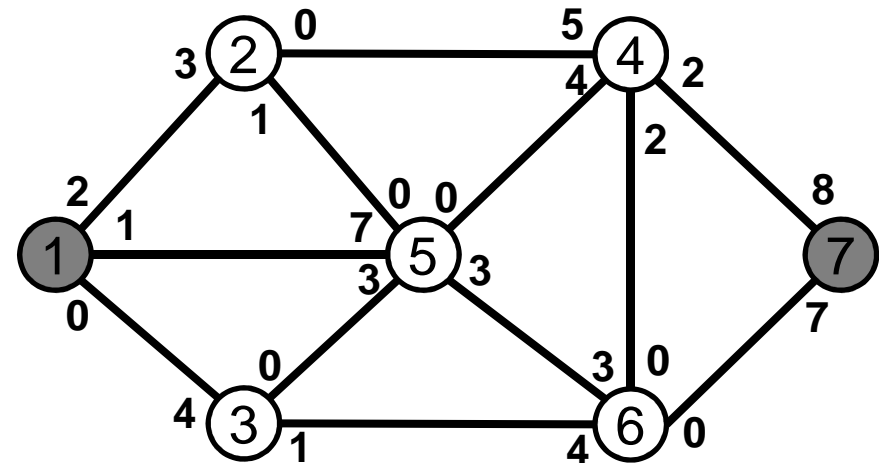
Non riesco ad individuare un cammino aumentante \rightarrow Il flusso che ho individuato è ottimo

Algoritmo del Grafo Ausiliario

Grafo iniziale



Grafo finale



Il valore delle variabili decisionali, per ogni arco del grafo di partenza, è pari alla differenza tra la capacità originale dell'arco meno quella residua nell'ultimo grafo ausiliario (il valore viene ignorato se negativo).

Ad esempio per l'arco (1,2) abbiamo una capacità iniziale pari a 5 e una finale pari a 2 quindi $x_{12}=3$.

Analogamente abbiamo

$$x_{15}=8-1=7, x_{13}=4-0=4,$$

$$x_{25}=1-1=0, x_{24}=3-0=3 \text{ ecc.ecc.}$$

Dettagli per la correttezza e l'implementazione

Si noti che nello schema generale del metodo di Ford e Fulkerson ci sono dei dettagli che devono essere meglio chiariti:

- Come si identifica un cammino aumentante o come si mostra che non esiste un cammino aumentante?
- Come certificare che il flusso ottenuto è quello massimo?

La risposta a queste domande può essere ottenuta considerando una particolare implementazione dell'algoritmo del cammino aumentante che dà luogo al ***Labeling Algorithm di Ford and Fulkerson***

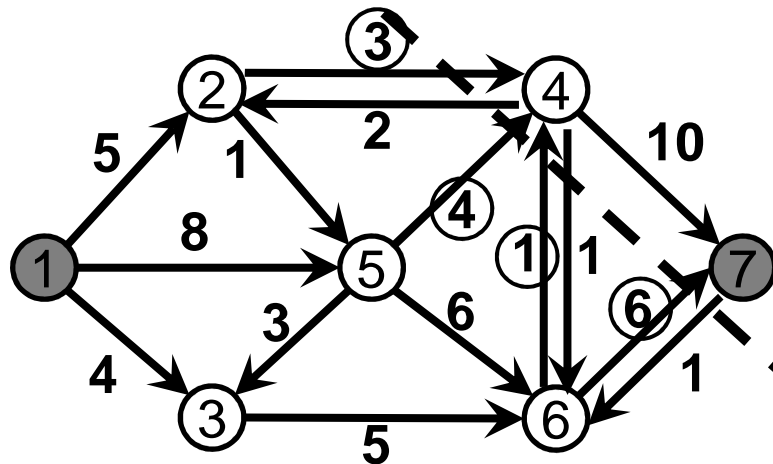
Labeling Algorithm di Ford and Fulkerson

Idea Principale:

- Per cercare un cammino si effettua una visita del grafo ausiliario a partire dalla sorgente e si etichettano (label) tutti i nodi che possono essere raggiunti
- Se il pozzo viene etichettato, allora esiste un cammino aumentante e si può incrementare il flusso da s a t attraverso il cammino trovato
- Se il pozzo non viene etichettato allora si costruisce un taglio nel seguente modo:
 - in V_1 si inseriscono i nodi etichettati
 - in V_2 si inseriscono i nodi non etichettati
- Poiché la capacità del taglio così costruito è pari al flusso f inviato fino a quel momento, dal teorema MinCut/MaxFlow tale flusso f è massimo.

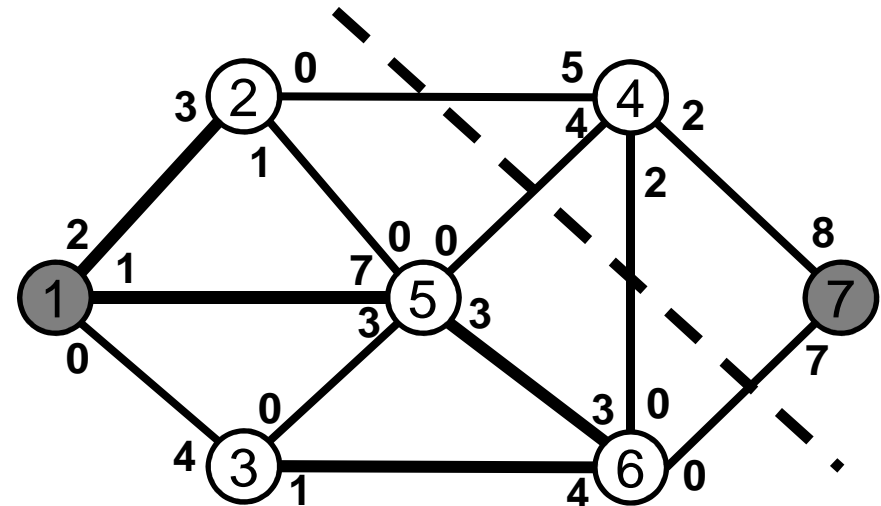
Individuazione taglio minimo

Grafo iniziale



Capacità = 14

Grafo finale



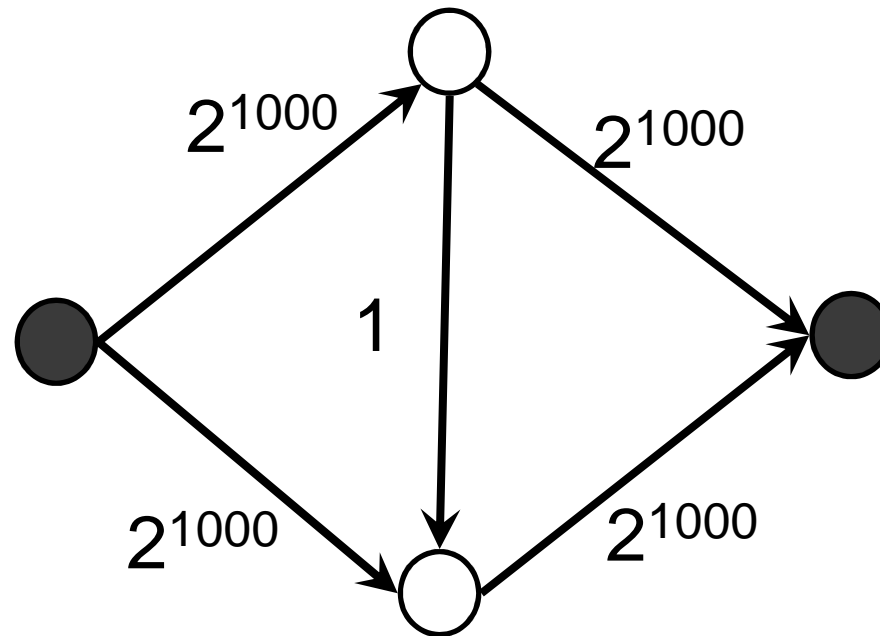
Taglio : $V_1 = \{1, 2, 3, 5, 6\}$, $V_2 = \{4, 7\}$

Per poter individuare il taglio minimo, la cui capacità sarà uguale al flusso massimo $f=14$, è sufficiente controllare quali sono i nodi raggiungibili dalla sorgente 1 attraverso archi con capacità residua >0 nell'ultimo grafo ausiliario.

L' algoritmo del cammino aumentante

1. Dato un grafo $G=(V,A,u)$:
 - 1.1 Sia \underline{X} un flusso ammissibile iniziale ($x_{ij}=0 \forall (i,j) \in A$)
 - 1.2 $f=0$
 2. Costruisci il grafo ausiliario $G(\underline{X})$
 3. Cerca in $G(\underline{X})$ un cammino aumentante p .
 - 3.1 Se non esiste alcun cammino aumentante allora STOP: il flusso corrente f è massimo
 - 3.2 Sia $\Delta = \min\{r_{ij} : (i,j) \in p\}$
Aggiorna il flusso: $f=f+\Delta$
Aggiorna le capacità residue r_{ij} degli archi di p nel seguente modo:
 - se $(i,j) \in p : r_{ij} = r_{ij} - \Delta$
 - se $(j,i) \in p : r_{ij} = r_{ij} + \Delta$
- Torna al passo 3

Complessità dell'algoritmo del grafo ausiliario



Approcci alternativi

Per migliorare la complessità dell' algoritmo ci sono diversi approcci:

- cercare un cammino con il numero minimo di archi (shortest augmenting path algorithm)
- posso cercare un cammino con una capacità almeno pari ad una quantità Δ fissata di volta in volta (capacity scaling algorithm)
- algoritmi di preflow push