

Problema del flusso massimo

Mauro Passacantando

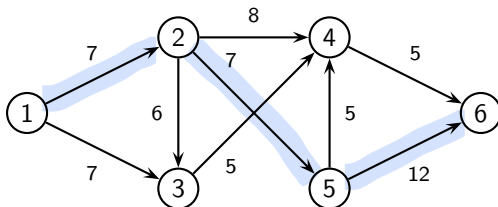
Dipartimento di Informatica, Università di Pisa
mauro.passacantando@unipi.it

Corso di Ricerca Operativa A
Laurea in Informatica - Università di Pisa - a.a. 2019/20

Problema

Sia (N, A) un grafo orientato in cui è definita una capacità superiore u_{ij} per ogni arco $(i, j) \in A$. Dati un nodo origine $s \in N$ e un nodo destinazione $t \in N$, vogliamo spedire la massima quantità di flusso da s a t in modo da rispettare le capacità superiori degli archi.

Esempio 1. Trovare il flusso di valore massimo dal nodo 1 al nodo 6 sul seguente grafo (sugli archi sono riportate le capacità superiori).



Modello di ottimizzazione

Variabili:

ad ogni arco $(i, j) \in A$ associamo x_{ij} = flusso sull'arco (i, j)

v = flusso totale uscente dal nodo s

Modello:

$$\max v$$

$$\left. \begin{aligned} \sum_{(i,s) \in A} x_{is} - \sum_{(s,j) \in A} x_{sj} &= -v \\ \sum_{(i,t) \in A} x_{it} - \sum_{(t,j) \in A} x_{tj} &= v \\ \sum_{(i,k) \in A} x_{ik} - \sum_{(k,j) \in A} x_{kj} &= 0 \quad \forall k \in N \setminus \{s, t\} \end{aligned} \right\} \text{vincoli di bilancio}$$

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

Condizioni di ottimalità

Un **taglio** è una partizione di N in due sottoinsiemi (N_s, N_t)

Un **taglio ammissibile** è un taglio (N_s, N_t) tale che $s \in N_s$ e $t \in N_t$

Dato un taglio ammissibile (N_s, N_t) ed un flusso x , si definiscono:

$A^+ = \{(i, j) \in A : i \in N_s, j \in N_t\}$ insieme degli **archi diretti** del taglio

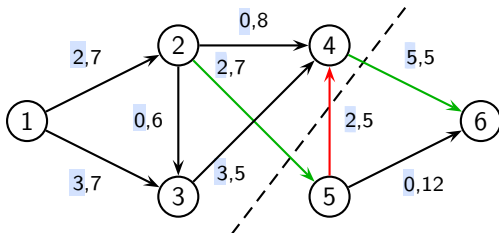
$A^- = \{(i, j) \in A : i \in N_t, j \in N_s\}$ insieme degli **archi inversi** del taglio

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

$$x(N_s, N_t) = \sum_{(i,j) \in A^+} x_{ij} - \sum_{(i,j) \in A^-} x_{ij} \quad \text{flusso sul taglio}$$

Condizioni di ottimalità

Esempio 2. Si consideri il problema del flusso massimo da 1 a 6 sul seguente grafo in cui su ogni arco sono indicati, in ordine, il **flusso** e la capacità superiore:



Il taglio (N_s, N_t) , dove $N_s = \{1, 2, 3, 4\}$ e $N_t = \{5, 6\}$, è ammissibile.

archi diretti: $A^+ = \{(2, 5), (4, 6)\}$ (tra quelli che attraversano il taglio)

archi inversi: $A^- = \{(5, 4)\}$

capacità del taglio: $u(N_s, N_t) = u_{25} + u_{46} = 7 + 5 = 12$

flusso sul taglio: $x(N_s, N_t) = x_{25} + x_{46} - x_{54} = 7 + 5 - 2 = 5$.

Condizioni di ottimalità

Lemma

Se x è un flusso ammissibile e (N_s, N_t) è un taglio ammissibile, allora

- ▶ $v = x(N_s, N_t)$ (valore del flusso = flusso sul taglio)
- ▶ $x(N_s, N_t) \leq u(N_s, N_t)$ (flusso sul taglio \leq capacità del taglio)

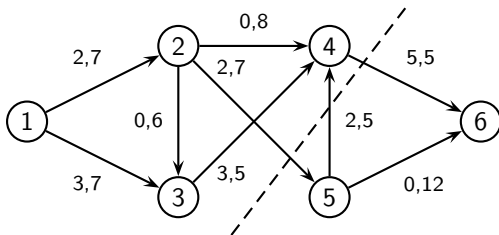
Condizioni di ottimalità

Lemma

Se x è un flusso ammissibile e (N_s, N_t) è un taglio ammissibile, allora

- ▶ $v = x(N_s, N_t)$ (valore del flusso = flusso sul taglio)
- ▶ $x(N_s, N_t) \leq u(N_s, N_t)$ (flusso sul taglio \leq capacità del taglio)

Esempio 2. Si consideri il problema del flusso massimo da 1 a 6 sul seguente grafo in cui su ogni arco sono indicati, in ordine, il flusso e la capacità superiore:



valore del flusso: $v = x_{12} + x_{13} = 2 + 3 = 5$

flusso sul taglio: $x(N_s, N_t) = x_{25} + x_{46} - x_{54} = 2 + 5 - 2 = 5$.

capacità del taglio: $u(N_s, N_t) = u_{25} + u_{46} = 7 + 5 = 12$

Condizioni di ottimalità

Teorema (max flow – min cut)

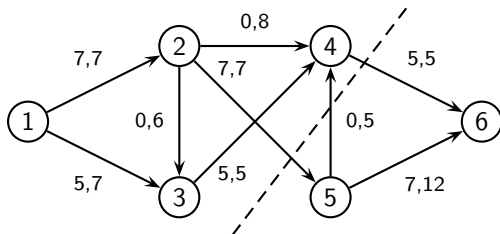
Se esistono un flusso ammissibile x ed un taglio ammissibile (N_s, N_t) tali che $x(N_s, N_t) = u(N_s, N_t)$, allora x è un flusso di valore massimo e (N_s, N_t) è un taglio di capacità minima.

Condizioni di ottimalità

Teorema (max flow – min cut)

Se esistono un flusso ammissibile x ed un taglio ammissibile (N_s, N_t) tali che $x(N_s, N_t) = u(N_s, N_t)$, allora x è un flusso di valore massimo e (N_s, N_t) è un taglio di capacità minima.

Esempio 3. Si consideri il problema del flusso massimo da 1 a 6 sul seguente grafo in cui su ogni arco sono indicati, in ordine, il flusso e la capacità superiore:



valore del flusso: $v = x_{12} + x_{13} = 7 + 5 = 12$

flusso sul taglio: $x(N_s, N_t) = x_{25} + x_{46} - x_{54} = 7 + 5 - 0 = 12$

capacità del taglio: $u(N_s, N_t) = u_{25} + u_{46} = 7 + 5 = 12$

Quindi il flusso è di valore massimo ed il taglio è di capacità minima.

Grafo residuo

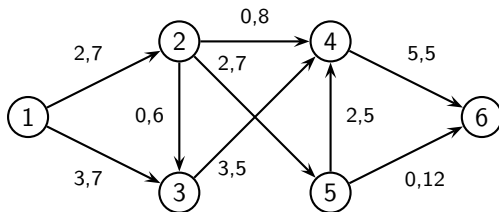
Dato un flusso ammissibile x , il **grafo residuo** relativo ad x è il **grafo** $G(x) = (N, A(x))$ avente gli **stessi nodi del grafo G** , mentre gli **archi e le loro capacità residue r_{ij}** sono definiti come segue:

se $(i, j) \in A$ con $x_{ij} < u_{ij}$ (**arco non saturo**) allora $(i, j) \in A(x)$ con $r_{ij} = u_{ij} - x_{ij}$

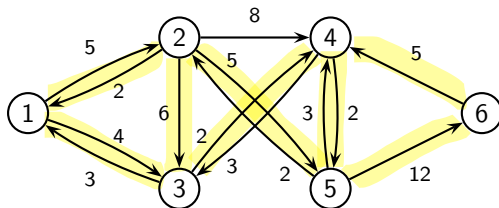
se $(i, j) \in A$ con $x_{ij} > 0$ (**arco non vuoto**) allora $(j, i) \in A(x)$ con $r_{ji} = x_{ij}$

Grafo residuo

Esempio 4. Si consideri il problema del flusso massimo da 1 a 6 sul seguente grafo in cui su ogni arco sono indicati, in ordine, il flusso e la capacità superiore:



Il **grafo residuo** associato al flusso indicato sugli archi è il seguente (sugli archi sono indicate le capacità residue):



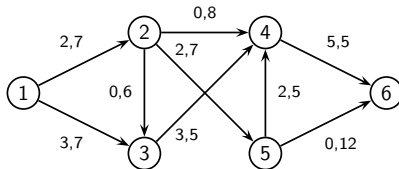
Cammino aumentante

Dato un flusso ammissibile x , un **cammino aumentante** rispetto ad x è un cammino orientato da s a t nel grafo residuo $G(x)$.

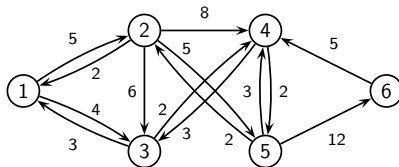
Cammino aumentante

Dato un flusso ammissibile x , un **cammino aumentante** rispetto ad x è un cammino orientato da s a t nel grafo residuo $G(x)$.

Esempio 4. Si consideri il problema del flusso massimo da 1 a 6 sul seguente grafo in cui su ogni arco sono indicati, in ordine, il flusso e la capacità superiore:



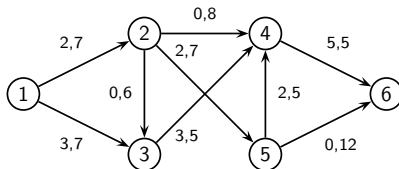
Il grafo residuo è:



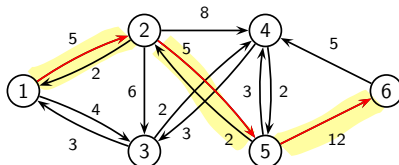
Cammino aumentante

Dato un flusso ammissibile x , un **cammino aumentante** rispetto ad x è un cammino orientato da s a t nel grafo residuo $G(x)$.

Esempio 4. Si consideri il problema del flusso massimo da 1 a 6 sul seguente grafo in cui su ogni arco sono indicati, in ordine, il flusso e la capacità superiore:



Il **grafo residuo** è:

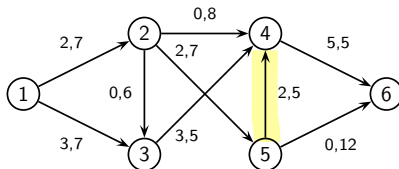


1-2-5-6 è un cammino aumentante (ed è un cammino orientato nel grafo iniziale)

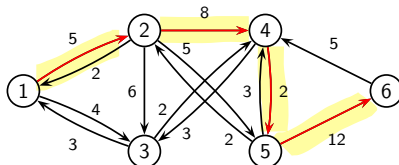
Cammino aumentante

Dato un flusso ammissibile x , un **cammino aumentante** rispetto ad x è un cammino orientato da s a t nel grafo residuo $G(x)$.

Esempio 4. Si consideri il problema del flusso massimo da 1 a 6 sul seguente grafo in cui su ogni arco sono indicati, in ordine, il flusso e la capacità superiore:



Il grafo residuo è:



1-2-5-6 è un cammino aumentante (ed è un cammino orientato nel grafo iniziale)

1-2-4-5-6 è un **cammino aumentante** (ma non è un cammino orientato nel grafo iniziale)

Algoritmo di Ford-Fulkerson

Ad ogni iterazione cerca un cammino aumentante rispetto al flusso corrente.

Se esiste un cammino aumentante, l'algoritmo aggiorna il flusso spedendo su tale cammino il massimo flusso possibile (pari alla minima capacità residua degli archi di tale cammino).

Se non esiste un cammino aumentante, l'algoritmo si ferma perché il flusso corrente è di valore massimo. Inoltre l'algoritmo trova anche un taglio di capacità minima.

Algoritmo di Ford-Fulkerson

Algoritmo

0. Poni $x = 0$ (flusso nullo su tutti gli archi) e $G(x) = G$
1. Se esiste un cammino aumentante C rispetto ad x allora
aggiorna il flusso:
calcola $\delta = \min\{r_{ij} : (i, j) \in C\}$ (max quantità da spedire lungo C)
per ogni $(i, j) \in A$:
se $(i, j) \in C$ allora $x_{ij} = x_{ij} + \delta$
se $(j, i) \in C$ allora $x_{ij} = x_{ij} - \delta$
altrimenti stop
2. Aggiorna il grafo residuo $G(x)$ e torna al passo 1

Algoritmo di Ford-Fulkerson

Teorema

Se le capacità degli archi sono intere, allora l'algoritmo di Ford-Fulkerson trova un flusso di valore massimo dopo un numero finito di iterazioni.

All'ultima iterazione un taglio di capacità minima (N_s, N_t) si può calcolare nel modo seguente:

$$N_s = \{i \in N : \text{esiste in } G(x) \text{ un cammino orientato da } s \text{ a } i\},$$

$$N_t = \{i \in N : \text{non esiste in } G(x) \text{ un cammino orientato da } s \text{ a } i\},$$

Un altro taglio di capacità minima (N'_s, N'_t) si può ottenere nel modo seguente:

$$N'_s = \{i \in N : \text{non esiste in } G(x) \text{ un cammino orientato da } i \text{ a } t\},$$

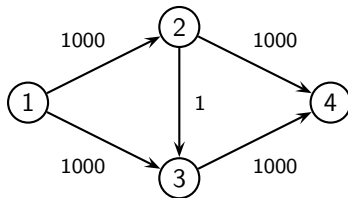
$$N'_t = \{i \in N : \text{esiste in } G(x) \text{ un cammino orientato da } i \text{ a } t\},$$

Il taglio di capacità minima è unico se e solo se i due tagli precedenti coincidono.

Algoritmo di Ford-Fulkerson

Se non viene data una regola precisa per la ricerca del cammino aumentante, la convergenza dell'algoritmo può essere molto lenta.

Esempio 5. Consideriamo il problema di flusso massimo da 1 a 4 sul seguente grafo:



Se scegliamo come cammini aumentanti 1-2-3-4, 1-3-2-4, 1-2-3-4, ecc. ad ogni iterazione spediamo solo 1 unità di flusso lungo il cammino aumentante e l'algoritmo termina dopo 2000 iterazioni.

Algoritmo di Edmonds-Karp

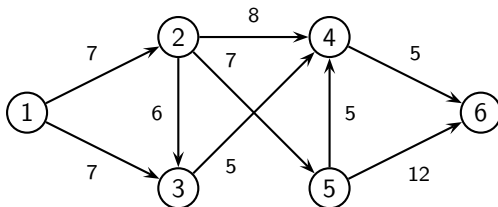
L'algoritmo di Edmonds-Karp è una visita a ventaglio del grafo residuo e trova un cammino aumentante con il minimo numero di archi.

Usa un vettore p di predecessori dei nodi ed un insieme Q di nodi.

1. Poni $p_i := \begin{cases} -1 & \text{se } i \neq s \\ 0 & \text{se } i = s \end{cases}$, $Q := [s]$
2. **if** $Q = \emptyset$
 then STOP (p fornisce un taglio di capacità minima
 $N_s = \{i \in N : p_i \neq -1\}$, $N_t = \{i \in N : p_i = -1\}$)
 else estrai il primo elemento i di Q .
3. **if** $(i, t) \in A(x)$ **then** $p_t := i$, STOP (p fornisce un cammino aumentante).
4. In $G(x)$ analizza gli archi uscenti da i in ordine lessicografico
 Per ogni $(i, j) \in A(x)$:
 if $p_j := -1$ **then** $p_j := i$, aggiungi j in fondo a Q .
 Torna al passo 2.

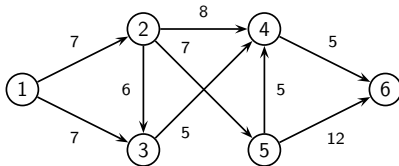
Algoritmo di Edmonds-Karp

Esempio 6. Applichiamo l'algoritmo di Ford-Fulkerson, utilizzando l'algoritmo di Edmonds-Karp per la ricerca del cammino aumentante, per trovare un flusso di valore massimo dal nodo 1 al nodo 6 sul seguente grafo (sugli archi sono riportate le capacità superiori).



Algoritmo di Edmonds-Karp

Iterazione 1. Partiamo con $x = 0$, quindi il grafo residuo $G(x) = G$:

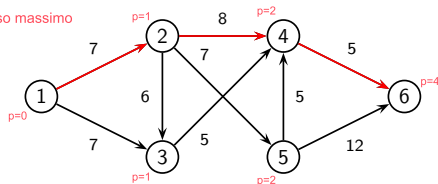


Algoritmo di Edmonds-Karp

$x=0$ (flusso nullo su tutti gli archi)

Iterazione 1. Partiamo con $x = 0$, quindi il grafo residuo $G(x) = G$:

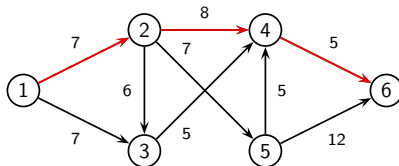
Ad ogni nodo (i) prendo l'arco di flusso massimo



L'algoritmo di Edmonds-Karp trova il vettore $p = (0, 1, 1, 2, 2, 4)$ che fornisce il cammino **aumentante 1-2-4-6**. Si ha $\delta = \min\{7, 8, 5\} = 5$.

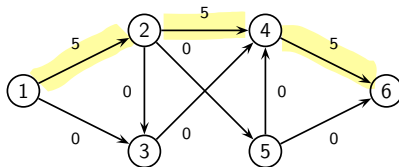
Algoritmo di Edmonds-Karp

Iterazione 1. Partiamo con $x = 0$, quindi il grafo residuo $G(x) = G$:



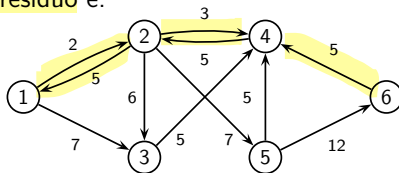
L'algoritmo di Edmonds-Karp trova il vettore $p = (0, 1, 1, 2, 2, 4)$ che fornisce il cammino aumentante 1-2-4-6. Si ha $\delta = \min\{7, 8, 5\} = 5$.

Il nuovo flusso è



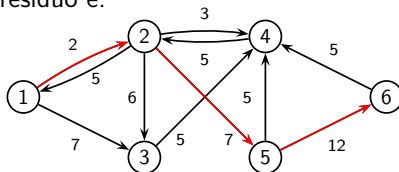
Algoritmo di Edmonds-Karp

Iterazione 2. Il grafo residuo è:



Algoritmo di Edmonds-Karp

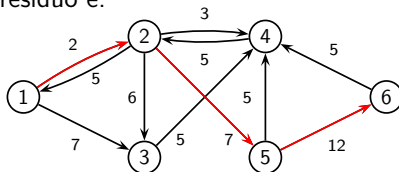
Iterazione 2. Il grafo residuo è:



L'algoritmo di Edmonds-Karp trova il vettore $p = (0, 1, 1, 2, 2, 5)$ che fornisce il cammino aumentante 1-2-5-6. Si ha $\delta = \min\{2, 7, 12\} = 2$.

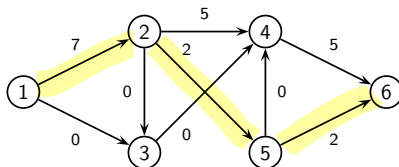
Algoritmo di Edmonds-Karp

Iterazione 2. Il grafo residuo è:



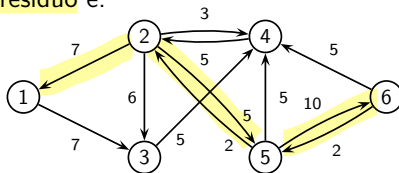
L'algoritmo di Edmonds-Karp trova il vettore $p = (0, 1, 1, 2, 2, 5)$ che fornisce il cammino aumentante 1-2-5-6. Si ha $\delta = \min\{2, 7, 12\} = 2$.

Il nuovo flusso è



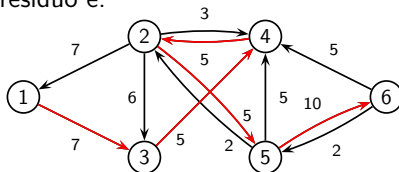
Algoritmo di Edmonds-Karp

Iterazione 3. Il **grafo residuo** è:



Algoritmo di Edmonds-Karp

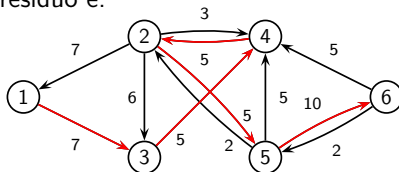
Iterazione 3. Il grafo residuo è:



L'algoritmo di Edmonds-Karp trova il vettore $p = (0, 4, 1, 3, 2, 5)$ che fornisce il cammino aumentante 1-3-4-2-5-6. Si ha $\delta = \min\{7, 5, 5, 5, 10\} = 5$.

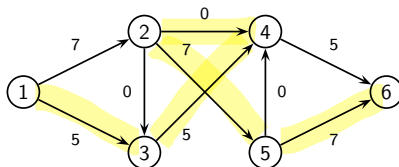
Algoritmo di Edmonds-Karp

Iterazione 3. Il grafo residuo è:



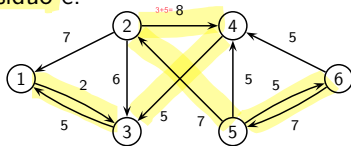
L'algoritmo di Edmonds-Karp trova il vettore $p = (0, 4, 1, 3, 2, 5)$ che fornisce il cammino aumentante 1-3-4-2-5-6. Si ha $\delta = \min\{7, 5, 5, 5, 10\} = 5$.

Il nuovo flusso è



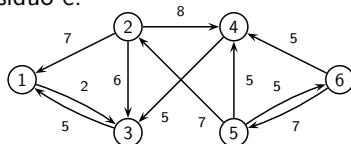
Algoritmo di Edmonds-Karp

Iterazione 4. Il grafo residuo è:



Algoritmo di Edmonds-Karp

Iterazione 4. Il grafo residuo è:



L'algoritmo di Edmonds-Karp trova il vettore $p = (0, -1, 1, -1, -1, -1)$, pertanto il flusso trovato all'iterazione precedente è ottimo ($v = 12$) ed un taglio di capacità minima è

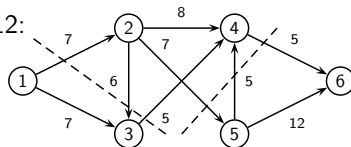
$N_s = \{1, 3\}$ (nodi raggiungibili da 1), $N_t = \{2, 4, 5, 6\}$ (nodi non raggiungibili da 1).

Notiamo che un altro taglio di capacità minima è

$N'_s = \{1, 2, 3, 4\}$ (nodi dai quali non è possibile raggiungere 6),

$N'_t = \{5, 6\}$ (nodi dai quali è possibile raggiungere 6).

Osserviamo che i due tagli hanno capacità 12:



Esercizio

La Scuola di Ingegneria si divide in 4 dipartimenti (Informatica, Elettronica, Meccanica, Civile). Ogni dipartimento ha ricevuto il seguente numero di domande di studenti per partecipare al programma Erasmus: 20 per Informatica, 16 per Elettronica, 15 per Meccanica e 10 per Civile. Sono disponibili varie convenzioni con diverse università straniere. In Spagna non si possono mandare più di 30 studenti, in Germania non più di 20, in Norvegia non più di 5, in Francia non più di 25, in Gran Bretagna non più di 15. Le convenzioni attivate non permettono di mandare più di 10 studenti di un dipartimento nello stesso paese. Inoltre, gli studenti di Informatica non possono andare in Francia, quelli di Elettronica in Spagna, quelli di Meccanica in Norvegia e quelli di Civile in Germania.

Formulare il problema di massimizzare il numero di studenti da mandare nei programmi Erasmus come un problema di flusso massimo su un opportuno grafo e trovare la soluzione ottima. Qual è un taglio di capacità minima?

Problema dell'accoppiamento di massima cardinalità

Un grafo orientato (N, A) è detto **bipartito** se esistono due insiemi $O, D \subset N$ tali che $N = O \cup D$ e $A \subseteq O \times D$.

In un grafo bipartito (N, A) un **accoppiamento** è un insieme di archi $M \subseteq A$ tale che su ogni nodo di N incide al più un arco di M .

Il problema dell'accoppiamento di massima cardinalità in un grafo bipartito (N, A) consiste nel trovare un **accoppiamento con il massimo numero di archi**.

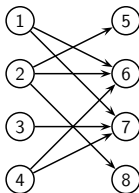
Problema dell'accoppiamento di massima cardinalità

Un grafo orientato (N, A) è detto **bipartito** se esistono due insiemi $O, D \subset N$ tali che $N = O \cup D$ e $A \subseteq O \times D$.

In un grafo bipartito (N, A) un **accoppiamento** è un insieme di archi $M \subseteq A$ tale che su ogni nodo di N incide al più un arco di M .

Il problema dell'accoppiamento di massima cardinalità in un grafo bipartito (N, A) consiste nel trovare un accoppiamento con il massimo numero di archi.

Esempio. Supponiamo che O sia un insieme di operai, D un insieme di lavori ed ogni arco $(i, j) \in A$ indichi che l'operaio i è in grado di svolgere il lavoro j . Ogni operaio può eseguire al più un lavoro ed ogni lavoro può essere eseguito da al più un operaio. Come assegniamo gli operai ai lavori in modo da eseguire il maggior numero di lavori possibile?



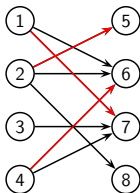
Problema dell'accoppiamento di massima cardinalità

Un grafo orientato (N, A) è detto **bipartito** se esistono due insiemi $O, D \subset N$ tali che $N = O \cup D$ e $A \subseteq O \times D$.

In un grafo bipartito (N, A) un **accoppiamento** è un insieme di archi $M \subseteq A$ tale che su ogni nodo di N incide al più un arco di M .

Il problema dell'accoppiamento di massima cardinalità in un grafo bipartito (N, A) consiste nel trovare un accoppiamento con il massimo numero di archi.

Esempio. Supponiamo che O sia un insieme di operai, D un insieme di lavori ed ogni arco $(i, j) \in A$ indichi che l'operaio i è in grado di svolgere il lavoro j . Ogni operaio può eseguire al più un lavoro ed ogni lavoro può essere eseguito da al più un operaio. Come assegniamo gli operai ai lavori in modo da eseguire il maggior numero di lavori possibile?



$M = \{(1, 7), (2, 5), (4, 6)\}$ è un accoppiamento. È anche di massima cardinalità?

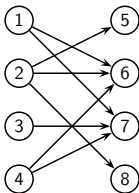
Problema dell'accoppiamento di massima cardinalità

Il problema dell'accoppiamento di massima cardinalità su un grafo bipartito (N, A) è equivalente al problema del flusso massimo sul grafo (N', A') , dove $N' = N \cup \{s, t\}$, $A' = A \cup \{(s, i) : i \in O\} \cup \{(j, t) : j \in D\}$ e le capacità superiori di tutti gli archi sono uguali a 1.

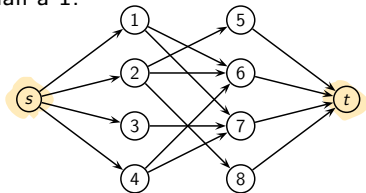
Problema dell'accoppiamento di massima cardinalità

Il problema dell'accoppiamento di massima cardinalità su un grafo bipartito (N, A) è equivalente al problema del flusso massimo sul grafo (N', A') , dove $N' = N \cup \{s, t\}$, $A' = A \cup \{(s, i) : i \in O\} \cup \{(j, t) : j \in D\}$ e le capacità superiori di tutti gli archi sono uguali a 1.

Esempio. L'accoppiamento di massima cardinalità sul grafo

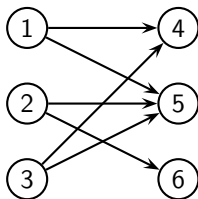


è equivalente al problema del flusso massimo sul grafo seguente in cui le capacità superiori degli archi sono uguali a 1.



Esercizio

Trovare un accoppiamento di massima cardinalità sul grafo



risolvendo il problema di flusso massimo equivalente.