Problemi di flusso e di accoppiamento
Max-Flow e Matching

Salvatore Filippone
salvatore.filippone@uniroma2.it

Problema del flusso in una rete

Una *rete di flusso* $(\mathcal{V}, \mathcal{E}, s, p, c)$ è data da un grafo

$$\mathcal{G} = (\mathcal{V}, \mathcal{E})$$

due vertici s e p detti rispettivamente *sorgente* e *pozzo*, ed una funzione di *capacità* a valori interi positivi $c : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{Z}^+ \cup \{0\}$, tale che $c(u, v) = 0$ se $(u, v) \notin \mathcal{E}$, allora

Definizione

Un flusso in $\mathcal{G}$ è una funzione a valori interi $F : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{Z}$ che soddisfa:

Simmetria opposta: $f(u, v) = -f(v, u)$ *per ogni* $u, v \in \mathcal{V}$;

Vincolo di capacità: $f(u, v) \leq c(u, v)$ *per ogni* $u, v \in \mathcal{V}$;

Conservazione del flusso: $\sum_v f(u, v) = 0$ *per ogni nodo* $u \in \mathcal{V} - \{s, p\}$.

Il *valore* del flusso è la somma dei flussi in uscita dalla sorgente (o in entrata nel pozzo)

$$|f| = \sum_v f(s, v).$$

Flusso Massimo

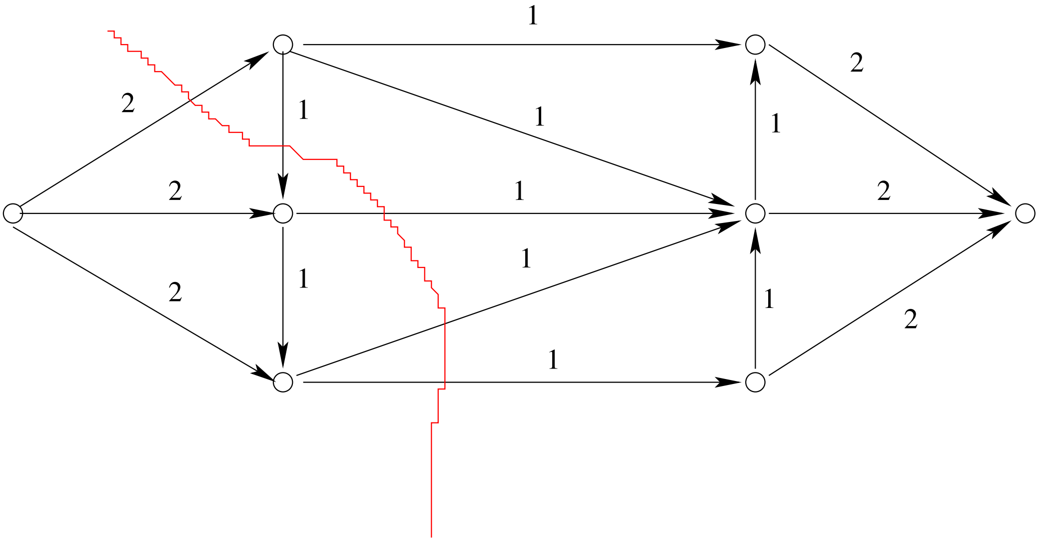
Data una rete di flusso $(\mathcal{V}, \mathcal{E}, s, p, c)$ trovare il flusso ottimale f^* che massimizza $|f^*|$ su tutti i possibili flussi.

Taglio

Un taglio è una partizione dei nodi $(\mathcal{S}, \mathcal{P})$ tale che:

1. $s \in \mathcal{S}$;
2. $p \in \mathcal{P}$;
3. $\mathcal{S} \cup \mathcal{P} = \mathcal{V}$;
4. $\mathcal{S} \cap \mathcal{P} = \emptyset$;

La capacità del taglio è $\sum_{u \in \mathcal{S}, v \in \mathcal{P}} c(u, v)$


1
2
2
1
1
1
2
2
1
1
2
1
1
1
2

Problema del flusso in una rete

Capacità residua e cammino aumentante

Capacità residua: $r(u, v) = c(u, v) - f(u, v)$;

Rete di flusso residua R: La rete contenente il sottoinsieme degli archi tale che $r(u, v) > 0$;

Cammino aumentante: un cammino da s a p in R; la sua capacità residua è il più piccolo degli $r(u, v)$ sugli archi (u, v) che appartengono al cammino.

Teorema

Il flusso massimo è uguale alla capacità del taglio minimo. Inoltre le seguenti condizioni sono equivalenti:

1. *f è un flusso massimo;*
2. *Non esiste alcun cammino aumentante;*
3. *Esiste un taglio $(\mathcal{S}, \mathcal{P})$ tale che $|f| = c(\mathcal{S}, \mathcal{P})$.*

Idea algoritmica di base: cercare i cammini aumentanti, finché ce ne sono.

Algorithm 1: maxFlow(Grafo G, Nodo s, Nodo p, **integer**[][] c)

Nodo u, v;
integer[][] $f \leftarrow$ **new integer** [][];
integer[][] $g \leftarrow$ **new integer** [][];
foreach $u, v \in G.V()$ **do**
  $f[u, v] \leftarrow 0$;
boolean*stop* $\leftarrow$ **false**;
while not *stop* **do**
  $R \leftarrow$ Rete di flusso residua di f in G;
  $g \leftarrow$ flusso associato ad uno o più cammini aumentanti;
  foreach $u, v \in G.V()$ **do**
    $f[u, v] \leftarrow f[u, v] + g[u, v]$;
  if $\forall u, v \in G.V() : g[u, v] = 0$ **then**
    stop $\leftarrow$ **true**;

Problema del flusso in una rete

Algoritmo Ford-Fulkerson: si usa un cammino aumentante qualsiasi; costo nel caso peggiore $O((n+m)\cdot|f^*|)$;

Algoritmo di Edmonds-Karp: Ricerca i cammini aumentanti con una visita in ampiezza BFS; complessità $O(nm^2)$;

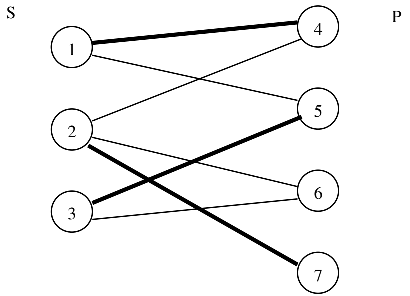
Algoritmo dei tre indiani: Aumentare lungo i cammini aumentanti più corti.

Esempio: abbinamento in un grafo bipartito

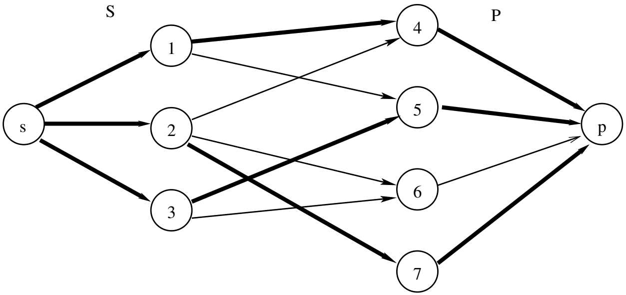
Un grafo bipartito è un grafo per cui esiste una partizione dei nodi $(\mathcal{S}, \mathcal{P})$ tale che

$$\forall (u, v) : u \in \mathcal{S},\ v \in \mathcal{P}$$

Un abbinamento (*matching*)è un sottoinsieme degli archi, ossia un insieme di coppie di nodi; un abbinamento è massimo se il numero di coppie è massimo.



Un modo per risolvere il problema è di aggiungere due nodi s e p, definire gli archi con capacità unitaria, e calcolare il flusso massimo



Esempio: abbinamento in un grafo bipartito

L'abbinamento può essere:

- Pesato (esiste una funzione $w(u, v)$);
- Su grafo non bipartito

Si usa, ad esempio, nella costruzione di solutori per sistemi lineari:
P. D'Ambra, S. Filippone, P. Vassilevski: BootCMatch: a software package for bootstrap AMG based on graph weighted matching, ACM TOMS, Vol. 44, No. 4, Aug. 2018