

# Esercitazione 5

## Network Flow

Diamo innanzitutto una definizione informale del concetto di riduzione polinomiale tra problemi:

Si dice che un problema  $A$  è riducibile polinomialmente ad un problema  $B$ , se è possibile trasformare tramite una funzione  $f$ , che opera in tempo polinomiale, un'istanza  $x$  del problema  $A$  in un'istanza  $y = f(x)$  del problema  $B$ , in modo che esiste una soluzione per  $A(x)$  se e solo se esiste una soluzione per  $B(y)$ .

Useremo tale concetto per risolvere una serie di problemi riducendoli a un problema di flusso. Faremo vedere come una soluzione sul problema di partenza ci definisce un *max-flow* su una rete ottenuta da tale istanza e viceversa.

### Problema 11

Supponiamo che  $n$  studenti decidano, per un periodo di  $m$  giorni, di utilizzare a turno le proprie automobili per andare a lezione. Supponiamo anche che ogni auto sia sufficientemente capiente da poter trasportare tutti gli studenti.

Non tutti gli studenti devono andare a lezione negli stessi giorni: in altri termini, esiste una funzione booleana che associa ad una coppia studente-giorno il valore *true* se lo studente deve andare a lezione quel giorno e *false* altrimenti.

Un esempio molto ridotto di tale problema, con  $n = 4$  e  $m = 5$  è riportato nella tabella sottostante, in cui per ogni giorno da Lunedì a Venerdì viene riportato se lo studente  $i$ -esimo deve andare a lezione.

	Lu	Ma	Me	Gi	Ve
1	X	X	X		
2		X		X	X
3	X				X
4	X		X	X	

Vogliamo trovare una assegnazione "equa" dell'utilizzo delle automobili, nel senso intuitivo che chi deve andare più spesso a lezione dovrà anche mettere più spesso a disposizione la proprio automobile.

Possiamo assegnare ad ogni studente un valore che misura la sua quota di utilizzo del car sharing nel modo seguente:

1. per ogni giorno, se  $k$  sono i viaggiatori, ognuno di essi riceve un valore pari a  $\frac{1}{k}$
2. il valore  $T_i$  assegnato ad uno studente è pari alla somma dei valori ricevuti ogni giorno

La tabella seguente mostra i valori nell'esempio considerato, l'ultima colonna specifica la quota di utilizzo per ogni studente.

	Lu	Ma	Me	Gi	Ve	T
1	1/3	1/2	1/2			4/3
2		1/2		1/2	1/2	3/2
3	1/3				1/2	5/6
4	1/3		1/2	1/2		4/3

L'assegnazione che cerchiamo dovrà far sì che il viaggiatore  $i$  non usi la propria automobile per più di  $\lceil T_i \rceil$  giorni, sugli  $m$  complessivi.

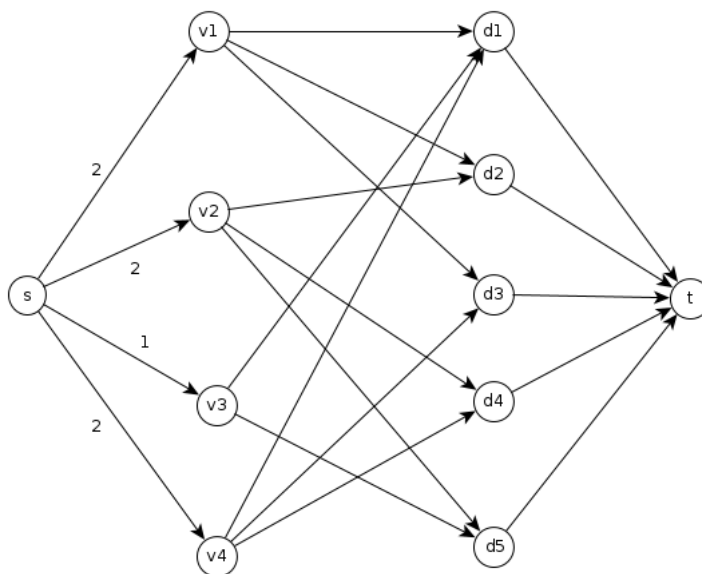
**Soluzione** Al fine di trovare un'assegnazione di automobili che rispetti i vincoli imposti dal problema, effettueremo una riduzione al problema *max-flow* su una rete costruita a partire dall'istanza del problema di partenza.

Nella nostra rete ci sarà un nodo  $v_i$  per ogni studente  $i$  ed un nodo  $d_j$  per ogni giorno  $j$ , sono inoltre presenti un nodo sorgente  $s$  e un nodo target  $t$ . Per quanto riguarda gli archi ci sarà un arco diretto dal nodo  $s$  verso ogni nodo  $v_i$  con capacità pari al valore  $\lceil T_i \rceil$ . Ogni nodo studente è collegato con un arco diretto di capacità 1 ai nodi che rappresentano i giorni in cui deve andare a lezione. Infine tutti i nodi giorno sono collegati al nodo  $t$  con un arco diretto di capacità 1.

Riassumendo definiamo la nostra rete  $G = (V, E)$  nel seguente modo:

- per i nodi abbiamo:  $V = \{s, t\} \cup \{v_1, \dots, v_n\} \cup \{d_1, \dots, d_m\}$ ;
- per gli archi si ha che  $E$  include:
  - per  $i = 1, \dots, n$ , si hanno gli archi  $(s, v_i)$  con capacità  $\lceil T_i \rceil$ ;
  - per  $i = 1, \dots, n$  e  $j = 1, \dots, m$  si ha l'arco  $(v_i, d_j)$  con capacità 1 se e solo se lo studente  $i$  deve andare a lezione il giorno  $j$ ;
  - per  $j = 1, \dots, m$  si hanno gli archi  $(d_j, t)$  con capacità 1.

Nella figura seguente è rappresentata la rete ottenuta a partire dall'istanza data in esempio, dove gli archi senza valore associato hanno capacità 1:



Ora che abbiamo definito come opera la funzione di riduzione facciamo vedere come possiamo trovare una soluzione al problema dato trovando un  $max - flow$  su  $G$  che satura il taglio composto da tutti i nodi tranne  $t$ .

**Claim.**  $\exists$  un'assegnazione di automobili ammissibile  $\iff \exists$  un flusso massimo  $f^*$  su  $G$  tale che  $v(f^*) = m$   
 Dimostriamo ora l'affermazione precedente:

•  $\implies$

In questo caso facciamo vedere come a partire da un'assegnazione ammissibile di automobili è possibile definire un flusso massimo ammissibile su  $G$  di valore  $m$  e quindi che satura il taglio composto da tutti i nodi eccetto  $t$ .

Per gli archi del tipo  $(v_i, d_j)$  poniamo  $f(v_i, d_j) = 1$  se nell'assegnazione in nostro possesso lo studente  $i$  deve portare la macchina il giorno  $j$ , 0 altrimenti. Per ogni  $v_i$  sia  $x_i$  il numero di archi uscenti con flusso 1 abbiamo che  $x_i \leq \lceil T_i \rceil$  (in quanto l'assegnazione è ammissibile), per rispettare il vincolo di conservazione sul flusso dobbiamo fornire in ingresso ad ogni nodo  $v_i$ ,  $x_i$  unità di flusso, l'unico arco entrante nei nodi  $v_i$  è del tipo  $(s, v_i)$  quindi poniamo  $f(s, v_i) = x_i$ , per tali archi il vincolo sulle capacità viene rispettato in quanto la loro capacità era stata posta a  $\lceil T_i \rceil$ .

Ora mostriamo che su ogni arco del tipo  $(d_j, t)$  si ha il vincolo sulle capacità rispettato e flusso pari ad 1: ancora una volta essendo l'assegnazione ammissibile abbiamo che per ogni giorno uno ed un solo studente deve portare la macchina, quindi per ogni nodo  $d_j$  si avrà flusso entrante pari ad 1, per rispettare il vincolo di conservazione facciamo defluire tale unità di flusso sull'arco  $(d_j, t)$ , dato che le capacità di tali archi erano state poste ad 1 avremo che tutte le capacità sono state rispettate ed inoltre saranno tutti saturi. Da ciò deriva che abbiamo trovato un flusso massimo ammissibile di valore  $m$ .

- $\Leftarrow$

Mostriamo ora come da un flusso ammissibile di valore  $m$  è possibile ricavare un'assegnazione di automobili ammissibile. Dopo aver fatto "girare" un algoritmo di *max-flow* sulla nostra rete avremo che alcuni archi del tipo  $(v_i, d_j)$  avranno flusso 1 mentre altri 0. Costruiamo l'assegnazione richiesta assegnando la macchina allo studente  $i$  il giorno  $j$  se e solo se si ha che  $f(v_i, d_j) = 1$ . Mostriamo che l'assegnazione così ottenuta è ammissibile: Innanzitutto ogni studente  $i$  non porta la macchina più di  $\lceil T_i \rceil$  giorni in quanto essendo  $c(s, v_i) = \lceil T_i \rceil$  da ogni nodo  $v_i$  non possono uscire più di  $\lceil T_i \rceil$  archi con flusso 1. Banalmente gli studenti porteranno la macchina solo i giorni in cui devono effettivamente andare a lezione, in quanto i relativi nodi erano stati collegati solo a tali giorni. Manca ancora da mostrare che ogni giorno uno ed un solo studente porterà la macchina, osservando che il flusso ha valore  $m$  si ha che ogni arco del tipo  $(d_j, t)$  ha flusso pari ad uno quindi per il vincolo di conservazione del flusso si ha anche che per ogni nodo  $d_j$  si ha uno ed un solo arco entrante con flusso 1 (proveniente dal nodo associato all'unico studente che porterà la macchina quel giorno).  $\square$

Quindi ricapitolando, avendo mostrato che il problema è riducibile ad un problema di *max-flow*, al fine di trovare una soluzione, possiamo trasformare l'istanza data in un'istanza di *max-flow*, trovare un flusso massimo sulla rete ottenuta, e da tale flusso ricavare la soluzione per il problema di partenza come descritto nel secondo punto della dimostrazione precedente.

Sapete analizzare la complessità temporale dell'algoritmo?

## Problema 12

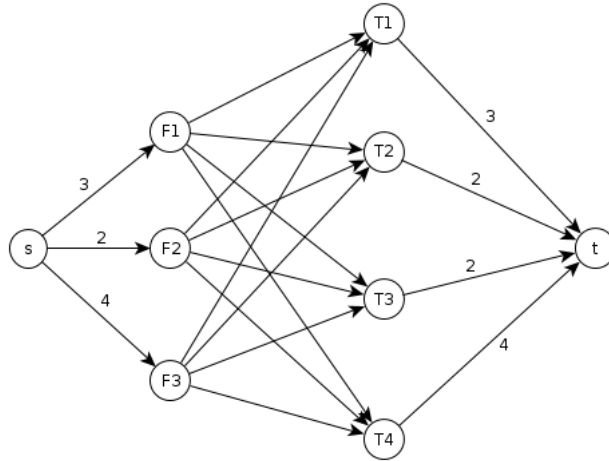
Si consideri il seguente problema. Un gruppo di famiglie  $\{F_1, \dots, F_n\}$  si incontra per una cena in un ristorante. Ogni famiglia  $F_i$  è composta da  $p_i$  persone. Il ristorante mette a disposizione un insieme di tavoli  $\{T_1, \dots, T_m\}$ , con il tavolo  $T_j$  adatto per  $t_j$  commensali. Si vogliono assegnare i posti a tavola in modo tale da far socializzare le famiglie, evitando quindi che due membri della stessa famiglia siedano allo stesso tavolo: ci chiediamo se esiste una possibile assegnazione che soddisfi questo vincolo.

**Soluzione** Come nell'esercizio precedente andremo a ridurre il problema ad un problema di *max-flow*. Data un'istanza del problema definiamo la rete  $G = (V, E)$  nel seguente modo: c'è un nodo  $F_i$  per ogni famiglia  $i$ , ed un nodo  $T_j$  per ogni tavolo  $j$ , sono inoltre presenti un nodo sorgente  $s$  ed un nodo target  $t$ . Aggiungiamo un arco dal nodo  $s$  verso ogni nodo famiglia mettendo capacità pari al numero di componenti della famiglia associata, colleghiamo ogni nodo famiglia con tutti i nodi tavolo ponendo capacità 1, infine colleghiamo ogni nodo tavolo al nodo  $t$  con capacità pari alla capienza del tavolo.

Riassumendo definiamo la nostra rete  $G = (V, E)$  nel seguente modo:

- per i nodi abbiamo:  $V = \{s, t\} \cup \{F_1, \dots, F_n\} \cup \{T_1, \dots, T_m\}$ ;
- per gli archi si ha che  $E$  include:
  - per  $i = 1, \dots, n$ , si hanno gli archi  $(s, F_i)$  con capacità  $p_i$ ;
  - per  $i = 1, \dots, n$  e  $j = 1, \dots, m$  si ha l'arco  $(F_i, T_j)$  con capacità 1;
  - per  $j = 1, \dots, m$  si hanno gli archi  $(T_j, t)$  con capacità  $t_j$ ;

Nella figura seguente è rappresentata la rete ottenuta a partire da un esempio così definito: abbiamo tre famiglie con  $p_1 = 3, p_2 = 2, p_3 = 4$  e quattro tavoli con  $t_1 = 3, t_2 = 2, t_3 = 2, t_4 = 4$ . Gli archi senza valore associato hanno capacità 1.



Ora che abbiamo definito come opera la funzione di riduzione facciamo vedere come possiamo trovare una soluzione al problema dato trovando un *max-flow* su  $G$  che satura il taglio composto dal solo nodo  $s$  e quindi di valore pari al numero di persone totali da mettere a tavola.

**Claim.**  $\exists$  un'assegnazione di persone ai tavoli  $\iff \exists$  un flusso massimo  $f^*$  su  $G$  tale che  $v(f^*) = \sum_{i=1}^n p_i$

Dimostriamo ora l'affermazione precedente:

•  $\implies$

In questo caso facciamo vedere come a partire da una assegnazione ammissibile di persone ai tavoli è possibile definire un flusso massimo ammissibile su  $G$  di valore  $\sum_{i=1}^n p_i$  e quindi che satura il taglio composto dal solo nodo  $s$ .

Per gli archi del tipo  $(F_i, T_j)$  poniamo  $f(F_i, T_j) = 1$  se nell'assegnazione in nostro possesso una persona della famiglia  $i$  siede al tavolo  $j$ , 0 altrimenti. Per ogni nodo del tipo  $T_j$  non ci sono più di  $t_j$  archi entranti con flusso 1 (visto che siamo partiti da un'assegnazione ammissibile), quindi è possibile far defluire il flusso entrante verso il nodo  $t$  in quanto la capacità dell'arco uscente dal generico nodo  $T_j$  è stata posta proprio a  $t_j$ . Per quanto riguarda i nodi  $F_i$ , ognuno avrà  $p_i$  archi uscenti con flusso 1, per rispettare il vincolo di conservazione del flusso quindi poniamo a  $p_i$  il flusso sull'arco  $(s, F_i)$  quindi avremo che tutte le capacità sugli archi uscenti da  $s$  sono rispettati e tutti tali archi saranno saturi dandoci un flusso di valore  $\sum_{i=1}^n p_i$ .

•  $\impliedby$

Mostriamo ora come da un flusso ammissibile di valore  $\sum_{i=1}^n p_i$  è possibile ricavare un'assegnazione di persone ai tavoli ammissibile. Dopo aver fatto "girare" un algoritmo di *max-flow* sulla nostra rete avremo che alcuni archi del tipo  $(F_i, T_j)$  avranno flusso 1 mentre altri 0. Costruiamo l'assegnazione richiesta assegnando una persona della famiglia  $i$  al tavolo  $j$  se e solo se si ha che  $f(F_i, T_j) = 1$ . Mostriamo che l'assegnazione così ottenuta è ammissibile: Dato che per ogni nodo  $F_i$  c'è un solo arco verso ogni nodo  $T_j$  avremo che due persone della stessa famiglia non sono state assegnate allo stesso tavolo.

Mostriamo ora che le capacità dei tavoli sono rispettate: dato che il flusso è ammissibile sugli archi del tipo  $(T_j, t)$  non passa flusso superiore a  $t_j$  e quindi per il vincolo di conservazione sul flusso in ogni nodo del tipo  $T_j$  non entrano più di  $t_j$  archi con flusso 1.

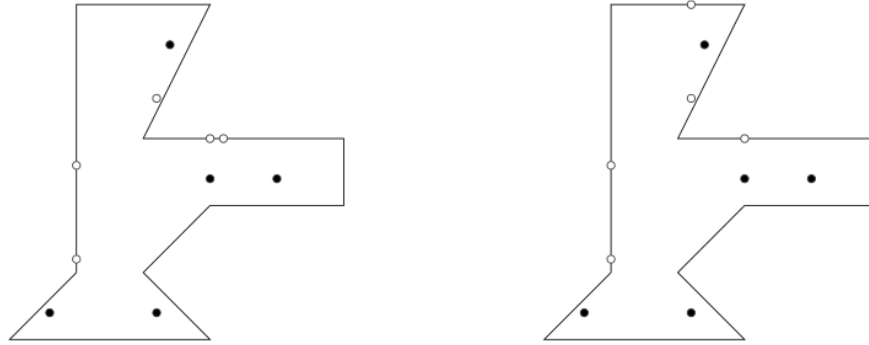
Mostriamo infine come tutte le persone vengono correttamente assegnate a un tavolo: dato che il flusso satura il taglio composto dal solo nodo  $s$  avremo che per ogni nodo  $F_i$  ci saranno  $p_i$  unità di flusso entranti, per il vincolo di conservazione questo flusso in uscita è stato ripartito in  $p_i$  archi con flusso 1, facendo così in modo che tutte le persone sono state correttamente assegnate a un qualche tavolo.  $\square$

Quindi ricapitolando, avendo mostrato che il problema è riducibile ad un problema di *max-flow*, al fine di trovare una soluzione, possiamo trasformare l'istanza data in un'istanza di *max-flow*, trovare un flusso massimo sulla rete ottenuta, e da tale flusso ricavare la soluzione per il problema di partenza come descritto nel secondo punto della dimostrazione precedente.

## Problema 13

Si consideri il seguente problema. Data una pianta di un appartamento con la posizione di  $n$  punti luce ed  $n$  interruttori, si vuole determinare se essa è "ergonomica", nel senso che è possibile collegare ciascun interruttore a un unico punto luce in modo tale che ogni punto luce sia visibile dall'interruttore che lo comanda e che ogni punto luce sia comandato da un solo interruttore.

In figura, a sinistra è mostrata una pianta per la quale ciò è possibile, mentre non lo è in quella a destra (i punti neri rappresentano i punti luce mentre i punti bianchi rappresentano gli interruttori).



Assumendo di avere un sottoprogramma che, dati due segmenti nel piano, determina in tempo  $O(1)$  se si intersecano, descrivere un algoritmo che, data una pianta, determina se è ergonomica in tempo polinomiale in  $m$ ,  $n$  ed il numero delle pareti.

**Soluzione** Il problema ci sta chiedendo se è possibile trovare una funzione biunivoca tra l'insieme degli interruttori e quello dei punti luce con il vincolo che un interruttore può essere associato solo ad un sottoinsieme dei punti luce (quelli da lui visibili). Pensandoci bene il problema coincide con il problema del matching perfetto su un grafo bipartito, dove l'insieme dei nodi è costituito dall'unione tra l'insieme degli interruttori e l'insieme dei punti luce. Ognuno dei possibili  $n^2$  archi è aggiunto al grafo se e soltanto se non c'è alcuna parete che interseca il segmento *interruttore - puntoluca*. Tale test richiede tempo lineare nel numero delle pareti. Quindi data una pianta riusciamo a costruire il relativo grafo in tempo  $O(n^2 \cdot \text{numeroPareti})$  quindi polinomiale nella dimensione dell'istanza. Come visto a lezione dato un grafo bipartito  $G = (V = (V_1, V_2), E)$  per trovare un matching perfetto su di esso possiamo ricondurci ad un problema di max-flow. La rete è derivata aggiungendo un nodo  $s$  collegato a tutti i nodi di  $V_1$  tramite un arco diretto; si orientano tutti gli archi tra i due sottoinsiemi verso i nodi in  $V_2$ . Si aggiunge un nodo target  $t$  e si collegano tutti i nodi in  $V_2$  a quest'ultimo nodo tramite un arco diretto.

La capacità di ogni arco viene impostata ad uno.

Facendo "girare" un algoritmo per il calcolo del *max-flow*, se otteniamo un flusso di valore  $n$  (dove  $n = |V_1| = |V_2|$ ), sul grafo originale è possibile costruire un matching perfetto, in particolare tale matching è costituito dagli archi tra nodi di  $V_1$  e  $V_2$  che avranno flusso pari ad uno.

Nelle figure sottostanti infine è riportato un esempio che mostra come una piantina viene modellata in un grafo bipartito e come quest'ultimo viene ridotto a un'istanza di *max-flow* che risolve il problema del matching perfetto.

