

Figura 7.11: Capacità anche sui vertici.

e si trova il flusso massimo da  $s^+$  a  $t^-$ . Poiché tutto il flusso entrante in un vertice  $v_i^+$  deve fluire lungo l'arco  $(v_i^+, v_i^-)$  la capacità dell'arco fittizio impone il rispetto della condizione (7.4). Si veda l'Esercizio 12.17.

## 7.7 Problema del flusso di costo minimo

UN ALTRO PROBLEMA DI FLUSSO di grande rilevanza è il problema del flusso di costo minimo. Anche in questo caso considereremo un grafo orientato  $G = (V, A)$  nel quale però ogni arco  $(v_i, v_j)$ , oltre ad avere una capacità intera  $q_{ij}$ , ha anche un costo intero positivo, definito da

$$c_{ij} = \text{costo per unità di flusso trasmesso lungo l'arco } (v_i, v_j).$$

Considereremo dunque il seguente *Problema del Flusso di Costo Minimo* (*Minimum Cost Flow Problem*, MCFP):

*Dato  $G = (V, A)$ , con capacità e costi associati agli archi, e dati due vertici  $s, t \in V$ , inviare un flusso di valore prefissato  $u$  da  $s$  a  $t$  in maniera tale che il costo complessivo sia minimo.*

Se il valore di  $u$  è maggiore del flusso massimo da  $s$  a  $t$  (Sezione 7.6) il problema non ha evidentemente soluzione. Altrimenti si deve trovare, fra le varie possibili configurazioni di flusso di valore  $u$ , quella che comporta il costo più basso. Esistono numerosi algoritmi per risolvere questo problema. Quello che esporremo, dovuto a Klein [36] e a Busacker e Gowen [8], è particolarmente elegante e consente di utilizzare tecniche viste in precedenza.

Il metodo opera in due fasi, ossia applica consecutivamente due distinti algoritmi:

*Fase 1:* si determina un flusso ammissibile di valore  $u$ ;

*Fase 2:* si modifica iterativamente il flusso, mantenendone inalterato il valore e diminuendone il costo.

### 7.7.1 Flusso ammissibile di valore prefissato

La Fase 1 è la più semplice. Si ricordi che l'algoritmo di Ford e Fulkerson incrementa iterativamente il flusso corrente mediante catene aumentanti, fino a trovare il flusso massimo. Basterà quindi arrestare il procedimento appena il flusso ha raggiunto il valore  $u$ . A tal fine sarà sufficiente sostituire il blocco **else** dell'algoritmo di pag. 134 con

```

begin
   $\delta^* := \delta(t)$ ;  $x := t$ ;
  let  $z$  (comment:  $z < u$ ) sia il valore del flusso attuale;
  if  $z + \delta^* > u$  then  $\delta^* := u - z$ 
    repeat
      ...
    until  $x = s$ ;
  if  $z$  (aggiornato)  $= u$  then stop else cancella tutte le etichette
end

```

La Fase 2 richiede l'introduzione di un nuovo concetto.

### 7.7.2 Grafo incrementale

Dato un flusso  $\xi_{ij}$  ( $i, j = 1, \dots, n$ ) di valore  $u$  nel grafo  $G = (V, A)$ , si definisce il *grafo incrementale*  $G^\mu = (V, A^\mu)$  come:

- $A^\mu = A_f^\mu \cup A_b^\mu$ ;
- $A_f^\mu = \{(v_i, v_j) : \xi_{ij} < q_{ij}\}$ ;
- $A_b^\mu = \{(v_j, v_i) : \xi_{ij} > 0\}$ ;
- $\forall (v_i, v_j) \in A_f^\mu$ , poni la capacità a  $q_{ij}^\mu = q_{ij} - \xi_{ij}$  e il costo a  $c_{ij}^\mu = c_{ij}$ ;
- $\forall (v_j, v_i) \in A_b^\mu$  poni la capacità a  $q_{ji}^\mu = \xi_{ij}$  e il costo a  $c_{ji}^\mu = -c_{ij}$ .

Nelle figure di questa sezione i costi degli archi verranno mostrati all'interno di quadrati. La Figura 7.12(a) mostra un arco di  $G$ , la Figura 7.12(b) i due corrispondenti archi di  $G^\mu$ .

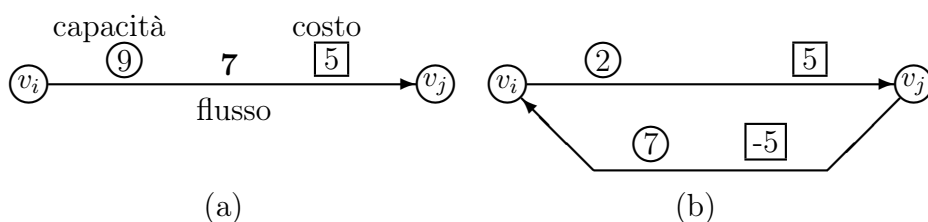


Figura 7.12: Corrispondenza tra arco del grafo e archi del grafo incrementale.

Un grafo incrementale rappresenta tutte le possibilità di modificare il flusso di un arco e le corrispondenti implicazioni sul costo. Si osservi infatti che, in Figura 7.12, l'arco  $(v_i, v_j)$  di  $G$  ha attualmente un costo non nullo e inferiore di 2 unità alla capacità. Tale flusso può quindi essere: (i) aumentato di una o due unità, pagando un costo 5 per ogni unità (arco  $(v_i, v_j)$  di  $G^\mu$ ), oppure: (ii) diminuito di non più di 7 unità, guadagnando un costo 5 per ogni unità (arco  $(v_j, v_i)$  di  $G^\mu$ ). Se l'arco  $(v_i, v_j)$  di  $G$  fosse saturo, nel grafo  $G^\mu$  ci sarebbe solo l'arco  $(v_j, v_i)$ , con capacità 9. Se viceversa l'arco  $(v_i, v_j)$  di  $G$  fosse scarico, nel grafo  $G^\mu$  ci sarebbe solo l'arco  $(v_i, v_j)$ , con capacità 9.

Si osservi anche che l'algoritmo di Ford e Fulkerson descritto nella Sezione 7.6 può essere interpretato come un metodo per trovare un cammino in un grafo incrementale (senza costi): la catena aumentante può infatti essere vista come un cammino in cui gli archi avanti corrispondono ad archi dell'insieme  $A_f^\mu$  e gli archi indietro ad archi dell'insieme  $A_b^\mu$ .

Si supponga ora che il grafo incrementale contenga un circuito. Si osservi preliminarmente che, data una qualunque configurazione di flusso, è sempre possibile (se le capacità lo consentono) trasmettere flusso, ad esempio di valore  $\delta$ , lungo un circuito senza alterare il valore attuale del flusso. Infatti, per ogni vertice del circuito, aumenteranno di  $\delta$  sia il flusso entrante che il flusso uscente per cui la legge di conservazione del flusso resterà soddisfatta. Abbiamo quindi implicitamente dimostrato un'importante proprietà:

**Proprietà 7.3** *Se il grafo incrementale  $G^\mu$  contiene un circuito tale che la somma dei costi dei suoi archi è negativa, si può trasmettere il massimo flusso possibile  $\delta$  lungo il circuito ottenendo una nuova configurazione di flusso che*

- (i) *trasmette la stessa quantità di flusso dalla sorgente al terminale;*
- (ii) *richiede un costo inferiore, rispetto alla configurazione precedente, di  $\delta$  volte il costo del circuito.*

La Figura 7.13(a) mostra una configurazione di flusso che trasmette un flusso di valore  $u = 10$  dal vertice  $v_1$  al vertice  $v_7$ . Si suppone che il costo di ogni arco sia pari ad 1, per cui il costo complessivo è pari a 41. Ci si soffermi

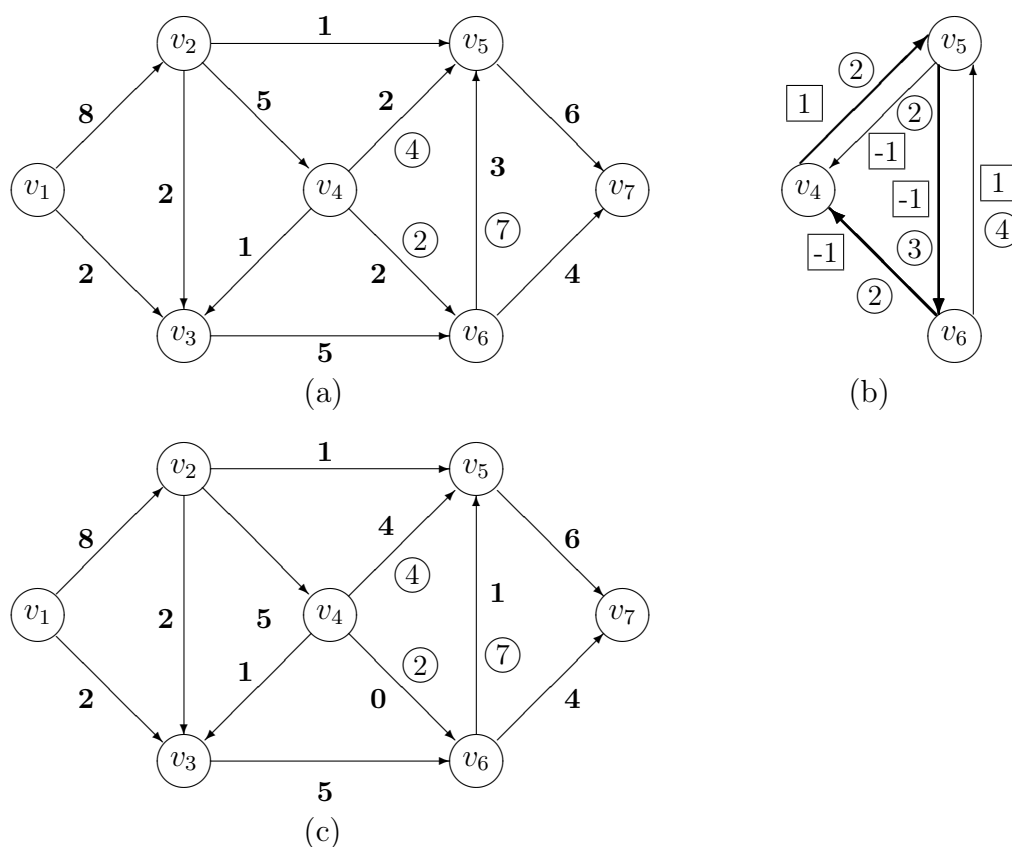


Figura 7.13: Circuito negativo nel grafo incrementale

sulla parte di grafo relativa ai vertici  $v_4, v_5$  e  $v_6$ : la Figura 7.13(b) mostra la corrispondente parte del grafo incrementale, nella quale si osserva il circuito  $\{v_4, v_5, v_6, v_4\}$ , di costo  $-1$ . La minima capacità di un arco del circuito è pari a 2, per cui possiamo trasmettere un flusso di tale valore lungo il circuito, ottenendo la nuova configurazione di flusso mostrata in Figura 7.13(c), di costo pari a 39.

Si sta dunque delineando l'algoritmo. Si trova un flusso di valore  $u$  (Fase 1) e se ne diminuisce iterativamente il costo trasmettendo flusso lungo circuiti di costo negativo nel grafo incrementale. E se il grafo incrementale non contiene alcun circuito di costo negativo? Vale in tal caso la seguente proprietà:

**Proprietà 7.4** *Condizione necessaria e sufficiente perché un flusso di valore  $u$  in  $G$  abbia il minimo costo possibile è che il corrispondente grafo incrementale  $G^\mu$  non contenga alcun circuito tale che la somma dei costi dei suoi archi sia negativa.*

**Dimostrazione** La necessità è ovvia: se il grafo incrementale contiene un circuito di costo negativo, il costo del flusso può essere diminuito. Omettiamo la (laboriosa) dimostrazione della sufficienza.  $\square$

### 7.7.3 Algoritmo

Premettiamo un'osservazione. Perché il grafo incrementale possa contenere un circuito di costo negativo, esso deve includere archi di costo negativo. Gli algoritmi per trovare cammini minimi in questo tipo di grafi, cui abbiamo accennato nell'ultima parte della Sezione 7.5 ('Grafici con pesi negativi', algoritmi di Dijkstra modificato e di Floyd–Warshall) identificano eventuali circuiti di costo negativo. Nell'algoritmo che segue, la seconda istruzione del ciclo **while** implica appunto l'uso di un tale algoritmo.

**procedure Minimum Cost Flow:**

**begin**

trova un flusso ammissibile  $[\xi_{ij}]$  di valore  $u$  con la procedura Flusso Massimo modificata;

$opt := false$ ;

**while**  $opt = false$  **do**

definisci il grafo incrementale  $G^\mu$  corrispondente al flusso attuale  $[\xi_{ij}]$ ;

trova un circuito  $\Phi$  (se esiste) di costo negativo in  $G^\mu$  con un appropriato algoritmo dei cammini minimi;

**if**  $G^\mu$  non contiene nessun circuito di costo negativo **then**  $opt := true$  **else**

**begin**

$\delta := \min_{(v_i, v_j) \in \Phi} \{q_{ij}^\mu\}$ ;

**for each**  $(v_i, v_j) \in \Phi$  with  $c_{ij}^\mu < 0$  **do**  $\xi_{ji} := \xi_{ji} - \delta$ ;

**for each**  $(v_i, v_j) \in \Phi$  with  $c_{ij}^\mu > 0$  **do**  $\xi_{ij} := \xi_{ij} + \delta$

**end**

**endwhile.**

**end.**

**Esempio 7.4** Il primo grafo di Figura 7.14 mostra un flusso da  $v_1$  a  $v_4$  di valore 4 e costo 18. A fianco è mostrato il corrispondente grafo incrementale, nel quale si ha un circuito  $\{v_2, v_3, v_4, v_2\}$  di costo  $-1$ . Trasmettendo 2 unità di flusso lungo il circuito, otteniamo la configurazione di flusso di costo 16, mostrata nel grafo successivo. Il nuovo grafo incrementale contiene il circuito  $\{v_1, v_3, v_4, v_2, v_1\}$  di costo  $-2$ . Trasmettendo 1 unità di flusso lungo il circuito, si ottiene la successiva configurazione di flusso di costo 14. Il grafo incrementale corrispondente non contiene alcun circuito di costo negativo. Il flusso ottenuto è quindi ottimo.  $\square$

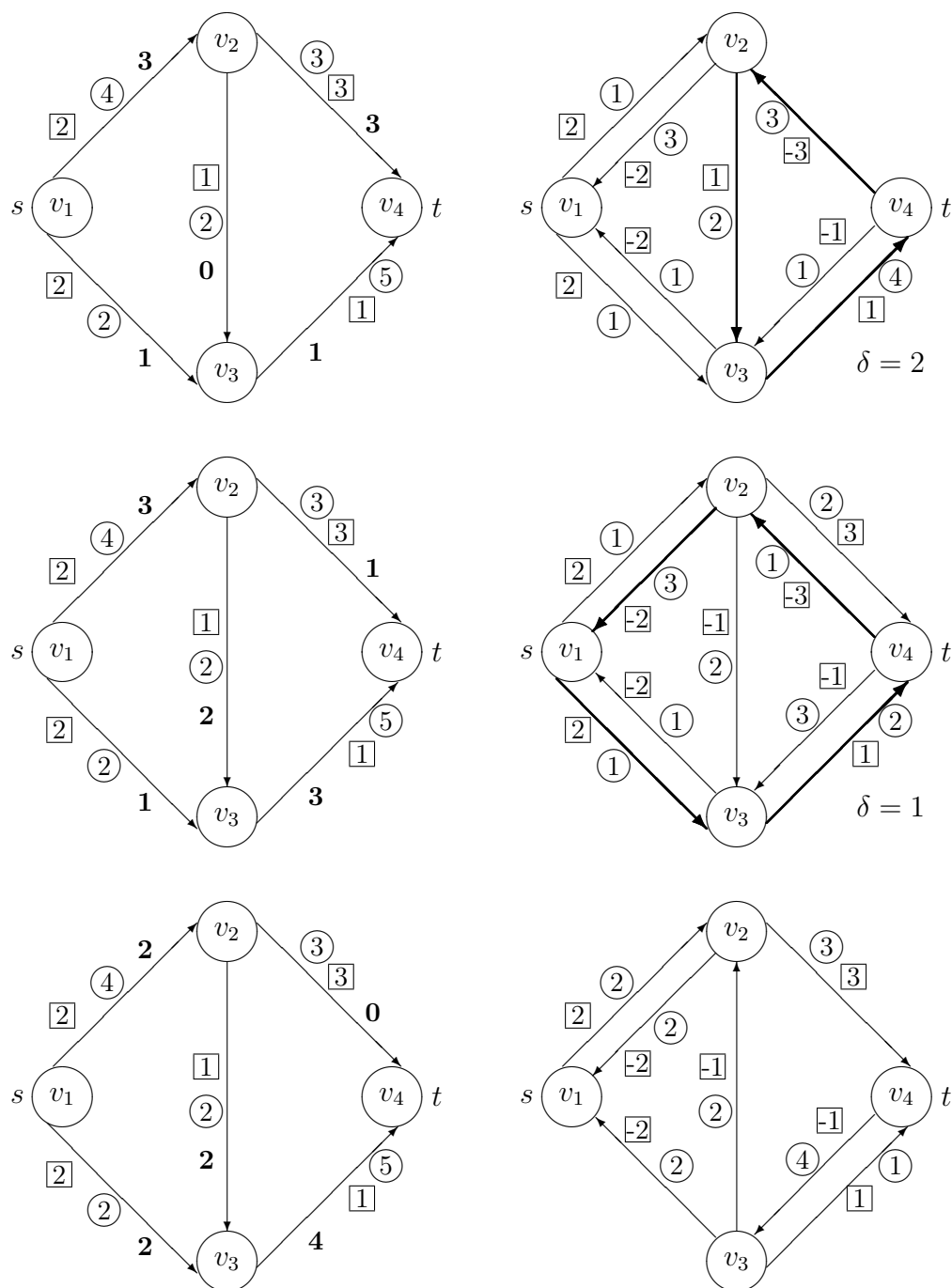


Figura 7.14: Esempio di applicazione dell'algoritmo di Klein, Busacker e Gowen.

Ad ogni iterazione il costo diminuisce almeno di una unità, per cui l'algoritmo converge. La Fase 1 richiede l'esecuzione di un algoritmo per il flusso

massimo, ed ogni iterazione della Fase 2 richiede l'esecuzione di un algoritmo per l'individuazione di un circuito di costo negativo. Il numero di iterazioni dipende dall'implementazione e può essere proporzionale a  $Q = \max_{ij}\{q_{ij}\}$  e  $C = \max_{ij}\{c_{ij}\}$ . Il tempo complessivo richiesto dall'algoritmo risulta elevato, di ordine  $O(n^3 Q C)$ . D'altra parte il problema del flusso di costo minimo è stato oggetto di molti studi, e numerosi altri algoritmi sono stati proposti per la sua soluzione. Il metodo che abbiamo visto è il più semplice ed elegante, ma altri metodi (anche molto più complicati) risolvono il problema richiedendo, nel caso peggiore, tempi notevolmente inferiori. Qualche informazione relativa a tali metodi verrà data nella Sezione 8.5.1.

## 7.8 Modelli matematici

I PROBLEMI DI TEORIA DEI GRAFI sono formalizzabili anche mediante modelli matematici. Vediamo come tali modelli possono essere definiti e quali relazioni si possono stabilire con la programmazione lineare.

Strumento importante per i nostri obiettivi è una rappresentazione dei grafi basata su una particolare struttura dati. Consideriamo un grafo orientato ed indichiamo con  $a_i$  ( $i = 1, \dots, m$ ) gli archi che lo compongono. Si definisce *matrice di incidenza* del grafo una matrice  $A = [a_{ij}]$  di  $n$  righe (una per vertice) ed  $m$  colonne (una per arco) in cui

$$a_{ik} = \begin{cases} +1 & \text{se l'arco } a_k \text{ esce dal vertice } v_i; \\ -1 & \text{se l'arco } a_k \text{ entra nel vertice } v_i; \\ 0 & \text{altrimenti.} \end{cases}$$

Nonostante l'evidente spreco di memoria rispetto alle strutture dati viste nella Sezione 7.4, questa matrice è molto utile per la formulazione di modelli.

### 7.8.1 Cammini minimi e programmazione lineare

Consideriamo il problema di trovare il cammino minimo da  $s$  a  $t$ , e riformuliamolo come: trasmettere, a costo minimo, un "flusso" di materiale da  $s$  a  $t$ . Sia  $\xi_k$  la quantità di materiale che fluisce lungo l'arco  $a_k$ , e interpretiamo  $c_k$  (la lunghezza dell'arco  $a_k$ ) come il costo per ogni unità di  $\xi_k$  che fluisce lungo  $a_k$ . Il problema del cammino minimo da  $s$  a  $t$  è quindi equivalente a quello di trovare un flusso unitario a costo minimo da  $s$  a  $t$ .

Osservando che una riga  $i$  della matrice di incidenza contiene un "+1" per ogni arco uscente da  $v_i$  e un "-1" per ogni arco entrante in  $v_i$ , la legge