Ottimizzazione Combinatoria

Corso di Laurea in Informatica

Seconda Parte: Reti di Flusso

Ugo Dal Lago

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

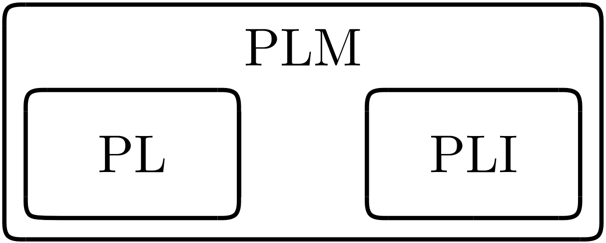
Inria informatiques mathématiques

Anno Accademico 2021-2022

Sezione 1

Reti di Flusso

Problemi su Reti



Problemi su Reti

PL

PL

Problemi su Reti

Reti — I

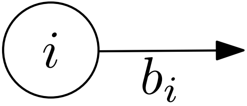
- Con il termine **rete** indichiamo un grafo $G = (N, A)$, di solito *diretto*, ai cui archi siano associati dei *pesi*.
- Gli *archi* di una rete sono interpretabili come **canali** in cui flusicono oggetti, rappresentati da grandezze:
  - **Discrete** (ad esempio passeggeri o veicoli);
  - **Continue** (ad esempio fluidi);
- I *nodi* indicano invece **punti di ingresso o di uscita** dalla rete.
- Una conseguenza di questo modo di interpretari archi e nodi è la **terminologia** che andiamo ad introdurre ora.

Reti — II

- Ad ogni nodo $i \in N$ è associato un reale b_i, detto **sbilanciamento**, che può essere:
  - *Positivo.*
    - Il nodo i è un nodo di uscita dalla rete e viene detto **destinazione**, **pozzo**, oppure **nodo di output**.
    - b_i è detto **domanda** di i.
  - *Negativo.*
    - Il nodo i è un nodo di entrata nella rete e veiene detto **origine**, **sorgente**, oppure **nodo di input**.
    - $-b_i$ è detto **offerta** di i.
  - *Nullo.*
    - Il nodo i è detto **nodo di trasferimento**.

Reti — II

- Ad ogni nodo $i \in N$ è associato un reale b_i, detto **sbilanciamento**, che può essere:
  - *Positivo.*
    - Il nodo i è un nodo di uscita dalla rete e viene detto **destinazione**, **pozzo**, oppure **nodo di output**.
    - b_i è detto **domanda** di i.
  - *Negativo.*
    - Il nodo i è un nodo di entrata nella rete e veiene detto **origine**, **sorgente**, oppure **nodo di input**.
    - $-b_i$ è detto **offerta** di i.
  - *Nullo.*
    - Il nodo i è detto **nodo di trasferimento**.

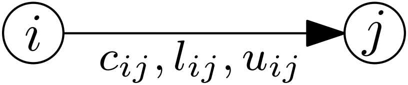


Reti — III

- Ad ogni arco $(i, j) \in A$ sono associati:
  - Un **costo** c_{ij} che indica quanto costa, per un'unità di bene, attraversare il canale.
  - Una **capacità inferiore** l_{ij}, ossia un limite inferiore alla quantità di beni che possono fluire sul canale.
  - Una **capacità superiore** u_{ij}, ossia un limite superiore alla quantità di beni che possono fluire sul canale.

Reti — III

- Ad ogni arco $(i, j) \in A$ sono associati:
  - Un **costo** c_{ij} che indica quanto costa, per un'unità di bene, attraversare il canale.
  - Una **capacità inferiore** l_{ij}, ossia un limite inferiore alla quantità di beni che possono fluire sul canale.
  - Una **capacità superiore** u_{ij}, ossia un limite superiore alla quantità di beni che possono fluire sul canale.



Problemi di Flusso

- Nei problemi di flusso, una soluzione non è altro che un **flusso**, ossia un assegnamento di *valori reali* agli archi di una certe rete $G = (N, A)$.
- Ciò viene formalizzato tramite una sequenza di variabili x_{ij}, ciascuna corrispondente ad un arco $(i, j) \in A$.
- Il **costo** di un flusso non è nient'altro che il costo complessivo di tutti i flussi presenti nella rete:

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

Problemi di Flusso — Vincoli

- **Domanda e offerta globale si equivalgono**:

$$\sum_{i \in D} b_i = -\sum_{i \in O} b_i \;\Leftrightarrow\; \sum_{i \in N} b_i = 0,$$

dove $D = \{i \in N \mid b_i > 0\}$ e $O = \{i \in N \mid b_i < 0\}$.

- **Il flusso si conserva**:

$$\sum_{(j,i) \in BS(i)} x_{ji} - \sum_{(i,j) \in FS(i)} x_{ij} = b_i \qquad i \in N,$$

dove

$$BS(i) = \{(k,i) \mid (k,i) \in A\};$$
$$FS(i) = \{(i,k) \mid (i,k) \in A\}.$$

- **Il flusso deve essere ammissibile**:

$$l_{ij} \leq x_{ij} \leq u_{ij} \qquad (i,j) \in A$$

Problemi di Flusso — Perché Studiarli?

- I problemi di flusso rappresentano un ottimo compromesso tra:
  - **Espressività**, visto che moltissimi problemi concreti si possono esprimere come istanze di problemi di flusso;
  - **Complessità**, visto che esistono algoritmi relativamente efficienti per i problemi di flusso, anche per i più generali tra essi.

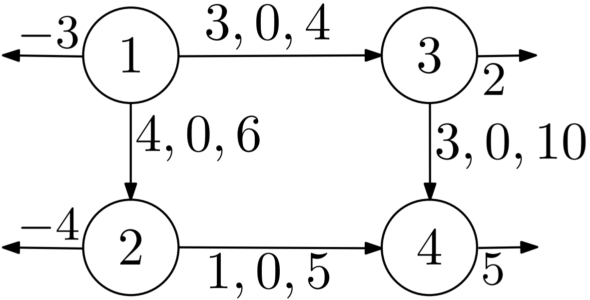
Il Problema del Flusso di Costo Minimo (MCF)

- Nel problema del *flusso di costo minimo* (o *minimum cost flow*, MCF):
  - Il costo del flusso è la funzione obiettivo, ovviamente da **minimizzare**.
  - Le capacità inferiori sono nulle.
- Il problema è formalizzabile facilmente in programmazione lineare:

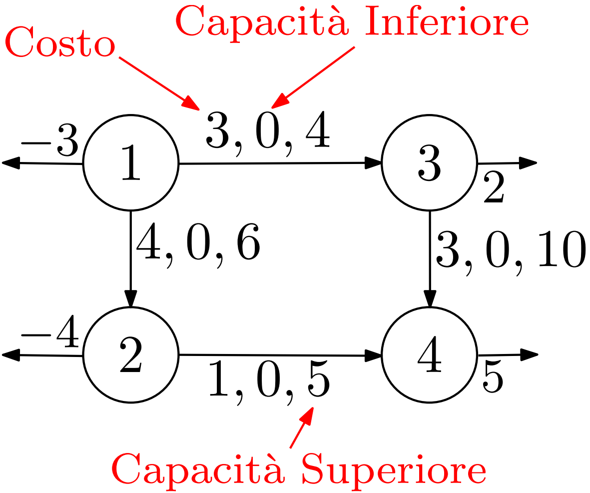
$$\min cx$$
$$0 \leq x \leq u \qquad Ex = b$$

dove
  - $c \in \mathbb{R}^{|A|}$ è il **vettore dei costi**;
  - $u \in \mathbb{R}^{|A|}$ è il **vettore delle capacità**;
  - $E \in \mathbb{R}^{|N| \times |A|}$ è una sorta di matrice di incidenza tra nodi e archi;
  - $b \in \mathbb{R}^{|N|}$ è il **vettore degli sbilanciamenti**.

MCF — Esempio



MCF — Esempio

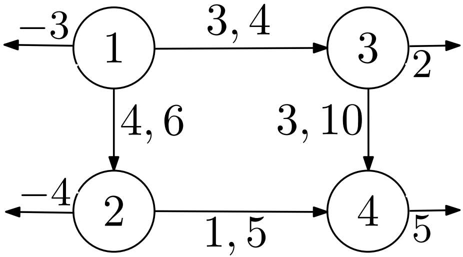


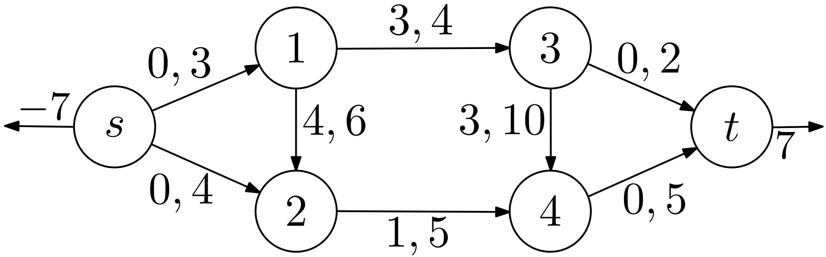
Costo
Capacità Inferiore
3, 0, 4
−3
1
3
2
4, 0, 6
3, 0, 10
−4
2
4
5
1, 0, 5
Capacità Superiore

MCF — Rilassare Alcune Assunzioni

- Spesso, nel progetto di algoritmi per MCF, conviene assumere che vi sia una sola sorgente e un solo pozzo.
- Un generico problema MCF si può trasformare in un problema con una sola sorgente e un solo pozzo:
  - Aggiungendo **due nodi** fittizi, uno corrispondente all'unica sorgente, e l'altro all'unico pozzo.
  - Aggiungendo **archi** fittizi **dalla sorgente** a ciascuna delle sorgenti della rete di partenza. Tali archi avranno costo nullo, e capacità superiore pari all'inverso dello sbilanciamento della sorgente.
  - Aggiungendo **archi** fittizi da ciascuno dei pozzi della rete di partenza **al pozzo**. Tali archi avranno anch'essi costo nullo, ma capacità superiore pari allo sbilanciamento dell pozzo.
  - Lo **sbilanciamento** dell'unica sorgente sarà uguale alla somma degli sbilanciamenti delle sorgenti della rete di partenza. Similmente per i pozzi.

MCF — Rilassare Alcune Assunzioni





-7
s
$0, 3$
$0, 4$
1
2
$3, 4$
$4, 6$
$1, 5$
3
4
$3, 10$
$0, 2$
$0, 5$
t
7

MCF — Rilassare Alcune Assunzioni

- Un'altra supposizione che seplifica un po' il problema consiste nell'imporre che $l_{ij} = 0$ per ogni arco $(i, j) \in A$, ossia che **le capacità inferiori siano nulle**.
  - La faremo sempre anche noi!
- Data una rete G, si può costruire una rete H che sia in un certo senso **equivalente** a G ma che abbia capacità inferiori nulle. Per ogni arco $(i, j) \in A$,
  - Si **sottrae** la quantità l_{ij} a b_j e a u_{ij};
  - Si **aggiunge** la quantità l_{ij} a b_i.
  - Occorrerà aggiungere la quantità

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

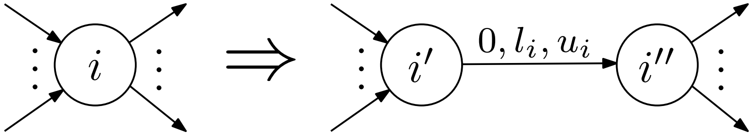
    alla funzione obiettivo.
- In altre parole, ad un flusso x_{ij} in H corrisponderà un flusso $x_{ij} + l_{ij}$ in G.

MCF — Rilassare Alcune Assunzioni

- Alcune volte è utile imporre che anche i *nodi* (e non solo gli *archi*) abbiano delle **capacità**, ossia che solo una quantità di flusso compresa nell'intervallo chiuso $[l_i, u_i]$ possa passare per il nodo $i \in N$.
- Situazioni come queste si possono modellare **sdoppiando** ciascun nodo i in due nodi i', i'', in modo che:
  - Tutti gli archi entranti in i vadano a finire in i'.
  - Tutti gli archi uscenti da i partano da i''.
  - Vi sia un arco fittizio (i', i'') con costo nullo, capacità inferiore l_i e capacità superiore u_i.

MCF — Rilassare Alcune Assunzioni

- Alcune volte è utile imporre che anche i *nodi* (e non solo gli *archi*) abbiano delle **capacità**, ossia che solo una quantità di flusso compresa nell'intervallo chiuso $[l_i, u_i]$ possa passare per il nodo $i \in N$.
- Situazioni come queste si possono modellare **sdoppiando** ciascun nodo i in due nodi i', i'', in modo che:
  - Tutti gli archi entranti in i vadano a finire in i'.
  - Tutti gli archi uscenti da i partano da i''.
  - Vi sia un arco fittizio (i', i'') con costo nullo, capacità inferiore l_i e capacità superiore u_i.



Sezione 2

Il Problema del Flusso Massimo

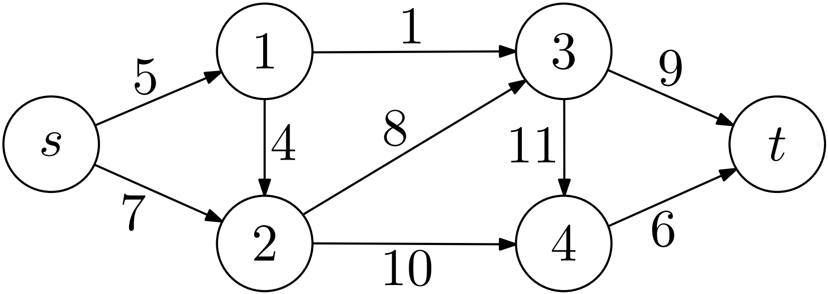
Definire il Problema — I

- Il **problema di flusso massimo** (o maximum flow, MF) è un problema di ottimizzazione su reti che può essere visto come una restrizione di MCF.
- Ciò che cambia è la funzione obiettivo: non vogliamo *minimizzare* i costi, ma piuttosto *massimizzare* i flussi.
- Formalmente, data una rete $G = (N, A)$:
  - Fissiamo due nodi s (detto **sorgente**) e t (detto **destinazione**);
  - Vogliamo massimizzare il flusso da s a t, ossia trovare il **massimo** valore v tale che se $b_s = -v$, $b_t = v$ e $b_i = 0$ in tutti gli altri casi, allora esiste un flusso ammissibile (di qualsiasi costo).
- Un valore v *ammissibile* per il problema di cui sopra si dice **valore** del flusso x.
- Il problema è formalizzabile direttamente in PL.

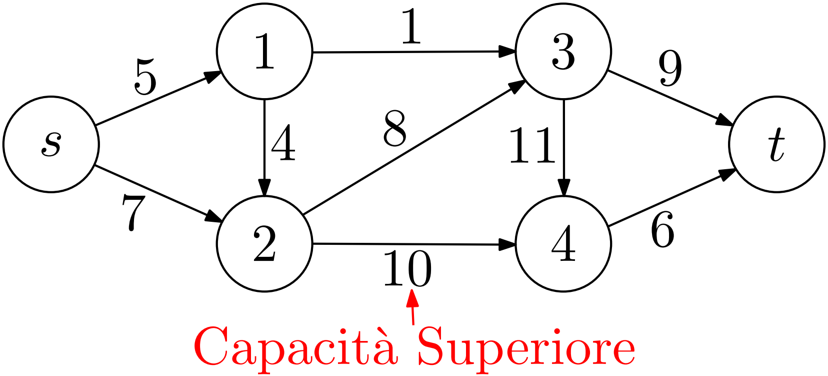
Definire il Problema — II

$$
\begin{aligned}
&\max v \\
&\sum_{(j,s)\in BS(s)} x_{js} + v = \sum_{(s,j)\in FS(s)} x_{sj}; \\
&\sum_{(j,i)\in BS(i)} x_{ji} - \sum_{(i,j)\in FS(i)} x_{ij} = 0, \quad i \in N - \{s,t\}; \\
&\sum_{(j,t)\in BS(t)} x_{jt} = \sum_{(t,j)\in FS(t)} x_{tj} + v; \\
&0 \le x_{ij} \le u_{ij}, \quad (i,j) \in A.
\end{aligned}
$$

Esempio



Esempio

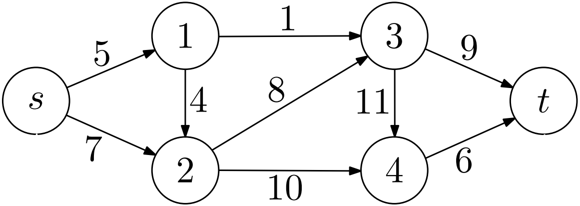


Massimo Flusso e MCF

- Il problema MF può essere visto come un caso particolare di MCF:
  - I costi sono nulli;
  - Gli sbilanciamenti sono nulli;
  - Si aggiunge però un arco fittizio da t a s con costo -1 e capacità infinita.

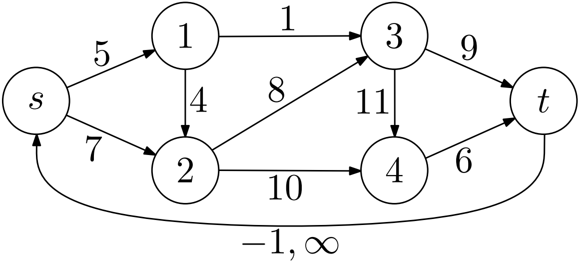
Massimo Flusso e MCF

- Il problema MF può essere visto come un caso particolare di MCF:
  - I costi sono nulli;
  - Gli sbilanciamenti sono nulli;
  - Si aggiunge però un arco fittizio da t a s con costo -1 e capacità infinita.
- Graficamente:



Massimo Flusso e MCF

- Il problema MF può essere visto come un caso particolare di MCF:
  - I costi sono nulli;
  - Gli sbilanciamenti sono nulli;
  - Si aggiunge però un arco fittizio da t a s con costo -1 e capacità infinita.
- Graficamente:

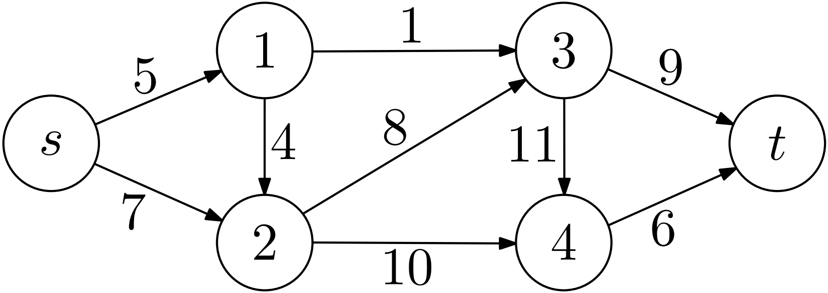


Tagli

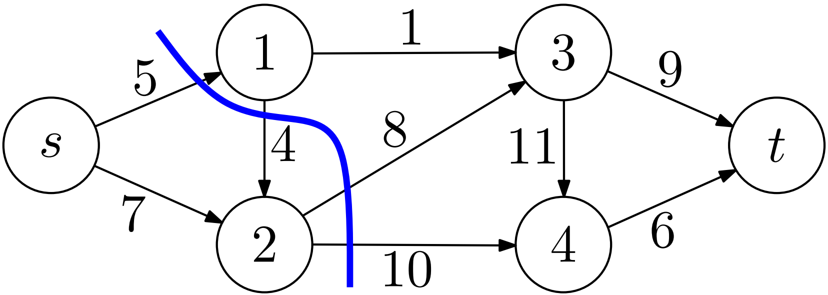
- Un **taglio** in una rete $G = (N, A)$ e dato da una coppia (N', N'') di sottoinsiemi di N tali che $N' \cap N'' = \emptyset$ e $N' \cup N'' = N$.
- Un (s,t)**-taglio** in una rete G è un taglio (N_s, N_t) dove $s \in N_s$ e $t \in N_t$.
- Dato un (s,t)-taglio in $G = (N, A)$, indichiamo con $A^+(N_s, N_t)$ e $A^-(N_s, N_t)$ i seguenti sottoinsiemi di A:

$$A^+(N_s, N_t) = \{(i,j) \in A \mid i \in N_s \land j \in N_t\};$$
$$A^-(N_s, N_t) = \{(i,j) \in A \mid i \in N_t \land j \in N_s\}.$$

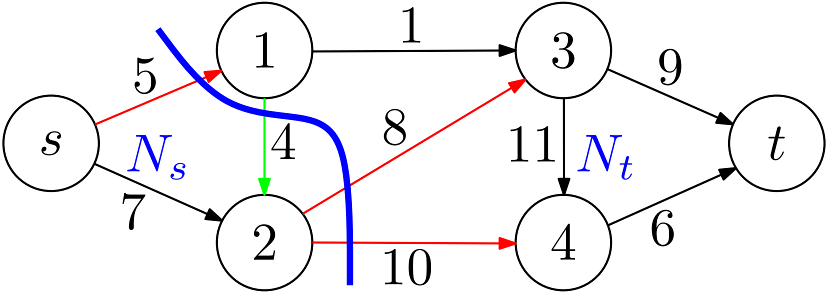
Tagli — Esempio



Tagli — Esempio



Tagli — Esempio



Tagli — Proprietà

Lemma

Per ogni (s,t)*-taglio* (N_s, N_t) *e ogni flusso ammissibile* x *con valore* v*:*

1. $v = \sum_{(i,j)\in A^+(N_s,N_t)} x_{ij} - \sum_{(i,j)\in A^-(N_s,N_t)} x_{ij}$;
2. $v \leq \sum_{(i,j)\in A^+(N_s,N_t)} u_{ij}$.

Dimostrazione.

1. Osserviamo che:

$$
\begin{aligned}
v &= \sum_{(s,j)\in A} x_{sj} - \sum_{(i,s)\in A} x_{is} = \sum_{k\in N_s} \left(\sum_{(k,j)\in A} x_{kj} - \sum_{(i,k)\in A} x_{ik} \right) \\
&= \sum_{(i,j)\in A^+(N_s,N_t)} x_{ij} - \sum_{(i,j)\in A^-(N_s,N_t)} x_{ij}
\end{aligned}
$$

2. E' una semplice conseguenza del punto 1.

$\square$

Tagli — Proprietà

- Diamo un nome alle due quantità che abbiamo studiato nel Lemma precedente:
  - La quantità $\sum_{(i,j)\in A^+(N_s,N_t)} x_{ij} - \sum_{(i,j)\in A^-(N_s,N_t)} x_{ij}$ è detta **flusso del taglio** (N_s, N_t), ed è indicata con $x(N_s, N_t)$.
  - La quantità $\sum_{(i,j)\in A^+(N_s,N_t)} u_{ij}$ è detta **capacità del taglio** (N_s, N_t), ed è indicata con $u(N_s, N_t)$.
- Quello che ci dice il Lemma precedente è che

$$v = x(N_s, N_t) \leq u(N_s, N_t).$$

- In altre parole: *il valore di un flusso ammissibile è sempre minore o uguale della capacità di qualunque taglio.*
- Ma esiste un taglio con capacità **identica** al valore di un flusso ammissibile (che quindi sarà massimo)?

Grafi Residui

- Data una rete $G = (N_G, A_G)$ e un flusso ammissibile x, il **grafo residuo** G_x è il *multigrafo* (N_{G_x}, A_{G_x}) tale che:
  - $N_{G_x} = N_G$;
  - Gli archi in A_{G_x} sono di due tipi:
    - Per ogni arco $(i, j) \in A$ tale che $x_{ij} < u_{ij}$ esiste un arco **da** i **a** j in G_x (detto **arco concorde**);
    - Per ogni arco $(i, j) \in A$ tale che $x_{ij} > 0$ esiste un arco **da** j **a** i in G_x (detto **arco discorde**).
  - Osserviamo come in N_{G_x} ci possano essere *due* archi da uno stesso nodo i ad uno stesso nodo j.

Cammini Aumentanti

- Un **cammino aumentante** P in una rete G rispetto a x non è nient'altro che un cammino semplice e orientato da s a t in G_x.
  - Sia P^+ l'insieme degli archi *concordi* in P, e P^- l'insieme dei suoi archi *discordi*.
- Dato un cammino aumentante P rispetto a x, definiamo la **capacità** di P rispetto a x come

$$\begin{aligned}\theta(P,x) = \min\{ & \min\{u_{ij} - x_{ij} \mid (i,j) \in P^+\}, \\ & \min\{x_{ij} \mid (j,i) \in P^-\}\}.\end{aligned}$$

- Dato un flusso x, un cammino P in G_x e un reale θ, definiamo $x(P,\theta)$ il flusso definito come segue:

$$(x(P,\theta))_{ij} = \begin{cases} x_{ij} + \theta & \text{se } (i,j) \in P^+; \\ x_{ij} - \theta & \text{se } (j,i) \in P^-; \\ x_{ij} & \text{altrimenti.} \end{cases}$$

L'Algoritmo di Ford-Fulkerson

1. $x \leftarrow 0$;
2. Costruisci G_x e determina se G_x ha un cammino aumentante P. In caso P non esista, termina e restituisci x;
3. $x \leftarrow x(P, \theta(P, x))$
4. Ritorna al punto 2.

Ford-Fulkerson — Correttezza

Lemma

Se x è ammissibile, allora anche $x(P, \theta(P, x))$ è ammissibile.

Ford-Fulkerson — Correttezza

Lemma

Se x è ammissibile, allora anche $x(P, \theta(P, x))$ è ammissibile.

Lemma

Se x è un flusso ammissibile massimo, allora G_x non ha cammini aumentanti.

Dimostrazione.

Se ci fosse un cammino aumentante in G_x, allora x non sarebbe massimo, perché sarebbe possibile aumentare il valore del flusso. $\square$

Ford-Fulkerson — Correttezza

Lemma

Se G_x non ha cammini aumentanti, allora esiste un taglio di capacità pari a v.

Dimostrazione.

Basta considerare Il taglio (N_s, N_t), dove N_s contiene tutti e soli i nodi raggiungibili da s in G_x (e $N_t = N - N_s$). Infatti

$$\begin{aligned} v = x(N_s, N_t) &= \sum_{(i,j)\in A^+(N_s,N_t)} x_{ij} - \sum_{(i,j)\in A^-(N_s,N_t)} x_{ij} \\ &= \sum_{(i,j)\in A^+(N_s,N_t)} u_{ij} - \sum_{(i,j)\in A^-(N_s,N_t)} 0 \\ &= u(N_s, N_t) \end{aligned}$$

$\square$

Ford-Fulkerson — Correttezza

Teorema (Correttezza)

Se l'algoritmo di Ford-Fulkerson termina, allora il flusso x è un flusso massimo.

Dimostrazione.

- Se Ford-Fulkerson termina, allora G_x non ha cammini aumentanti.
- Ma quindi esiste un taglio di capacità pari a v.
- E a questo punto v non può essere che massimo, perché se non lo fosse avremmo un taglio di capacità inferiore al valore di un flusso ammissibile.

□

Max-Flow Min-Cut

Teorema

Il valore del massimo flusso è uguale alla minima capacità dei tagli.

Dimostrazione.

- Basta dimostrare che il valore del massimo flusso è maggiore o uguale alla capacità di *un* taglio.
- Ma se x è ammissibile e massimo, G_x non ha cammini aumentanti, e quindi esiste un taglio di capacità pari a v.

$\square$

Ford-Fulkerson — Complessità

Teorema

Se le capacità di G sono numeri interi, allora esiste almeno un flusso intero massimo.

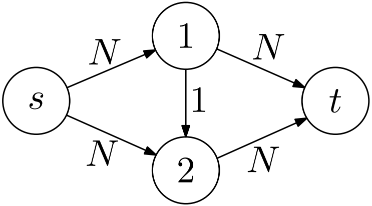
Dimostrazione.

- Se le capacità sono intere, allora il flusso massimo sarà al più nU dove $U = \max\{u_{ij} \mid (i,j) \in A\}$.
- Si parte da un flusso intero, e l'interezza è preservata, perché per ogni cammino aumentante P, $\theta(P,x)$ è un numero intero.
- Di conseguenza, il valore del flusso aumenta almeno di 1 ad ogni iterata.
- L'algoritmo terminerà quindi dopo al più nU iterate.

$\square$

Ford-Fulkerson — Complessità

- La dimostrazione del teorema precedente ci dice che se le capacità sono intere, allora la complessità è $O(mnU)$, ovvero solo **pseudopolinomiale** nella dimensione della rete
- Controesempio alla polinomialità:



- In assenza del vincolo di interezza, non si può dire molto sulla complessità dell'algoritmo.

L'Algoritmo di Edmonds-Karp

- E' possibile rendere la complessità di Ford-Fulkerson **polinomiale** in tempo?
- Un modo consiste nel lasciare l'algoritmo così com'è, ma agire sul **modo** in cui i cammini aumentanti vengono scoperti, ossia sull'algoritmo usato per **visitare** il grafo residuo G_x.
- L'**Algoritmo di Edmonds-Karp** non è nient'altro che l'algoritmo di Ford-Fulkerson dove, però, la ricerca del cammino aumentante viene eseguita visitando **in ampiezza** (BFS) il grafo residuo G_x.
- Notiamo che in questo modo i cammini aumentanti saranno sempre cammini **di lunghezza minima**.

Edmonds-Karp — Proprietà

- EK è trivialmente **corretto**, essendo nient'altro che un caso particolare di FF.
- Studiarne la **complessità**, invece, risulta più difficile:
  - Si procede osservando che, *se in FF i cammini aumentanti sono di lunghezza minima*, allora la distanza di un generico nodo i dalla sorgente s in G_x **non può diminuire**.
  - Da ciò si deduce che il numero di iterazioni di EK non può, asintoticamente essere più grande di $N \cdot A$.

Edmonds-Karp — Proprietà

- Data una rete G, un flusso ammissibile x e due nodi i, j in G, indichiamo con $\delta_x(i, j)$ la distanza tra i e j nel grafo residuo G_x.

Edmonds-Karp — Proprietà

- Data una rete G, un flusso ammissibile x e due nodi i, j in G, indichiamo con $\delta_x(i, j)$ la distanza tra i e j nel grafo residuo G_x.

Lemma

Se, durante l'esecuzione di EK, il flusso y è ottenuto da x tramite un'operazione di aumento del flusso in un cammino aumentante, allora per ogni nodo $i \in N$, vale che $\delta_x(s, i) \leq \delta_y(s, i)$.

Edmonds-Karp — Proprietà

- Data una rete G, un flusso ammissibile x e due nodi i, j in G, indichiamo con $\delta_x(i, j)$ la distanza tra i e j nel grafo residuo G_x.

Lemma

Se, durante l'esecuzione di EK, il flusso y è ottenuto da x tramite un'operazione di aumento del flusso in un cammino aumentante, allora per ogni nodo $i \in N$, vale che $\delta_x(s, i) \leq \delta_y(s, i)$.

- La prova è molto interessante ed ingegnosa. Non abbiamo purtroppo tempo di vederla in dettaglio. Gli studenti interessati possono consultare le note.

Edmonds-Karp — Proprietà

Teorema

Il numero di iterazioni di EK è $O(NA)$, quindi la sua complessità è $O(NA^2)$.

Dimostrazione.

- Un arco (i, j) in G_x è detto *critico* per un cammino aumentante P se la sua capacità (ossia $u_{ij} - x_{ij}$ se è concorde o x_{ji} se discorde) è uguale a $\theta(P, x)$.
  - Dopo l'aumento del flusso lungo P, l'arco (i, j) sparisce dal grafo residuo.
  - In ogni cammino aumentante, esiste almeno un arco critico.
- Dati i, j connessi da un arco in A, quante volte è possibile che (i, j) sia arco critico? Si può dimostrare che tale numero è al più $O(N)$, e visto che di tali coppie ne esistono al più $O(A)$, in totale potremo avere al più $O(NA)$ iterazioni. Nel seguito dimostriamo proprio il limite $O(N)$ al numero di volte in cui (i, j) può diventare critico.

- Quando (i,j) diventa critico la prima volta, deve valere che

$$\delta_x(s,j) = \delta_x(s,i) + 1,$$

dove x è il flusso, e a quel punto sparisce dal grafo residuo.

- L'unico modo per ricomparirvi è fare in modo che il flusso (reale o virtuale) da i a j *diminuisca*, e questo vuol dire che

$$\delta_y(s,i) = \delta_y(s,j) + 1,$$

dove y è il flusso.

- Dunque:

$$\delta_y(s,i) = \delta_y(s,j) + 1 \geq \delta_x(s,j) + 1 = \delta_x(s,i) + 2.$$

- Di conseguenza, da un momento in cui (i,j) diventa critico al successivo, la sua distanza da s aumenta di almeno 2. Siccome tale distanza non può essere superiore a $|N|$, il numero di volte in cui (i,j) può diventare critico è lineare in $|N|$.

$\square$

Ci Sono Solo FF e EK?

Ci Sono Solo FF e EK?

Algoritmo	*Complessità*
Dinic	$O(V^2 A)$
MKM	$O(V^3)$
Goldberg-Tarjan	$O(V^2 A)$
KRT	$O(VA \log_{\frac{A}{V \log V}} V)$
Orlin+KRT	$O(VA)$

L'Algoritmo di Goldberg-Tarjan — I

- Una delle strade che ci permettono di scendere sotto la soglia di una complessità $O(VA^2)$ è quella che Goldberg e Tarjan intrapresero nel 1986 e che consiste nel rendere la costruzione del flusso massimo più **locale**.
  - Questo in opposizione con FF ed EK, in cui ad ogni iterazione occorre procedere con un'analisi **globale**.

L'Algoritmo di Goldberg-Tarjan — I

- Una delle strade che ci permettono di scendere sotto la soglia di una complessità $O(VA^2)$ è quella che Goldberg e Tarjan intrapresero nel 1986 e che consiste nel rendere la costruzione del flusso massimo più **locale**.
  - Questo in opposizione con FF ed EK, in cui ad ogni iterazione occorre procedere con un'analisi **globale**.
- C'è un prezzo da pagare, ossia il fatto che durante l'esecuzione dell'algoritmo, non si lavora solo con flussi **ammissibili**, ma anche con **preflussi**.

L'Algoritmo di Goldberg-Tarjan — II

- Un **preflusso** è un vettore x tale che:

$$\sum_{(j,i)\in BS(i)} x_{ji} - \sum_{(i,j)\in FS(i)} x_{ij} \geq 0, \quad i \in N - \{s,t\};$$
$$0 \leq x_{ij} \leq u_{ij}, \quad (i,j) \in A.$$

  In altre parole, i vincoli di capacità sono soddisfatti, mentre quelli di bilanciamento ai nodi possono non esserlo.

- Un nodo si dice **attivo** se il suo **eccesso**

$$e_i = \sum_{(j,i)\in BS(i)} x_{ji} - \sum_{(i,j)\in FS(i)} x_{ij}$$

  è positivo, altrimenti si dice **bilanciato**.

L'Algoritmo di Goldberg-Tarjan — III

- L'idea centrale dell'algoritmo è quella di eliminare gli sbilanciamenti presenti nel preflusso corrente in modo *iterativo* e *locale*.

L'Algoritmo di Goldberg-Tarjan — III

- L'idea centrale dell'algoritmo è quella di eliminare gli sbilanciamenti presenti nel preflusso corrente in modo *iterativo* e *locale*.
- Lo sbilanciamento presente in un nodo i viene eliminato spostando parte del flusso in eccesso:
  - In avanti, ossia attraverso un arco (i, j), ogniqualvolta $x_{ij} < u_{ij}$
  - Indietro, ossia attraverso un arco (j, i), ogniqualvolta $x_{ij} > 0$.

  Tali operazioni si chiamano **push forward** e **push backward**.

L'Algoritmo di Goldberg-Tarjan — III

- L'idea centrale dell'algoritmo è quella di eliminare gli sbilanciamenti presenti nel preflusso corrente in modo *iterativo* e *locale*.
- Lo sbilanciamento presente in un nodo i viene eliminato spostando parte del flusso in eccesso:
  - In avanti, ossia attraverso un arco (i, j), ogniqualvolta $x_{ij} < u_{ij}$
  - Indietro, ossia attraverso un arco (j, i), ogniqualvolta $x_{ij} > 0$.

  Tali operazioni si chiamano **push forward** e **push backward**.
- E' sempre possibile effettuare operazioni di push?
  - **No**, altrimenti sarebbe molto difficile tenere la complessità dell'algoritmo sotto controllo.
  - Si usa invece un sistema basato su **etichettature**.

GoldbergTarjan(G, s, t)

1. $x \leftarrow 0$;
2. $x_{sj} \leftarrow u_{sj} \quad \forall (s, j) \in FS(s)$;
3. $d \leftarrow$ EtichettaturaValida(G);
4. $d_s \leftarrow n$;
5. Se tutti i nodi (diversi da s e t) sono bilanciati, allora termina e restituisci x.
6. Sia v un qualunqe nodo sbilanciato
7. Se esiste (v, j) ammissibile per v, allora esegui PushForward(v, j) e torna a 6, altrimenti prosegui
8. se esiste (i, v) ammissibile per v, allora esegui PushBackward(i, v) e torna a 6, altrimenti prosegui
9. esegui Relabel(v) e torna a 6.

Goldberg-Tarjan — IV

Teorema

L'algoritmo di Goldberg e Tarjan è corretto, e la sua complessità in tempo è $O(N^2 A)$.

Sezione 3

Il Problema del Flusso di Costo Minimo

Nozioni e Risultati Preliminari — I

- Uno **pseudoflusso** è un vettore x che soddisfa i vincoli di capacità, ossia tale che

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

- Se x è uno pseudoflusso, definiamo **sbilanciamento** di un nodo i rispetto a x la quantità

$$e_x(i) = \sum_{(j,i) \in BS(i)} x_{ji} - \sum_{(i,j) \in FS(i)} x_{ij} - b_i$$

- Possiamo anche vedere e_x come un vettore, detto **vettore degli sbilanciamenti**.

Nozioni e Risultati Preliminari — II

- Dato uno pseudoflusso x, i nodi sbilanciati rispetto a x fanno parte di uno dei seguenti due insiemi:

$$O_x = \{i \in N \mid e_x(i) > 0\};$$
$$D_x = \{i \in N \mid e_x(i) < 0\}.$$

  I nodi in O_x sono detti **nodi con eccesso di flusso**, mentre quelli in D_x sono detti **nodi con difetto di flusso**.

- Se $O_x = D_x = \emptyset$, allora x è un flusso.
- Lo **sbilanciamento complessivo** di x è definito come:

$$g(x) = \sum_{i \in O_x} e_x(i) = -\sum_{j \in D_x} e_x(j)$$

Cammini Aumentanti — I

- Quando si lavora con pseudoflussi, la nozione stessa di cammino aumentante diventa più generale.
- La nozione di **grafo residuo** G_x per uno pseudoflusso x si generalizza banalmente al problema MCF. Ogni arco, però, avrà ora un costo, parametro importante.
  - in un arco concorde (i, j) di G_x, il costo è semplicemente c_{ij};
  - in un arco discorde (j, i) di G_x, il costo è invece $-c_{ij}$.
- Un cammino P tra i e j in G_x verrà quindi detto **cammino aumentante** tra i e j.
  - I suoi archi possono essere partizionati in P^+ e P^-.
  - La sua capacità $\theta(P, x)$ è definita come al solito.
  - Un cammino aumentante tra i e i viene anche detto **ciclo aumentante**

Cammini Aumentanti — II

- Dato uno pseudoflusso x e un cammino aumentante P, è possibile inviare $0 \leq \theta \leq \theta(P, x)$ unità di flusso lungo P, attraverso l'operazione $x(P, \theta)$, che conosciamo già
  - In questo contesto, $x(P, \theta)$ verrà spesso indicato anche con $x \oplus P\theta$.
- Se P è un cammino aumentante da i a j in G_x, allora lo pseudoflusso $x(P, \theta)$ avrà gli stessi sbilanciamenti di x, tranne in i e in j.
  - Se $i = j$, allora il vettore degli sbilanciamenti resterà addirittura *inalterato*.
- Il **costo** di un cammino aumentante P è definito come

$$c(P) = \sum_{(i,j) \in P^+} c_{ij} - \sum_{(i,j) \in P^-} c_{ij}$$

- Si verifica facilmente che

$$c \cdot (x(P, \theta)) = c \cdot (x \oplus P\theta) = c \cdot x + \theta c(P).$$

Teorema (Struttura degli Pseudoflussi)

Siano x e y due pseudoflussi qualunque. Allora esistono $k \leq n + m$ cammini aumentanti $P_1, \ldots, P_k$, tutti per x, di cui al più m sono cicli, tali che

$$\begin{aligned} z_1 &= x \\ z_{i+1} &= z_i \oplus \theta_i P_i \quad 1 \leq i \leq k \\ z_{k+1} &= y \\ 0 &\leq \theta_i \leq \theta(P_i, z_i). \end{aligned}$$

Inoltre, tutti i P_i hanno come estremi dei nodi in cui lo sbilanciamento di x è diverso da quello di y.

Pseudoflussi Minimali — I

- A differenza di quello che succede in MF, in MCF non possiamo permetterci di aumentare il flusso indiscriminatamente.
- Un problema centrale è quindi quello di determinare *quali siano* le operazioni di aumento lecite e quali siano le proprietà sui flussi che esse garantiscano.
- Centrale da questo punto di vista è la nozione di **pseudoflusso minimale**, che è uno pseudoflusso x che abbia costo *minimo tra tutti* gli pseudoflussi aventi lo stesso vettore di sbilanciamento e_x.

Pseudoflussi Minimali — II

Lemma

Uno pseudoflusso (rispettivamente, un flusso ammissibile) è minimale (rispettivamente, ottimo) sse non esistono cicli aumentanti di costo negativo.

Dimostrazione.

$\Longrightarrow$ Per contrapposizione: se esite un ciclo aumentante di costo negativo in G_x, applicarlo fa diminuire il costo senza alterare lo sbilanciamento, in contraddizione con la minimalità di x.

$\Longleftarrow$ Ancora per contrapposizione: supponiamo che x non sia minimale, ossia che esista y con $cy < cx$ e $e_y = e_x$. Allora per il teorema sugli pseudoflussi possiamo scrivere $y = x \oplus \theta_1 P_1 \oplus \ldots \oplus \theta_n P_n$, dove $\theta_i > 0$ e ciascun P_i è un ciclo. Da $cy < cx$ discende però che:

$$cx > cx + \theta_1 c(P_1) + \ldots + \theta_n c(P_n)$$

e quindi che $c(P_i) < 0$ per qualche i.

$\square$

Pseudoflussi Minimali — III

Teorema

Sia x uno pseudoflusso minimale e sia P un cammino aumentante rispetto ad x avente costo minimo tra tutti i cammini che uniscono un nodo di O_x ad un nodo di D_x. Allora, qualunque sia $\theta \leq \theta(x, P)$, abbiamo che $x(\theta, P) = x \oplus \theta P$ è ancora pseudoflusso minimale.

Dimostrazione.

- Siano s e t i vertici che P collega. Supponiamo che $\theta \leq \theta(x, P)$ e che y sia un qualunque pseudoflusso con vettore di sbilanciamento $e_{x(\theta,P)}$.
- Per il Teorema sulla struttura degli pseudoflussi esistono:
  - k cammini aumentanti $P_1, \ldots, P_k$ rispetto a x, tutti da s a t;
  - h cicli aumentanti $C_1, \ldots, C_h$ rispetto a x.

  tali che $y = x \oplus \theta_1 P_1 \oplus \ldots \oplus \theta_k P_k \oplus \mu_1 C_1 \oplus \ldots \oplus \mu_h C_h$, (dove tutti gli θ_i, μ_j sono positivi).

- Deve essere , per ragioni che hanno a che fare con lo sbilanciamento, che $\sum_{1 \le i \le k} \theta_i = \theta$.
- Poiché x è minimale, $c(C_i) \ge 0$.
- Siccome P ha costo minimo, $c(P_i) \ge c(P)$.
- Di conseguenza:

$$\begin{aligned} cy &= cx + \theta_1 c(P_1) + \ldots + \theta_k c(P_k) + \mu_1 c(C_1) + \ldots + \mu_h c(C_h) \\ &\ge cx + \theta c(P) = cx(\theta, P). \end{aligned}$$

$\square$

Alcuni Algoritmi Ausiliari

- È abbastanza facile costruire uno pseudoflusso minimale x, se non si bada agli sbilanciamenti. Ad esempio, il flusso x definito come

$$x_{ij} = \begin{cases} 0 & \text{se } c_{ij} \geq 0 \\ u_{ij} & \text{altrimenti} \end{cases}$$

  In questo modo i costi degli archi in G_x sono tutti non-negativi e quindi G_x non avrà cicli negativi (e in ultima analisi x sarà minimale).
- Determinare un cammino di costo minimo tra O_x e D_x è semplice: basta usare uno degli algoritmi basati sui cammini minimi.

$\textsc{CamminiMinimiSuccessivi}(G)$

1. $x \leftarrow \textsc{PseudoflussoMinimale}(G)$;
2. Se $g(x) = 0$, allora termina e restituisci x;
3. Cerca un cammino di costo minimo P tra un nodo di $i \in O_x$ e $j \in D_x$; se non esiste, termina: il problema è vuoto;
4. $x \leftarrow x(P, \min\{\theta(P,x), e_x(i), -e_x(j)\})$;
5. Torna al punto 2.

Cammini Minimi Successivi — Correttezza e Terminazione

- Ad ogni passo, il flusso x rimane minimale.
- **Se l'algoritmo termina**, allora $g(x) = 0$ e quindi x è uno pseudoflusso minimale con sbilanciamento nullo, ossia un flusso di costo minimo.
- Riguardo la **terminazione**, se b e u sono vettori di numeri *interi*, possiamo osservare che:
  - Lo pseudoflusso iniziale è esso stesso intero;
  - Se x è pseudoflusso intero, allora la capacità $\theta(x, P)$ rimane intera;
  - Lo pseudoflusso x rimane quindi *sempre* intero.
  - Ad ogni passo, $g(x)$ diminuisce di almeno 1.

Cammini Minimi Successivi — Complessità

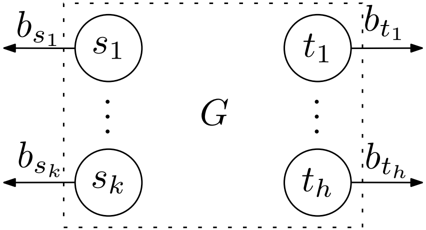
- L'analisi della terminazione che abbiamo appena fatto si estende facilmente anche alla complessità.
- Lo sbilanciamento iniziale $\overline{g}$ è al più

$$\overline{g} \leq \sum_{b_i > 0} b_i + \sum_{c_{ij} < 0} u_{ij}$$

- Come già detto, lo sbilanciamento $g(x)$ cala di almeno 1 ad ogni interazione. Le **iterazioni** saranno quindi al più $\overline{g}$.
- Il costo computazionale di **ogni iterazione** è dominato dalla ricerca di un cammino minimo, che possiamo eseguire in tempo $O(NA)$.
- La complessità sarà quindi nel caso peggiore $O(\overline{g}NA)$, **pseudopolinomiale** nella dimensione del grafo.

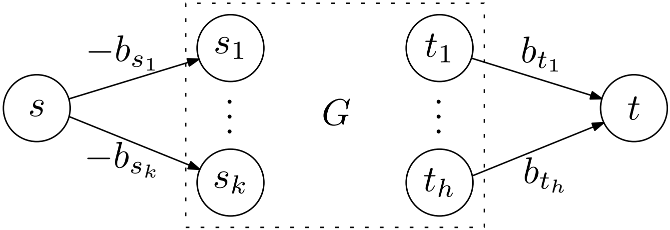
Costruire un Flusso Ammissibile

- Data una rete G, è possibile determinare se esiste un **flusso** ammissibile (ma non necessariamente ottimo) per essa?
- Basta risolvere il problema MF sulla rete ottenuta nel modo seguente



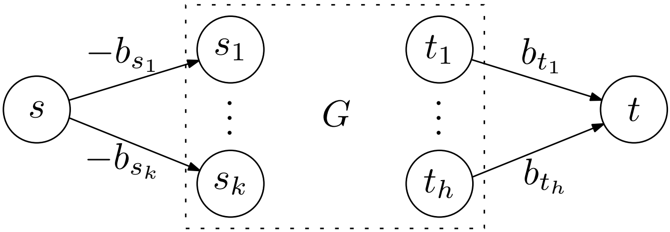
Costruire un Flusso Ammissibile

- Data una rete G, è possibile determinare se esiste un **flusso** ammissibile (ma non necessariamente ottimo) per essa?
- Basta risolvere il problema MF sulla rete ottenuta nel modo seguente



Costruire un Flusso Ammissibile

- Data una rete G, è possibile determinare se esiste un **flusso** ammissibile (ma non necessariamente ottimo) per essa?
- Basta risolvere il problema MF sulla rete ottenuta nel modo seguente



- Otteniamo così un algoritmo, che chiamiamo FlussoAmmissibile(G), che data una rete, calcola un flusso ammissibile per essa (se esiste).

CANCELLAZIONECICLI(G)

1. Se FLUSSOAMMISSIBILE(G) restituisce un flusso ammissibile, allora mettilo in x, altrimenti termina: il problema è vuoto.
2. Cerca un ciclo di costo negativo in G_x. Se non lo trovi, allora termina e restituisci x, altrimenti metti il ciclo in C.
3. $x \leftarrow x(C, \theta(C, x))$;
4. Torna al punto 2.

Cancellazione di Cicli — Proprietà

- La **correttezza** dell'algoritmo è una banale conseguenza del Lemma (cha abbiamo dimostrato) sull'equivalenza tra assenza di cicli aumentanti e ottimalità.
- Come al solito, se le capacità sono numeri interi, allora qualcosa diminuisce di almeno 1 ad ogni iterazione, ossia il *costo* (e quindi l'algoritmo **termina**).
- Il costo di qualunque flusso ammissibile è compreso tra $-A\overline{uc}$ e $A\overline{uc}$, dove

$$\overline{u} = \max\{u_{ij} \mid (i,j) \in N\};$$
$$\overline{c} = \max\{c_{ij} \mid (i,j) \in N\}.$$

  La **complessità** dell'algoritmo sarà quindi *pseudoplinomiale*, ossia

$$O(NA) \cdot O(A\overline{uc}) = O(NA^2\overline{uc})$$

Esistono Algoritmi Polytime?

Esistono Algoritmi Polytime?

Algoritmo	*Complessità*
EM	$O((A \log V) \cdot (\min\{V \log(VC), AV \log V\}))$
Goldberg-Tarjan	$O(V^2 A^3 (\log V))$

Sezione 4

Problemi di Accoppiamento

Nozioni Preliminari — I

- Nei problemi di accoppiamento, si lavora con **grafi bipartiti non orientati** ossia con grafi nella forma $G = (O \cup D, A)$, dove:
  - $O = \{1, \ldots, n\}$ è l'insieme dei **nodi origine**;
  - $D = \{n+1, \ldots, n+d\}$ è l'insieme dei **nodi destinazione**;
  - $A \subseteq O \times D$ è l'insieme degli **archi**, a ciascuno dei quali è associato un **costo**.
- Un **accoppiamento** per un grafo bipartito $G = (O \cup D, A)$ è un sottoinsieme M di A i cui archi non abbiano nodi in comune.
  - Gli archi in M si dicono **interni**, mentre quelli in $A - M$ sono detti **esterni**.
  - I nodi che compaiono in qualche arco di M si dicono **accoppiati**, gli altri nodi si dicono invece **esposti**.

Nozioni Preliminari — II

- M è detto **accoppiamento perfetto** sse non vi sono nodi esposti.
- Il *costo* di un accoppiamento M è nient'altro che

$$c(M) = \sum_{(i,j) \in M} c_{ij}.$$

- Dato M, l'arco $(i,j) \in M$ di costo massimo è detto **arco bottleneck** e il valore

$$\max\{c_{ij} \mid (i,j) \in M\}$$

  è detto **valore di bottleneck**

Problemi di Accoppiamento

- **Accoppiamento di Massima Cardinalità**
  - Si vuole determinare, semplicemente, l'accoppiamento di massima cardinalità
- **Accoppiamento di Costo Minimo**
  - Si vuole determinare l'accoppiamento di costo minimo tra tutti gli accopiamenti *perfetti*.
- **Accoppiamento di Massima Cardinalità Bottleneck**
  - Tra tutti gli accoppiamenti di massima cardinalità, si vuole determinare quello con valore di bottleneck minimo.

Accoppiamento di Massima Cardinalità — I

- Il problema può essere visto come un problema di **flusso massimo** con più sorgenti (i nodi in O) e più pozzi (i nodi in D).
  - Le **capacità** saranno tutte pari a 1.
  - Ci interessano solamente **flussi interi**.
  - Come sappiamo, una rete con molte sorgenti e molte destinazioni può essere poi tradotta in una rete con **una** sorgente e **una** destinazione.
- *Flussi ammissibili interi* e *accoppiamenti* sono in corrispondenza biunivoca.
  - Indicheremo quindi, ad esempio, un flusso con M e il relativo grafo residuo con G_M.

Accoppiamento di Massima Cardinalità — II

- È possibile utilizzare gli algoritmi classici per il problema MF, ma c'è spazio per sfruttare le caratteristiche peculiari dei problemi di accoppiamento.
- A tal proposito osserviamo che ogni cammino aumentante in G_M deve:
  - Essere **alternante**, ossia consistere di archi interni, seguiti da archi esterni, seguiti da archi interni, e così via.
  - Partire da un'origine **esposta** e arrivare ad una destinazione **esposta**.
  - In altre parole, se $P_E = P - M$ e $P_I = M \cap P$ sono rispettivamente gli archi esterni e interni di un cammino aumentante P, allora $|P_E| - |P_I| = 1$.
- La capacità $\theta(M, P)$ di un cammino aumentante P sarà sempre 1, e di conseguenza

$$M \oplus (\theta(M, P))P = (M - P_I) \cup P_E.$$

Accoppiamento di Massima Cardinalità — III

- Otteniamo in questo modo un algoritmo del tutto simile a FF, ma in cui la ricerca del cammino aumentante può essere eseguita tramite una semplice procedura di visita.
- La complessità di quest'algoritmo, a differenza di quella di FF, sarà $O(mn)$, perché

$$U = \max\{c_{ij} \mid (i,j) \in A\} = 1.$$

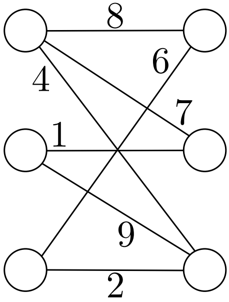
- Nel caso il problema in question sia l'accoppiamento **di costo minimo**, si può procedere similmente all'accoppiamento di massima cardinalità, ma in questo caso specializzando gli algoritmi per MCF.
  - In particolare, i camminimi minimi aumentanti potrebbero essere visti come cammini esposti tra due vertici esposti, rispettivamente in O e in D.

Accoppiamento di Costo Minimo

- Seguendo lo stesso identico tipo di ragionamento, è possibile vedere una qualunque istanza del problema di accoppiamento di costo minimo come un'istanza di MCF.

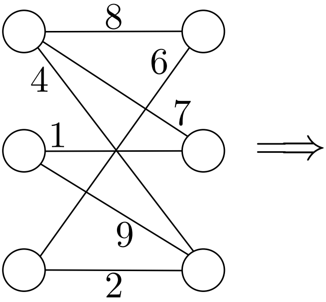
Accoppiamento di Costo Minimo

- Seguendo lo stesso identico tipo di ragionamento, è possibile vedere una qualunque istanza del problema di accoppiamento di costo minimo come un'istanza di MCF.
- Ad esempio:



Accoppiamento di Costo Minimo

- Seguendo lo stesso identico tipo di ragionamento, è possibile vedere una qualunque istanza del problema di accoppiamento di costo minimo come un'istanza di MCF.
- Ad esempio:



Accoppiamento di Costo Minimo

- Seguendo lo stesso identico tipo di ragionamento, è possibile vedere una qualunque istanza del problema di accoppiamento di costo minimo come un'istanza di MCF.
- Ad esempio: