Metodi e Modelli per l'Ottimizzazione Combinatoria

Il problema del flusso di costo minimo

L. De Giovanni G. Zambelli

1 Problema del flusso a costo minimo

Il problema del flusso a costo minimo è definito come segue.

Dato un grafo orientato $D = (V, A)$, sia $n = |V|$, $m = |A|$. Sono dati *costi* c_e per ogni arco $e \in A$, *capacità* u_e per ogni $e \in A$, e *domande* b_v su ogni nodo $v \in V$.

Vogliamo assegnare ad ogni arco $e \in A$ un numero nonnegativo x_e, che sará il *flusso* portato dall'arco e, in modo tale che x_e non ecceda la capacità u_e e, per ogni $v \in V$, il flusso totale portato dagli archi che entrano in v meno il flusso totale portato dagli archi che escono da v sia uguale a b_v. Un vettore $(x_e)_{e\in A}$ che soddisfa tali vincoli è detto un *flusso ammissibile*. Se la domanda b_v su un nodo v è positiva, allora il flusso entrante deve essere maggiore del flusso uscente, se b_v è negativa allora il flusso entrante deve essere minore del flusso uscente, se $b_v = 0$ allora il flusso che entra in v è uguale al flusso che esce. Il flusso x_e sull'arco e incorre un costo $c_e x_e$ (ovvero c_e il costo per unitá di flusso portato dall'arco e). Il problema del *flusso a costo minimo* consiste nel determinare un flusso ammissibile di costo totale minimo. Se, inoltre, richiediamo che i flussi siano interi, ovvero $x_{vw} \in \mathbb{Z}$ per ogni $(v, w) \in A$, allora parleremo del *problema del flusso intero a costo minimo*. La Figura 1 mostra un esempio di flusso ammissibile.

Il problema del flusso a costo minimo puó essere formulato come un problema di programmazione lineare come segue:

$$\begin{array}{rcll} & \min \sum_{(v,w)\in A} c_{vw}x_{vw} & & \\ \sum_{w\,:\,(w,v)\in A} x_{wv} - \sum_{w\,:\,(v,w)\in A} x_{vw} & = & b_v & v \in V \\ & 0 \le x_{vw} \le u_{vw} & & (v,w) \in A \end{array} \qquad (1)$$

I vincoli di uguaglianza sono detti *vincoli di bilanciamento sui nodi*.

Per il problema del flusso intero a costo minimo, abbiamo gli ulteriori vincoli che $x_{vw} \in \mathbb{Z}$ per ogni $(v, w) \in A$. Si noti che, in forma matriciale, il problema (5) può essere scritto nella forma

$$\begin{array}{c} \max c^T x \\ A(d)\, x = b \\ 0 \le x \le u, \end{array}$$

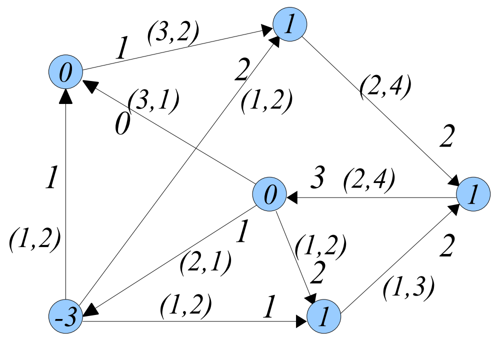


Figura 1: Esempio di un flusso ammissibile. I numeri all'interno dei nodi rappresentano le domande b_v, le coppie ordinate di numeri vicino ad ogni arco rappresentano, nell'ordine, costo e capacitá, (c_e, u_e). I numeri sugli archi sono i flussi. Si noti che il flusso ha costo totale 23.

ove $A(D)$ è la matrice di incidenza del grafo (orientato) D, e dove denotiamo con b, c, u, rispettivamente, il vettore delle domande, dei costi e delle capacitá. Poichè la matrice $A(D)$ è totalmente unimodulare (vedi Teorema 4 delle note sul problema dell'assegnamento), tutte le soluzioni di base del sistema precedente sono intere, qualora i vettori b ed u siano interi. Poichè nel problema del flusso intero a costo minimo le domande e le capacitá devono essere intere, questo ci dice che i vincoli di interezza sulle variabili intere non rappresentano una ulteriore difficoltà.

Anche se in principio il problema del flusso a costo minimo puó essere risolto con qualunque algoritmo per la programmazione lineare (ad esempio con il metodo del simplesso), esistono algoritmi ad-hoc che sono molto piú efficienti. Descriveremo dunque un algoritmo per risolvere il problema del flusso a costo minimo. Qualora $b_v \in \mathbb{Z}$ per ogni $v \in V$ e $u_e \in \mathbb{Z}$ per ogni $e \in A$, la soluzione determinata da tale algoritmo sará intera.

Il problema del flusso a costo minimo generalizza svariati problemi di ottimizzazione combinatoria. Ne menzioniamo due.

1.1 Problemi di cammini minimi

Dato un grafo orientato $D = (V, A)$, due nodi $s, t \in V$, $s \neq t$, e lunghezze ℓ_e su ogni arco $e \in A$, vogliamo trovare il cammino da s a t di lunghezza minima. Definiamo variabili binarie x_e per ogni $e \in A$, ove

$$x_e = \begin{cases} 1 & \text{se } e \text{ è nel cammino minimo da } s \text{ a } t, \\ 0 & \text{altrimenti} \end{cases} \qquad e \in A.$$

Il problema del cammino minimo diventa dunque

$$\begin{array}{rcll} \min \sum_{(v,w)\in A} \ell_{vw} x_{vw} & & & \\ \sum_{w\,:\,(w,v)\in A} x_{wv} - \sum_{w\,:\,(v,w)\in A} x_{vw} & = & 0 & v \in V \setminus \{s,t\} \\ \sum_{w\,:\,(w,s)\in A} x_{ws} - \sum_{w\,:\,(s,w)\in A} x_{sw} & = & -1 & \\ \sum_{w\,:\,(w,t)\in A} x_{wt} - \sum_{w\,:\,(t,w)\in A} x_{tw} & = & +1 & \\ x_{vw} \geq 0 & & & (v,w) \in A \\ x_{vw} \in \mathbb{Z} & & & (v,w) \in A \end{array}$$

che è un problema di flusso (intero) di costo minimo.

Il problema del cammino minimo può essere risolto in tempo $O(nm)$ usando l'algoritmo di Bellman-Ford. Ricordiamo che tale algoritmo risolve il seguente problema più generale: dato $r \in V$, trova i cammini di lunghezza minima da r a v per ogni $v \in V$. L'algoritmo di Bellman-Ford è in grado di risolvere tale problema se D non contiene cicli di lunghezza negativa. Quando termina, l'algoritmo di Bellman-Ford può dare uno dei due output seguenti:

1. per ogni $v \in V$, la distanza minima y_v da r a v,

   oppure

2. un ciclo C di lunghezza negativa, cioè un ciclo C tale che $\sum_{e\in C} \ell_e < 0$.

L'algoritmo fornisce l'output 1. solo se non ci sono cicli negativi, poiché altrimenti la lunghezza minima di un cammino da r a v non è ben definita (infatti percorrere un ciclo negativo diminuisce la lunghezza totale percorsa). Se non ci sono cicli negativi, allora gli elementi y_v, $v \in V$, soddisfano le seguenti condizioni, dette *condizioni di Ford*:

$$y_w - y_v \leq \ell_{vw}, \qquad \text{per ogni } (v,w) \in A. \tag{2}$$

Infatti, per ogni arco (v,w), dato un cammino da r a v di lunghezza y_v, allora, poiché w può essere raggiunto da v mediante un arco di lunghezza ℓ_{vw} e poiché non vi sono cicli negativi, esiste un cammino da r a w di lunghezza al più $y_v + \ell_{vw}$, e dunque $y_w \leq y_v + \ell_{vw}$.

1.2 Problema del flusso massimo

Dato un grafo orientato $D = (V,A)$, due nodi $s,t \in V$, $s \neq t$, e capacità c_e su ogni arco $e \in A$, un *flusso da* s *a* t è un vettore $(f_e)_{e\in A}$ tale che $0 \leq f_e \leq u_e$ e che il flusso che entra in ogni nodo $v \in V \setminus \{s,t\}$ sia uguale al flusso che esce.

Il *valore* del flusso f da s a t, denotato da $val(f)$, è il flusso totale uscente da s meno il flusso che entra in s (che è uguale al flusso che entra in t meno il flusso che esce). Vogliamo trovare un flusso di valore massimo.

Dunque, dobbiamo risolvere il seguente problema.

$$\begin{array}{rcll} \max \sum_{w\,:\,(s,w)\in A} f_{sw} - \sum_{w\,:\,(w,s)\in A} f_{ws} & & & \\ \sum_{w\,:\,(w,v)\in A} f_{wv} - \sum_{w\,:\,(v,w)\in A} f_{vw} & = & 0 & v \in V \setminus \{s,t\} \\ 0 \leq f_{vw} \leq u_{vw} & & & (v,w) \in A \end{array} \tag{3}$$

Possiamo costruire un problema di flusso a costo minimo in un qualche grafo diretto, la cui soluzione fornisce una soluzione al problema del flusso massimo. La costruzione è la seguente.

Consideriamo il grafo $\tilde{D} = (V, \tilde{A})$, con gli stessi nodi di D, e in cui $\tilde{A} = A \cup \{\tilde{e}\}$, ove $\tilde{e}$ è un arco da t ad s. Definiamo costi c_e, $e \in \tilde{A}$, come segue

$$c_e = \begin{cases} -1 & \text{se } e = \tilde{e}, \\ 0 & \text{altrimenti} \end{cases} \qquad e \in \tilde{A}.$$

Definiamo richieste $b_v = 0$ per ogni $v \in V$. Il problema di flusso a costo minimo così definito in $\tilde{D}$ è

$$\begin{array}{rcll} \min -x_{\tilde{e}} & & & \\ \sum_{w\,:\,(w,v)\in A} x_{wv} - \sum_{w\,:\,(v,w)\in A} x_{vw} & = & 0 & v \in V \setminus \{s,t\} \\ \sum_{w\,:\,(w,s)\in A} x_{ws} + x_{\tilde{e}} - \sum_{w\,:\,(s,w)\in A} x_{sw} & = & 0 & \\ \sum_{w\,:\,(w,t)\in A} x_{wt} - \sum_{w\,:\,(t,w)\in A} x_{tw} - x_{\tilde{e}} & = & 0 & \\ 0 \leq x_{vw} \leq u_{vw} & & & (v,w) \in A \\ x_{\tilde{e}} \geq 0 & & & \end{array} \tag{4}$$

Si noti che, data una soluzione x di (4), allora il vettore f definito da $f_e = x_e$, $e \in A$ definisce un flusso da s a t in D, il cui valore è uguale a $x_{\tilde{e}}$.
Viceversa, dato un flusso f da s a t, se definiamo $x_e = f_e$ per ogni $e \in A$ e $x_{\tilde{e}} = \sum_{w\,:\,(s,w)\in A} f_{sw} - \sum_{w\,:\,(w,s)\in A} f_{ws}$, allora x è una soluzione ammissibile per (4) di costo $-val(f)$.

Dunque, il valore ottimo di (3) è uguale a meno il valore ottimo di (4).

2 L'algoritmo di cycle canceling

Descriviamo ora un algoritmo per risolvere il problema del flusso a costo minimo (5). Tale algoritmo partirà da un qualche flusso ammissibile, e ad ogni iterazione cerca un flusso ammissibile di costo minore fino ad arrivare all'ottimo.

Supponiamo di avere un flusso ammissibile $\tilde{x}$. Discuteremo il problema di trovare un flusso ammissibile da cui iniziare nella Sezione 2.1.

Un ciclo non orientato C di D è una sequenza alternata di nodi ed archi

$$C = v_0, e_1, v_1, e_2, v_2, \ldots, v_{k-1}, e_k, v_k$$

tale che $v_0 = v_k$ e, per $i = 1, \ldots, k$, e_i è un arco con estremi v_{i-1} e v_i.

Se $e_i = (v_{i-1}, v_i)$, allora diciamo che e_i è un *arco forward*, se $e_i = (v_i, v_{i-1})$, allora diciamo che e_i è un *arco backward*.

Definiamo un nuovo flusso $\bar{x}$ a partire da $\tilde{x}$ come segue:

$$\bar{x}_e = \begin{cases} \tilde{x}_e + \epsilon & \text{se } e \text{ è forward in } C, \\ \tilde{x}_e - \epsilon & \text{se } e \text{ è backward in } C, \\ \tilde{x}_e & \text{se } e \notin C, \end{cases} \qquad e \in A.$$

Si noti che, se $\bar{x}$ così definito soddisfa i vincoli di bilanciamento sui nodi. Affinché $0 \leq \bar{x}_e \leq u_e$ per ogni $e \in A$, dobbiamo avere

$$\begin{array}{ll} \epsilon \leq u_e - \tilde{x}_e & \text{per ogni arco } e \text{ forward in } C, \\ \epsilon \leq \tilde{x}_e & \text{per ogni arco } e \text{ backward in } C, \end{array}$$

e dunque il valore massimo che possiamo assegnare a ϵ è

$$\epsilon := \min \left\{ \min\{u_e - \tilde{x}_e \, : \, e \text{ forward in } C\}, \min\{\tilde{x}_e \, : \, e \text{ backward in } C\} \right\}.$$

Diremo che $\bar{x}$ è ottenuto *aumentando $\tilde{x}$ lungo C*.

Si noti che la differenza tra il costo di $\bar{x}$ e il costo di $\tilde{x}$ è

$$\sum_{e \in A} c_{vw}(\bar{x}_e - \tilde{x}_e) = \epsilon(\sum_{\substack{e \in A \\ e\, forward}} c_e - \sum_{\substack{e \in A \\ e\, backward}} c_e)$$

Dunque $\bar{x}$ è una soluzione di costo minore di $\tilde{x}$ se e solo se $\epsilon > 0$ e

$$\sum_{\substack{e \in A \\ e\, forward}} c_e - \sum_{\substack{e \in A \\ e\, backward}} c_e < 0.$$

Definizione 1 *Dato un flusso ammissibile $\tilde{x}$, un ciclo non orientato C di D è detto un* ciclo migliorante per $\tilde{x}$ *se*

$$\begin{array}{ll} u_e - \tilde{x}_e > 0 & \textit{per ogni arco e forward in } C, \\ \tilde{x}_e > 0 & \textit{per ogni arco e backward in } C, \end{array}$$

e se

$$\sum_{\substack{e \in A \\ e\, forward}} c_e - \sum_{\substack{e \in A \\ e\, backward}} c_e < 0.$$

Abbiamo dunque visto che, se aumentiamo $\tilde{x}$ lungo un ciclo migliorante, otteniamo un nuovo flusso ammissibile di costo strettamente minore del precedente. Questo suggerisce il seguente algoritmo, detto di *cycle canceling*.

Algoritmo di cycle canceling

Sia dato un flusso ammissibile $\tilde{x}$.
Finché esiste un ciclo migliorante C per $\tilde{x}$,
aumenta $\tilde{x}$ lungo C;

Vi sono due problemi da considerare:

- Come trovare un ciclo migliorante,
- Dimostrare che, se non ci sono cicli miglioranti per $\tilde{x}$, allora $\tilde{x}$ è un flusso di costo minimo.

Per trovare un ciclo migliorante per $\tilde{x}$, costruiamo un nuovo grafo orientato $D(\tilde{x}) = (V, A(\tilde{x}))$, detto *grafo residuo.*

Definiamo gli archi in $D(\tilde{x})$ e delle lunghezze ℓ su tali archi come segue. Per ogni arco $(v, w) \in A$, il grafo residuo $D(\tilde{x})$ contiene gli archi

$$\begin{array}{lll} (v,w) & \text{se } u_{vw} - \tilde{x}_{vw} > 0, & \text{di costo } \ell_{vw} = c_{vw} \\ (w,v) & \text{se } \tilde{x}_{vw} > 0, & \text{di costo } \ell_{wv} = -c_{vw}. \end{array}$$

Proposizione 1 *D contiene un ciclo migliorante per $\tilde{x}$ se e solo se $D(\tilde{x})$ contiene un ciclo orientato di lunghezza negativa.*

Pertanto, trovare un ciclo migliorante consiste nel trovare un ciclo orientato di lunghezza negativa in $D(\tilde{x})$. Questo può essere fatto applicando l'algoritmo di Bellman-Ford. Infatti, una volta costruito $D(\tilde{x})$, poniamo la lunghezza di ogni arco $e \in A(\tilde{x})$ uguale a ℓ_e definito in precedenza. Fissato un nodo r a piacere, applichiamo l'algoritmo di Bellman-Ford per determinare i cammini di lunghezza minima da r agli altri nodi. Se l'esecuzione dell'algoritmo di Bellman-Ford restituisce tali cammini, allora $D(\tilde{x})$ non ha nessun ciclo orientato di lunghezza negativa. Altrimenti l'algoritmo ritorna un ciclo orientato di lunghezza negativa, che corrisponde ad un ciclo migliorante per $\tilde{x}$.

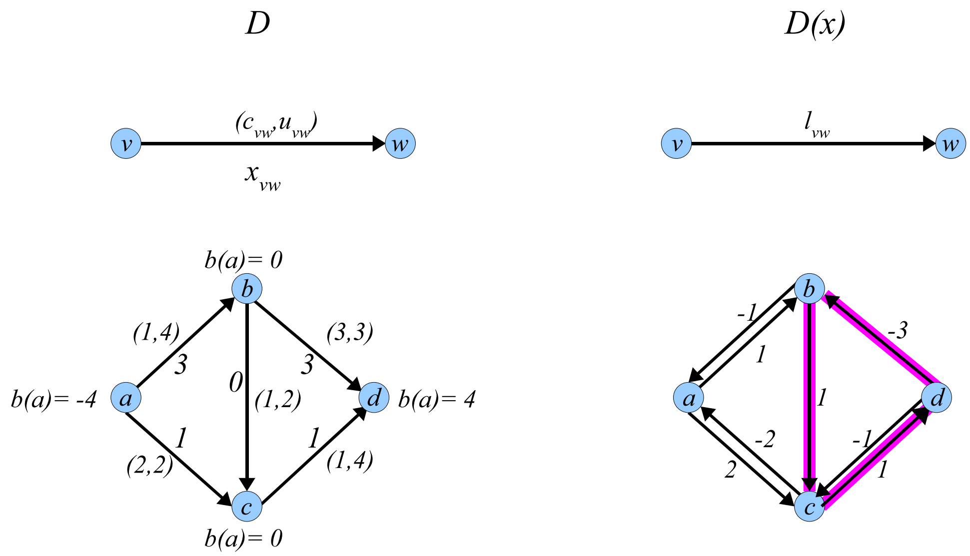
Illustriamo l'algoritmo di cycle canceling nell'esempio in Figura 2.

Osserviamo che, se $b_v \in \mathbb{Z}$ per ogni $v \in V$ e $u_e \in \mathbb{Z}$ per ogni $e \in A$, ed eseguiamo l'algoritmo di cycle canceling a partire da un flusso ammissibile $\tilde{x}$ intero, allora ad ogni iterazione il flusso su ogni arco cambierà di una quantità intera, e dunque la soluzione ottima finale sarà intera. Nella sezione 3, mostreremo come utilizzare l'algoritmo di cycle canceling per ottenere un flusso ammissibile, e osserveremo che, se $b_v \in \mathbb{Z}$ per ogni $v \in V$ e $u_e \in \mathbb{Z}$ per ogni $e \in A$, allora il flusso ammissibile che otterremo sarà intero. Questo dimostra dunque il fatto seguente.

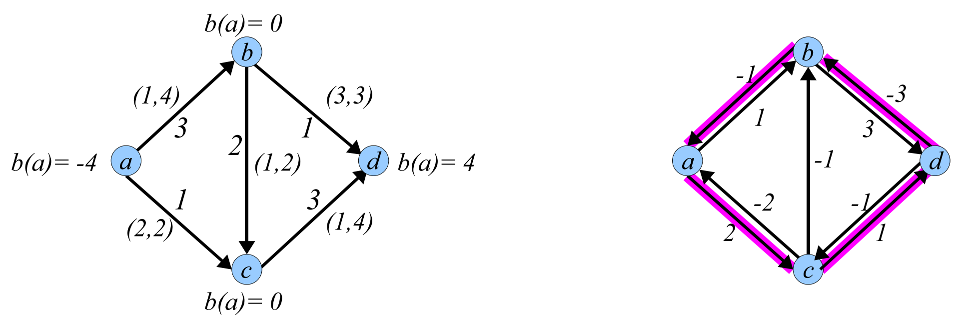
Teorema 1 *Se $b_v \in \mathbb{Z}$ per ogni $v \in V$, $u_e \in \mathbb{Z}$ per ogni $e \in A$, e se il problema di minimum cost flow (5) ammette una soluzione ottima, allora esiste una soluzione ottima che è intera.*

Dunque l'algoritmo di cycle canceling può essere utilizzato anche per risolvere il problema del flusso intero di costo minimo.

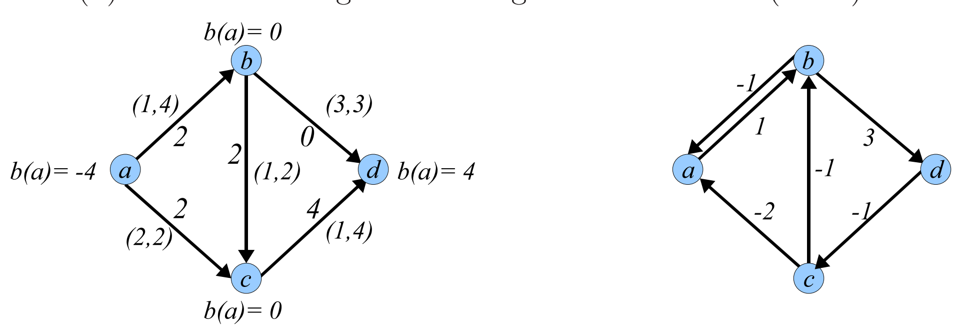
Per quanto riguarda il running-time dell'algoritmo di cycle canceling, si osservi che ogni iterazione richiede la costruzione del grafo residuo, che richiede tempo $O(n + m)$, e la ricerca di un ciclo orientato di lunghezza negativa nel grafo residuo, che richiede tempo $O(nm)$ usando l'algoritmo di Bellman-Ford. Pertanto, ogni iterazione richiede tempo



(a) Aumenta x lungo il ciclo migliorante b-c-d-b ($\epsilon = 2$).



(b) Aumenta x lungo il ciclo migliorante a-c-d-b-a ($\epsilon = 1$).



(c) $D(x)$ non contiene cicli orientati di lunghezza negativa. Il flusso trovato è di costo minimo.

Figura 2: Esecuzione dell'algoritmo di cycle canceling: la colonna a sinistra rappresenta il grafo originale con il flusso ad ogni iterazione, quella a destra rappresenta il grafo residuo relativo al flusso corrente, con le lunghezze ℓ sugli archi.

$O(nm)$. Se $b_v \in \mathbb{Z}$ per ogni $v \in V$, $c_e, u_e \in \mathbb{Z}$ per ogni $e \in A$, allora ad ogni iterazione il costo del flusso decresce almeno di 1 unità, pertanto, se c^* è il costo minimo di un flusso e $\tilde{x}$ è il flusso ammissibile di partenza, l'algoritmo termina in al più $\sum_{e\in A} c_e\tilde{x}_e - c^*$ iterazioni. Si noti che, se $U = \max\{|u_e| \,:\, e \in A\}$ e $C = \max\{|c_e| \,:\, e \in A\}$, allora $\sum_{e\in A} c_e\tilde{x}_e - c^* \leq 2mCU$, e dunque il running time dell'algoritmo è $O(nm^2CU)$. Questo non è un bound polinomiale nella taglia dell'input, tuttavia accorgimenti ulteriori nella scelta del ciclo migliorante permettono di ridurre la complessità dell'algoritmo di cycle canceling a $O(nm\log(n)\log(nC))$.

Il teorema seguente dimostra che l'algoritmo è corretto, ovvero che, se non ci sono cicli miglioranti per $\tilde{x}$, allora $\tilde{x}$ è un flusso di costo minimo.

Teorema 2 *Sia $\tilde{x}$ un flusso ammissibile. Allora $\tilde{x}$ è ottimo se e solo se non esistono cicli miglioranti per $\tilde{x}$*

Dimostrazione: Abbiamo già dimostrato che, se esiste un ciclo migliorante per $\tilde{x}$, allora $\tilde{x}$ non è ottima.

Dimostriamo ora che, se non ci sono cicli miglioranti per $\tilde{x}$, allora $\tilde{x}$ è un flusso di costo minimo.

La dimostrazione utilizza le condizioni di ottimalità per (5) che si evincono dal duale. Il problema di flusso a costo minimo può essere scritto come segue

$$
\begin{array}{lrcll}
 & \min \sum_{(v,w)\in A} c_{vw}x_{vw} & & & \\
(\text{var. duale: } y_v) & \sum_{w\,:\,(w,v)\in A} x_{wv} - \sum_{w\,:\,(v,w)\in A} x_{vw} & = & b_v & v \in V \\
(\text{var. duale: } z_{vw}) & -x_{vw} \geq -u_{vw} & & & (v,w) \in A \\
 & x_{vw} \geq 0 & & & (v,w) \in A
\end{array}
\tag{5}
$$

ove abbiamo scritto, a sinistra di ogni vincolo, la corrispondente variabile duale. Dunque y_v è la variabile duale relativa al vincolo di bilanciamento per v, $v \in V$, mentre z_e è la variabile duale corrispondente al vincolo $-x_e \geq -u_e$, $e \in A$. Il problema duale è dunque

$$
\begin{array}{rl}
\max \sum_{v\in V} b_v y_v - \sum_{(v,w)\in A} u_{vw}z_{vw} & \\
y_w - y_v - z_{vw} \leq c_{vw} & (v,w) \in A \\
z_{vw} \geq 0 & (v,w) \in A.
\end{array}
\tag{6}
$$

Si noti che le variabili z_e, $e \in A$, sono in qualche misura "inutili", nel senso che in una soluzione ammissibile duale (y,z) deve valere $z_{vw} \geq \max\{0, y_w - y_v - c_{vw}\}$, e dunque in una soluzione ottima duale $z_{vw} = \max\{0, y_w - y_v - c_{vw}\}$. Pertanto, possiamo considerare solo soluzioni duali ammissibili (y,z) in cui $z_{vw} = \max\{0, y_w - y_v - c_{vw}\}$.

Per il Teorema di Dualità forte, $\tilde{x}$ è ottimo se e solo esiste una soluzione ammissibile per il duale $(\tilde{y},\tilde{z})$ tale che $\sum_{(v,w)\in A} c_{vw}\tilde{x}_{vw} = \sum_{v\in V} b_v\tilde{y}_v - \sum_{(v,w)\in A} u_{vw}\tilde{z}_{vw}$.

Ricordiamo che questo avviene se e solo se $\tilde{x}$ e $(\tilde{y},\tilde{z})$ soddisfano le condizioni di ortogonalità. Per tale problema, le condizioni di ortogonalità sono le seguenti:

(i) Per ogni $(v,w) \in A$ tale che $\tilde{x}_{vw} > 0$, vale $\tilde{y}_w - \tilde{y}_v - \tilde{z}_{vw} = c_{vw}$, (*ii*) Per ogni $(v, w) \in A$ tale che $\tilde{z}_{vw} > 0$, vale $\tilde{x}_{vw} = u_{vw}$.

La condizione (*i*) implica che $\tilde{y}_w - \tilde{y}_v \geq c_{vw}$, poiché $\tilde{z}_{vw} \geq 0$.
La condizione (*ii*) è equivalente alla seguente: per ogni $(v, w) \in A$ tale che $\tilde{x}_{vw} < u_{vw}$, vale $\tilde{z}_{vw} = 0$. Si noti che, poichè $z_{vw} = \max\{0, y_w - y_v - c_{vw}\}$, abbiamo che $z_{vw} = 0$ se e solo se $\tilde{y}_w - \tilde{y}_v \leq c_{vw}$. Dunque, (*ii*) è equivalente alla condizione che, per ogni $vw \in A$ tale che $\tilde{x}_{vw} < u_{vw}$, vale $\tilde{y}_w - \tilde{y}_v \leq c_{vw}$.

Dunque, se $\tilde{x}$ è una soluzione ottima, allora esiste $\tilde{y}$ che soddisfa

$$\tilde{y}_w - \tilde{y}_v \geq c_{vw} \qquad \text{per ogni } (v, w) \in A \text{ tale che } \tilde{x}_{vw} > 0 \tag{7}$$

$$\tilde{y}_w - \tilde{y}_v \leq c_{vw} \qquad \text{per ogni } (v, w) \in A \text{ tale che } \tilde{x}_{vw} < u_{vw} \tag{8}$$

Viceversa, dato un flusso ammissibile $\tilde{x}$ e dato $\tilde{y}$ che soddisfi (7) e (8), allora se definiamo $\tilde{z}_{vw} = \max\{0, \tilde{y}_w - \tilde{y}_v - c_{vw}\}$ non è difficile verificare che $(\tilde{y}, \tilde{z})$ è ammissibile per il duale e che $\tilde{x}$ e $(\tilde{y}, \tilde{z})$ soddisfano le condizioni di ortogonalità (*i*),(*ii*), e che dunque $\tilde{x}$ è un flusso ottimo.

Pertanto, $\tilde{x}$ è un flusso ottimo se e solo se esiste $\tilde{y} \in \mathbb{R}^V$ che soddisfa le condizioni (7) e (8).

Ora, supponiamo che non esista un ciclo migliorante per $\tilde{x}$, ovvero che $D(\tilde{x})$ non contenga un ciclo di lunghezza negativa rispetto alle lunghezze ℓ_e ($e \in A(\tilde{x})$) definite in precedenza. Vogliamo dimostrare che $\tilde{x}$ è ottima. Per fare ció definiremo una soluzione duale $\tilde{y}$ tale che $\tilde{x}$ e $\tilde{y}$ soddisfino (7) e (8).

Fissato un qualunque nodo $r \in V$, applichiamo l'algoritmo di Bellman-Ford su $D(\tilde{x})$ con lunghezze ℓ_e. Poiché $D(\tilde{x})$ non contiene cicli di lunghezza negativa, l'algoritmo terminerá avendo determinato la lunghezza del cammino minimo da r a v per ogni nodo $v \in V$. Chiamiamo $\tilde{y}_v$ la lunghezza minimo di un cammino da r a v in $D(\tilde{x})$. Abbiamo visto nella Sezione 1.1 che $\tilde{y}$ soddisfa le condizioni di Ford (2). Le condizioni di Ford nel nostro caso sono:

$$\tilde{y}_w - \tilde{y}_v \leq \ell_{vw}, \quad \text{per ogni } (v, w) \in A(\tilde{x}).$$

Notiamo che tali condizioni implicano che $\tilde{y}$ soddisfa (7) e (8), e che dunque $\tilde{x}$ è un flusso ottimo. Infatti, dato $(v, w) \in A$, se $\tilde{x}_{vw} > 0$ allora $(w, v) \in A(\tilde{x})$ e $\ell_{wv} = -c_{vw}$, pertanto la condizione di Ford diviene $\tilde{y}_v - \tilde{y}_w \leq -c_{vw}$, che è equivalente a (7); se invece $\tilde{x}_{vw} < u_{vw}$ allora $(v, w) \in A(\tilde{x})$ e $\ell_{vw} = c_{vw}$, pertanto la condizione di Ford diviene $\tilde{y}_w - \tilde{y}_v \leq c_{vw}$, che è (8). $\square$

2.1 Trovare un flusso ammissibile

Rimane da considerare il problema di come determinare un flusso ammissibile per iniziare l'algoritmo di cycle canceling.

Si noti, prima di tutto, che affinché (5) ammetta soluzione, è necessario che $\sum_{v \in V} b_v = 0$. Infatti, se x è un flusso ammissibile, allora x soddisfa

$$\sum_{v \in V} \left(\sum_{w\,:\,(w,v) \in A} x_{wv} - \sum_{w\,:\,(v,w) \in A} x_{vw} \right) = \sum_{v \in V} b_v.$$

Si può verificare che il right-hand-side della precedente equazione è 0, e dunque $\sum_{v \in V} b_v = 0$.

Vediamo ora che il problema di trovare un flusso ammissibile è equivalente a trovare un flusso ammissibile di costo minimo in un grafo orientato "ausiliario".

Definiamo il grafo orientato $\tilde{D} = (\tilde{V}, \tilde{A})$ come segue:

- $\tilde{V} = V \cup \{r\}$, ove r è un nuovo nodo "dummy",
- $\tilde{A} = A \cup \{(r, v) : v \in V \text{ tale che } b_v > 0\} \cup \{(v, r) : v \in V \text{ tale che } b_v < 0\}$.

Le richieste $\tilde{b}_v$ sui nodi di $v \in \tilde{V}$ sono definite da

$$\tilde{b}_v = \begin{cases} b_v & \text{se } v \neq r, \\ 0 & \text{se } v = r. \end{cases}$$

Le capacità $\tilde{u}_e$ sugli archi $e \in \tilde{A}$ sono definite da

$$\tilde{u}_e = \begin{cases} u_e & \text{se } e \in A, \\ +\infty & \text{se } e \in \tilde{A} \setminus A. \end{cases}$$

I costi $\tilde{c}_e$ sugli archi $e \in \tilde{A}$ sono definiti da

$$\tilde{c}_e = \begin{cases} 0 & \text{se } e \in A, \\ 1 & \text{se } e \in \tilde{A} \setminus A. \end{cases}$$

Si noti che ogni flusso ammissibile ottimo x^* in $\tilde{D}$ (rispetto alle capacità $\tilde{u}$, richieste sui nodi $\tilde{b}$, e costi $\tilde{c}$) ha costo non-negativo, ed ha costo 0 se e solo se $x^*_e = 0$ per ogni $e \in \tilde{A} \setminus A$. In tal caso, il flusso determinato da x^* sugli archi del grafo originale D è un flusso ammissibile.

Viceversa, dato un flusso ammissibile $\bar{x}$ per D, il flusso x^* definito da

$$x^*_e = \begin{cases} \bar{x}_e & \text{se } e \in A, \\ 0 & \text{se } e \in \tilde{A} \setminus A. \end{cases}$$

è ammissibile in $\tilde{D}$ ed ha costo 0 rispetto a $\tilde{c}$, ed è pertanto ottimo.

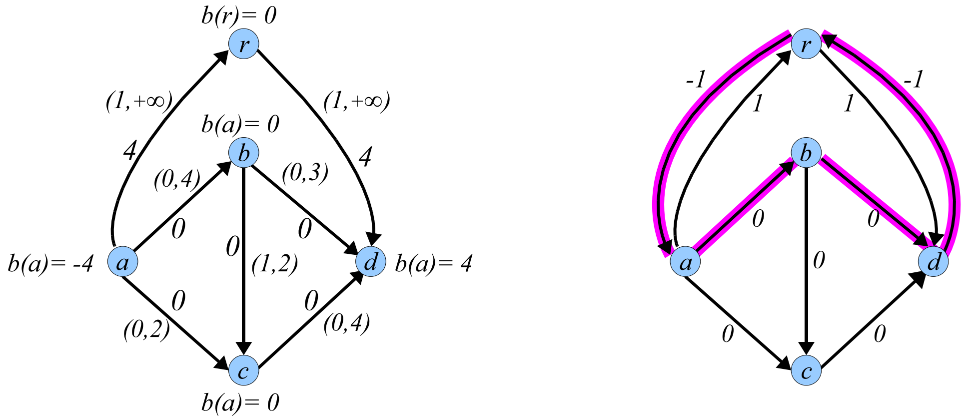
Dunque esiste un flusso ammissibile per D se e solo se il flusso di costo minimo in $\tilde{D}$ (rispetto ai costi $\tilde{c}$) ha costo 0.

Per trovare un flusso ammissibile in D dobbiamo dunque risolvere un problema di flusso a costo minimo in $\tilde{D}$ rispetto ai costi $\tilde{c}$, con capacità $\tilde{u}$ e richieste $\tilde{b}$. Basterà dunque applicare l'algoritmo di cycle canceling a tale problema, partendo dal seguente flusso ammissibile in $\tilde{D}$ rispetto alle capacità $\tilde{u}$ e richieste sui nodi $\tilde{b}$:

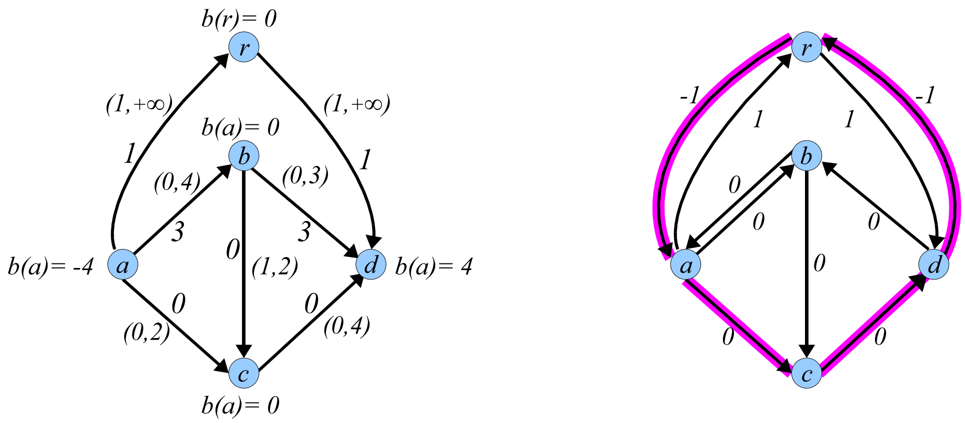
$$\tilde{x}_e = \begin{cases} 0 & \text{se } e \in A, \\ b_v & \text{se } e = (r, v), \\ -b_v & \text{se } e = (v, r). \end{cases}$$

Un esempio è mostrato in Figura 3.

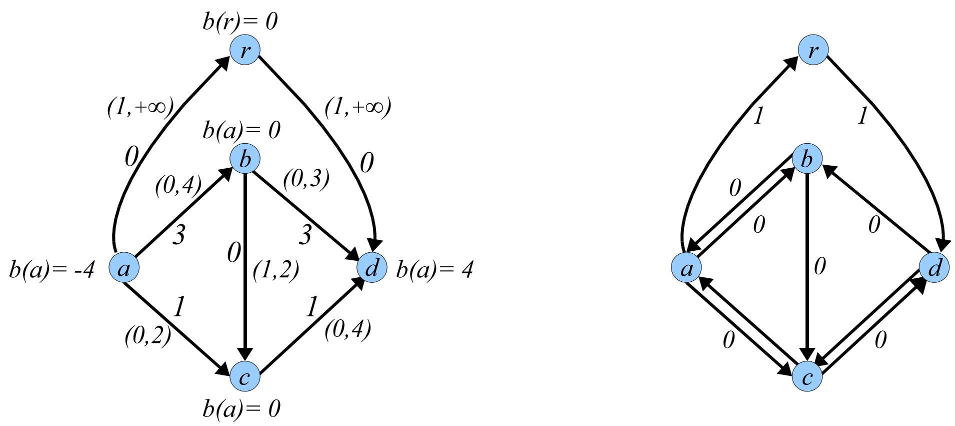
Si noti che, poiché le capacità sugli archi e le richieste sui nodi del grafo ausiliario sono intere, allora il flusso ottimo finale è intero.



(a) Aumenta x lungo il ciclo migliorante r-a-b-d-r ($\epsilon = 3$).



(a) Aumenta x lungo il ciclo migliorante r-a-c-d-r ($\epsilon = 1$).



(a) Trovato flusso ammissibile per D (il flusso di partenza in Figura 2(a)).

Figura 3: La colonna di sinistra mostra il grafo ausiliare $\tilde{D}$ e il flusso corrente. La colonna di destra mostra il grafo residuo del grafo ausiliario e i costi corrispondenti sugli archi.