Problemi di ottimizzazione su reti di flusso e algoritmi per il problema dei cammini minimi

Luigi De Giovanni

Dipartimento di Matematica, Università di Padova

Problemi di flusso di costo minimo: un esempio

Una società di produzione di energia elettrica dispone di diverse centrali di produzione e distribuzione, collegate tra loro con cavi reofori. Ogni centrale i può:

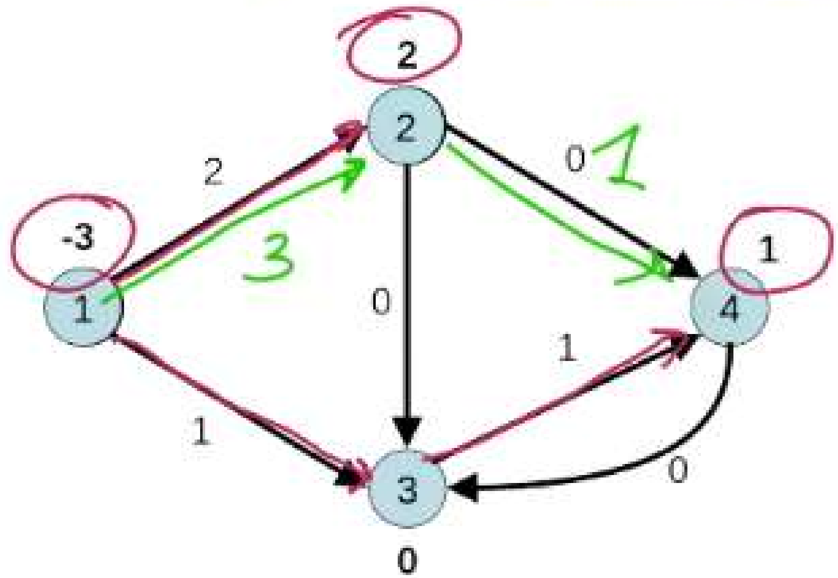
- produrre p_i kW di energia elettrica ($p_i = 0$ se la centrale non produce energia);
- distribuire energia elettrica su una sottorete di utenti la cui domanda complessiva è di d_i kW ($d_i = 0$ se la centrale non serve utenti);
- smistare l'energia da e verso altre centrali.

I cavi che collegano una centrale i ad una centrale j hanno una capacità massima di u_{ij} kW e costano c_{ij} euro per ogni kW trasportato. La società vuole determinare il piano di distribuzione dell'energia elettrica di costo minimo, sotto l'ipotesi che l'energia complessivamente prodotta sia pari alla domanda totale delle sottoreti di utenti.

Un modello su grafi: reti di flusso

- $G = (N, A)$
- $v \in N$: $b_v = d_v - p_v$ (richiesta del nodo v)
- $(i, j) \in A$: c_{ij}, u_{ij}
- nodi domanda (richiesta $b_i > 0$);
- nodi offerta (richiesta $b_i < 0$);
- nodi di transito (richiesta $b_i = 0$).

Esempio di possibili soluzioni ($\equiv$ flussi ammissibili)



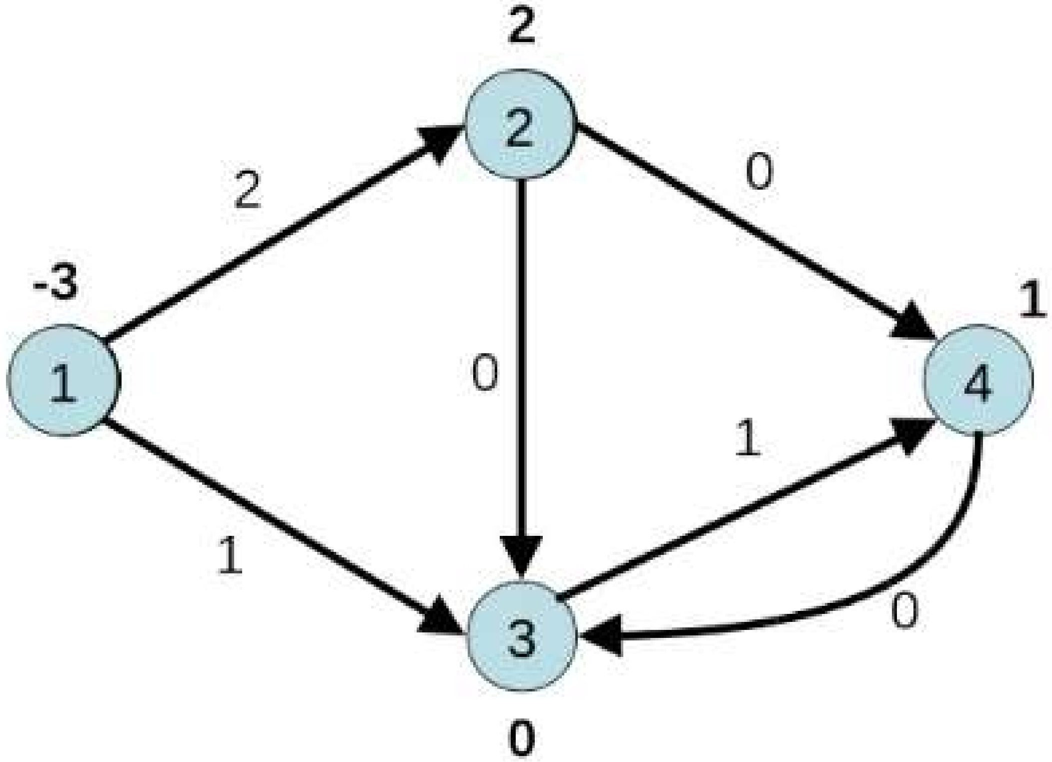
Modello PL: problema del flusso di costo minimo (min cost flow - MCF)

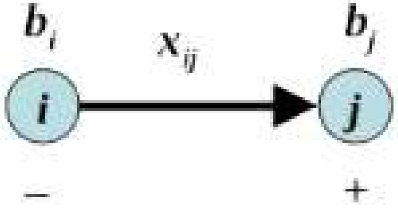
- variabili: quantità x_{ij} da far fluire sull'arco $(i,j) \in A$

$$
\begin{array}{llcll}
\min & \displaystyle\sum_{(i,j)\in A} c_{ij}x_{ij} & & & \\
s.t. & \displaystyle\sum_{(i,v)\in A} x_{iv} - \sum_{(v,j)\in A} x_{vj} & = & b_v & \forall\ v \in N \\
 & x_{ij} & \leq & u_{ij} & \forall\ (i,j) \in A \\
 & x_{ij} & \in & \mathbb{R}_+ &
\end{array}
$$

Nota: posso risolvere con un algoritmo per PL (ad es. simplesso)

Esempio di sistema dei vincoli di flusso





$$
\begin{array}{lr}
-x_{12}-x_{13} & =-3 \text{ bilanciamento del nodo 1}\\
+x_{12}-x_{23}-x_{24} & = 2 \text{ bilanciamento del nodo 2}\\
+x_{13}+x_{23}-x_{34}+x_{43} & = 0 \text{ bilanciamento del nodo 3}\\
+x_{24}+x_{34}-x_{43} & = 1 \text{ bilanciamento del nodo 4}
\end{array}
$$

Integer MCF

- le quantità x_{ij} sono discrete (container, automobili)

$$
\begin{array}{lrcll}
\min & \displaystyle\sum_{(i,j)\in A} c_{ij}x_{ij} & & & \\
s.t. & \displaystyle\sum_{(i,v)\in A} x_{iv} - \sum_{(v,j)\in A} x_{vj} & = & b_v & \forall \; v \in N \\
 & x_{ij} & \leq & u_{ij} & \forall \; (i,j) \in A \\
 & x_{ij} & \in & \mathbb{Z}_+ &
\end{array}
$$

Come lo risolvo?

Matrice dei vincoli

$$
\begin{array}{rrrrrrcr}
-x_{12} & -x_{13} & & & & & = & -3 \\
+x_{12} & & -x_{23} & -x_{24} & & & = & 2 \\
& +x_{13} & +x_{23} & & -x_{34} & +x_{43} & = & 0 \\
& & & +x_{24} & +x_{34} & -x_{43} & = & 1
\end{array}
$$

x_{12}	x_{13}	x_{23}	x_{24}	x_{34}	x_{43}
−1	−1	0	0	0	0
+1	0	−1	−1	0	0
0	+1	+1	0	−1	+1
0	0	0	+1	+1	−1

- È la *matrice di incidenza nodo-arco* del grafo G (indicata con E)
- la somma delle righe è il vettore nullo: $det(E) = 0$
- si dimostra $\rho(E) = |N| - 1$ [sfrutta: un +1 e un -1 per colonna]
- ammissibile $\iff \sum_{v \in N} b_v = 0$ (rete *bilanciata*)

Matrici Totalmente Unimodulari (TU)

Proprietà

Sia E la matrice corrispondente ai vincoli di bilanciamento di flusso e D una qualsiasi sottomatrice quadrata (di qualsiasi dimensione). Allora $\det(D) \in \{-1, 0, +1\}$: E è **Totalmente Unimodulare**.

Sia B sottomatrice quadrata di E (meno una riga...). Si ha:

- $(B^{-1})_{i,j} = (-1)^{i+j} \dfrac{\det(B^{ji})}{\det(B)}$
- B^{-1} è intera (componenti in $\{0, +1, -1\}$)

Teorema

Siano $A \in \mathbb{R}^{m \times n}$ una matrice TU e $b \in \mathbb{R}^m$ un vettore intero. Allora tutte le soluzioni di base del sistema $Ax = b$ $(x \geq 0)$ sono intere.

Matrici TU e problemi di flusso

$$
\begin{array}{llcll}
\min & \displaystyle\sum_{(i,j)\in A} c_{ij}x_{ij} & & & \\
s.t. & \displaystyle\sum_{(i,v)\in A} x_{iv} - \sum_{(v,j)\in A} x_{vj} & = & b_v & \forall v \in N \\
 & x_{ij} & \leq & u_{ij} & \forall (i,j) \in A \\
 & x_{ij} & \in & \mathbb{Z}_+ \equiv \mathbb{R}_+ &
\end{array}
$$

Se $b_v \in \{0,1\} \forall\, v \in N$, posso risolvere con il metodo del simplesso!

Più in generale, sia il PLI $\min\{c^T x : Ax = b, x \in \mathbb{Z}^n_+\}$.
Se A è TU e $b \in \mathbb{Z}^m_+$, allora il problema può essere risolto con l'algoritmo del simplesso (fornisce direttamente la soluzione ottima intera). **Altrimenti, il simplesso non basta!**

Problema del cammino minimo

Variabili decisionali: $x_{ij} = \begin{cases} 1, & \text{l'arco } (i,j) \text{ è sul cammino minimo;} \\ 0, & \text{altrimenti.} \end{cases}$

$$\min \quad \sum_{(i,j)\in A} c_{ij}x_{ij}$$

s.t.

$$\sum_{(i,v)\in A} x_{iv} - \sum_{(v,j)\in A} x_{vj} \quad = \quad \begin{cases} -1, & v = s; \\ +1, & v = d; \\ 0, & v \in N \setminus \{s,d\}. \end{cases}$$

$$x_{ij} \quad \in \quad \{0,1\} \quad \forall\, (i,j) \in A$$

Caso **particolare MCF**: un nodo domanda s e un nodo offerta d

Passaggio al duale

$\max \pi^T b$

$a_j^T \pi \quad c_j$

Obiettivo:

sfruttare teoria della dualità **in PL** per algoritmo più efficiente del simplesso

	x_{12}	x_{13}	x_{23}	x_{24}	x_{34}	x_{43}	b_i	
min	c_{12}	c_{13}	c_{23}	c_{24}	c_{34}	c_{43}		
	-1	-1	0	0	0	0	-1	$\pi_1 = \pi_s$
	$+1$	0	-1	-1	0	0	0	π_2
	0	$+1$	$+1$	0	-1	$+1$	0	π_3
	0	0	0	$+1$	$+1$	-1	$+1$	$\pi_4 = \pi_d$

$-\pi_1 + \pi_2$ $\qquad$ $i \; j$ $\qquad$ $-\pi_i + \pi_j \le c_{ij}$

$$\boxed{\max \quad \pi_d - \pi_s}$$

$$\text{s.t.} \quad \boxed{\pi_j - \pi_i \quad \le \quad c_{ij}} \quad \forall\,(i,j) \in A$$

$$\boxed{\pi_v} \quad \in \quad \boxed{\mathbb{R}} \quad \forall\, v \in N$$

Algoritmo Label Correcting generico per SPP

Notazione: variabili duali o etichette o *label*

π_s := 0; set $p(s) = \wedge$;

for each $v \in N - s$ { set π_v:=$+\infty$, set $p(v) = \wedge$; }

while ($\exists\,(i,j) \in A : \pi_j > \pi_i + c_{ij}$) do {

  set π_j := $\pi_i + c_{ij}$ ipotesi di soluzione duale

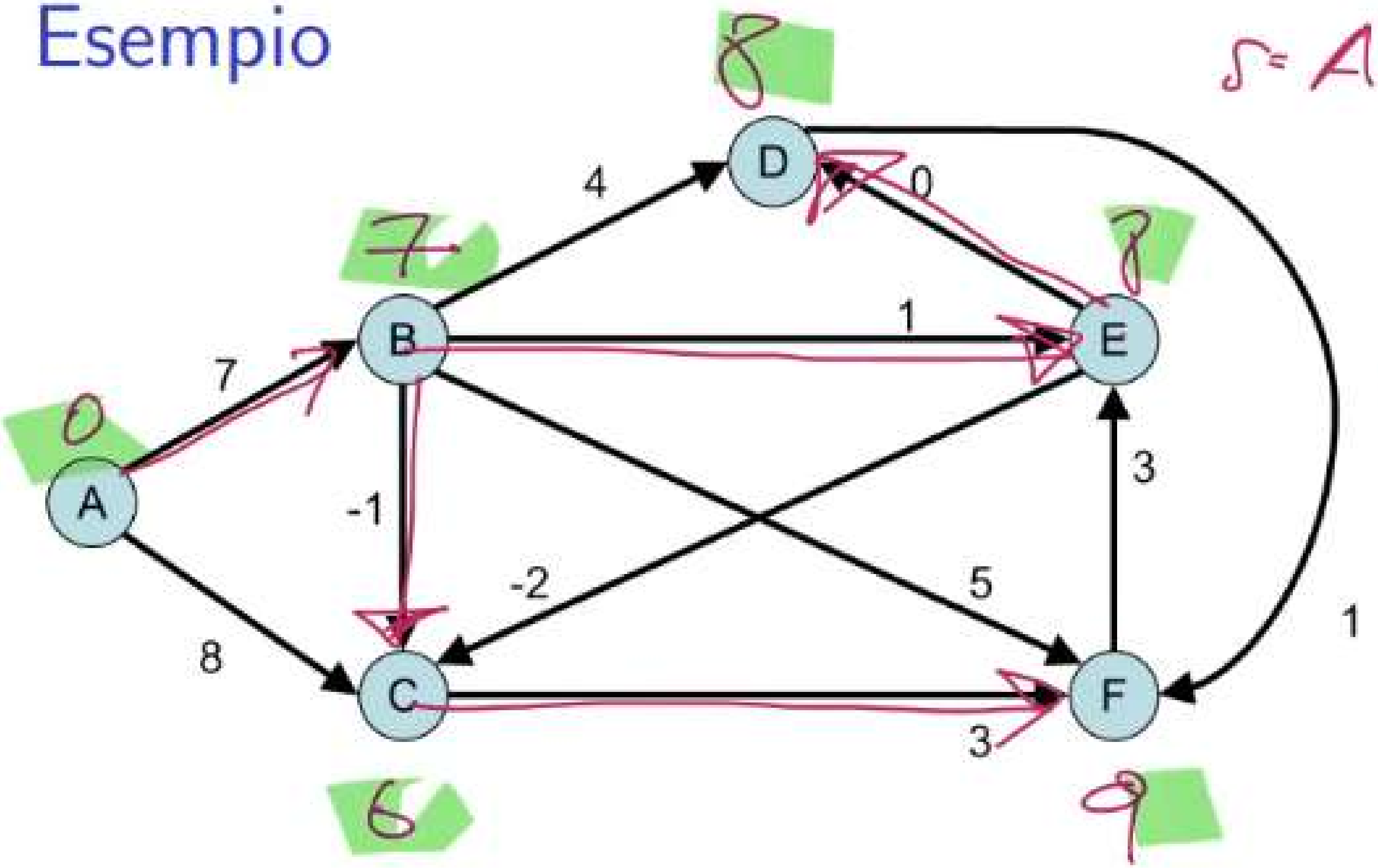
  set $p(j)$:= i;   ipotesi di cammino (sol.primale)

}

- $\pi_s = 0$: un vincolo primale ridondante
- convenzione: $+\infty \pm cost. = +\infty$

funziona?

Esempio



$s = A \quad d = F$

$\pi_B > \pi_A + 7$

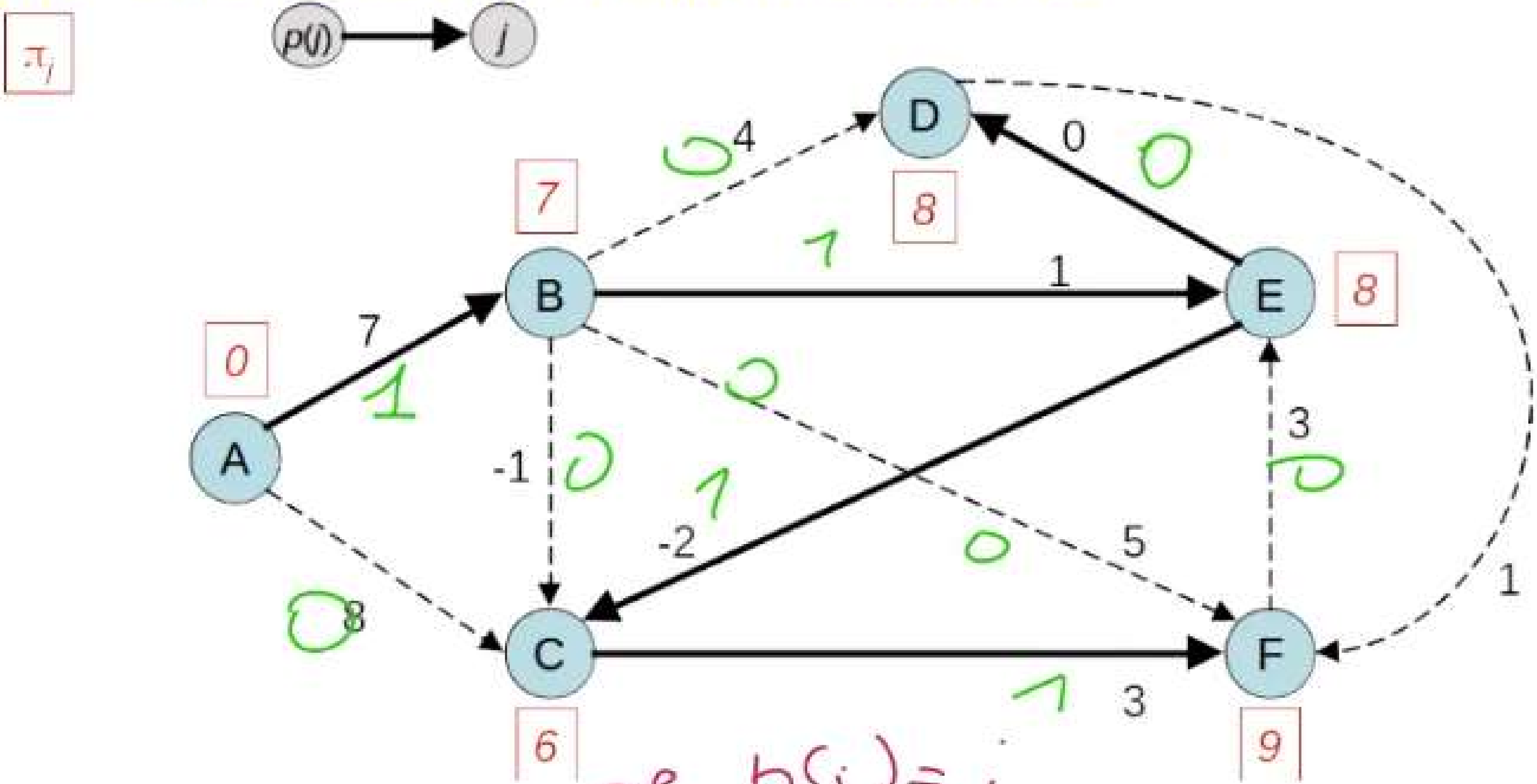
$+\infty > 0 + 7$

$\pi_B = 7$

$\pi_A = 0 \quad \pi_B = 7 \quad \ldots\ldots$

$\pi_j > \pi_i + c_{ij}$?

Esempio: etichette ammissibili e ottime



Soluzione primale: $x_{ij} = 1$ su cammino $[A, B, E, C, F]$, 0 altrimenti

$$\sum_{(i,j)\in A} c_{ij}\, x_{ij} \;=\; \pi_F - \pi_A \quad \Rightarrow \quad x \text{ ottima!}$$

Proprietà dell'algoritmo label correcting

Lemma

Alla fine di ogni iterazione, se $\pi_j < +\infty$ allora

- $\Rightarrow$ esiste cammino P da s a j
- $\Rightarrow$ $p(j)$ è il predecessore di j su P
- $\Rightarrow$ costo di P è π_j

Al termine dell'algoritmo, se esiste un cammino da s a j, allora $\pi_j < +\infty$.

Dimostrazione prima parte per induzione.

$$\begin{cases} \Pi(1) = true \\ \Pi(k) = true \Rightarrow \Pi(k+1) = true \end{cases} \quad \Rightarrow \quad \Pi(n) = true,\ \forall\, n \geq 1$$

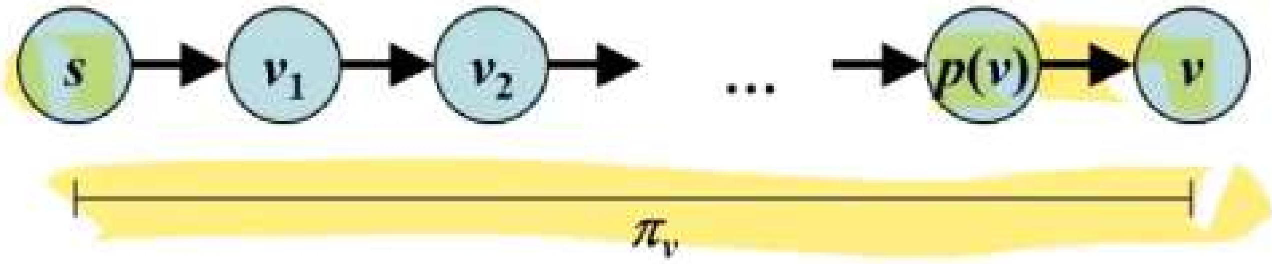
Prima parte: dimostrazione per $k = 1$

A fine iterazione 1:

- Per s:
  $\pi_s = 0$: $P = \emptyset$, ✓
- Altri w con $\pi_w < +\infty$:
  $\rightarrow \quad \pi_w = \pi_s + c_{sw}$ e $p(w) = s \quad \rightarrow \quad P = [s, w]$, ✓
- nessun altro nodo j con $\pi_j < +\infty$

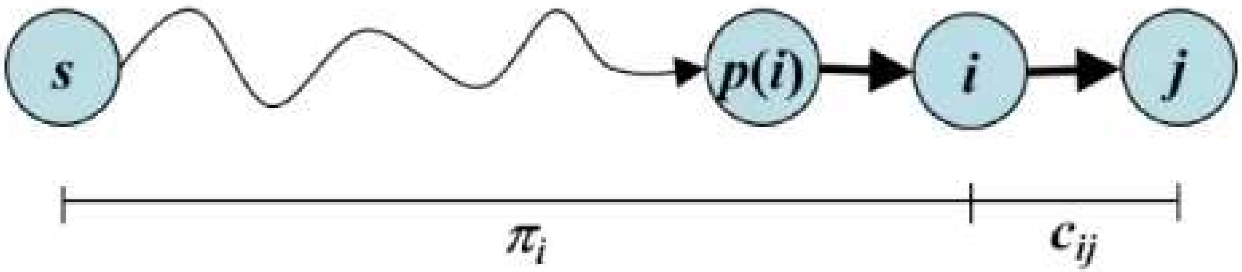
Prima parte: dimostrazione per $k > 1$

Ipotesi induttiva: all'iterazione k, se $\pi_v < +\infty$, allora:



All'iterazione $k+1$, sia (i,j) l'arco interessato. Allora:

- $\pi_j : \pi_j \geq \pi_i + c_{ij} \quad \rightarrow \quad \pi_i < +\infty$ all'iterazione k
- i ricade nell'ipotesi induttiva, cammino P:



- cammino $[P, j]$ con costo $\pi_i + c_{ij} = \pi_j$ e $p(j) = i$ ✓

Seconda parte

- Per ipotesi, esiste $P = [s, v_1, v_2, \ldots, v_{n-2}, v_{n-1}, v]$.

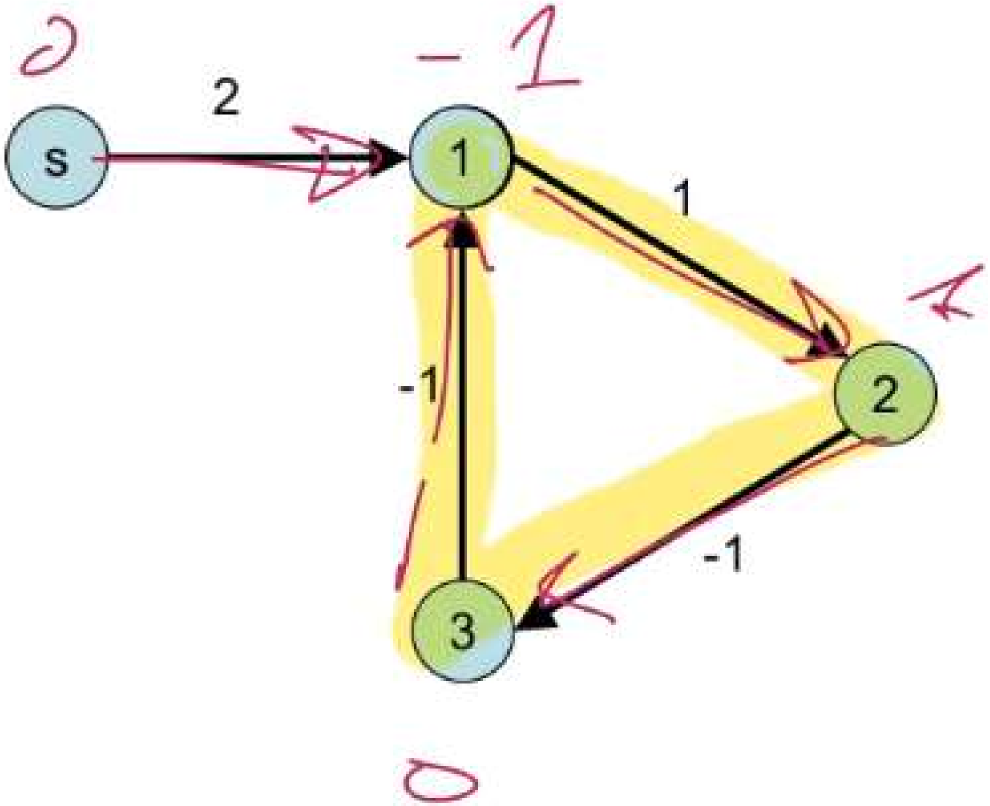
- Assumiamo per assurdo che $\pi_v = +\infty$.

- Al termine dell'algoritmo, π ammissibile:
$$+\infty = \pi_v \leq \pi_{v_{n-1}} + c_{v_{n-1}\,v} \leq \pi_{v_{n-2}} + c_{v_{n-2}\,v_{n-1}} + c_{v_{n-1}\,v} \leq \ldots$$
$$\ldots \leq \pi_{v_1} + c_{v_1 v_2} + \ldots + c_{v_{n-2}v_{n-1}} + c_{v_{n-1}v} \leq \ldots$$
$$\ldots \leq \pi_s + c_{sv_1} + c_{v_1 v_2} + \ldots + c_{v_{n-2}v_{n-1}} + c_{v_{n-1}v}$$

- Pertanto, essendo $\pi_s = 0$,
$c_{sv_1} + c_{v_1 v_2} + \ldots + c_{v_{n-2}v_{n-1}} + c_{v_{n-1}v} \geq +\infty$: contraddizione!

Convergenza

In presenza di un **ciclo di lunghezza negativa**, l'algoritmo **non converge**.



Init	$\pi_s = 0$; $\pi_1 = \pi_2 = \pi_3 = +\infty$;
It 1	arco $(s, 1)$: $\pi_1 = 2$; $p(1) = s$.
It 2	arco $(1, 2)$: $\pi_2 = 3$; $p(2) = 1$.
It 3	arco $(2, 3)$: $\pi_3 = 2$; $p(3) = 2$.
It 4	arco $(3, 1)$: $\pi_1 = 1$; $p(1) = 3$.
It 5	arco $(1, 2)$: $\pi_2 = 2$; $p(2) = 1$.
It 6	arco $(2, 3)$: $\pi_3 = 1$; $p(3) = 2$.
It 7	arco $(3, 1)$: $\pi_1 = 0$; $p(1) = 3$.
It 8	arco $(1, 2)$: $\pi_2 = 1$; $p(2) = 1$.
It 9	arco $(2, 3)$: $\pi_3 = 0$; $p(3) = 2$.
It 10	arco $(3, 1)$: $\pi_1 = -1$; $p(1) = 3$.

...

Convergenza senza cicli negativi

- Ad ogni iterazione, una etichetta diminuisce strettamente
- le etichette finite indicano il costo di un cammino *ammissibile*, non possono diminuire indefinitamente: $\pi_j \geq c(P^*) > -\infty$
- π_j rimane sempre a $+\infty$ se j non raggiungibile

Senza cicli negativi (e assumendo c_{ij} razionali):
convergenza in un numero finito di iterazioni

Osservazione

Senza cicli negativi, un cammino minimo contiene al più $|N| - 1$ archi

Complessità computazionale senza cicli negativi

- max iterazioni: $N-1$ etichette da $+\infty$ a costo minimo
- $+\infty$: possiamo usare un upperbound $\overline{M}$
- sia $\underline{M}$ un lower bound per il costo minimo
- ogni iterazione costa $O(|A|)$

Proprietà: complessità computazionale

Senza cicli negativi, l'algoritmo LCG converge in $O((|N|-1)\,(\overline{M}-\underline{M})\,|A|)$

Dimensionamento $\overline{M}$ e $\underline{M}$

- max costo P ammissibile $\overline{M} = (|N|-1)\max\left\{0, \max_{(i,j)\in A} c_{ij}\right\}$
- min costo P ammissibile $\underline{M} = (|N|-1)\min\left\{0, \min_{(i,j)\in A} c_{ij}\right\}$

Correttezza

Proprietà

Senza cicli negativi, l'algoritmo LCG trova il cammino minimo da s a d

- π è soluzione ammissibile duale di costo π_d
- Senza cicli negativi, se esiste cammino da s a d, allora $\pi_d < +\infty$.
- Per il Lemma, esiste $P = [s, ..., d]$ di costo π_d. Tale cammino si ottiene con la catena dei predecessori.
- Costruisco una soluzione primale ponendo $x_{ij} = 1$ sugli archi di P
- La soluzione è ammissibile primale (flusso bilanciato) e ha costo

$$\sum_{(i,j)\in A} c_{ij}x_{ij} = \sum_{(i,j)\in P} c_{ij} = \pi_d$$

$\Rightarrow$ P è ottimo! (dualità forte[1])

[1] applicabile in quanto il problema è stato ricondotto a un PL con variabili **continue**

Stessa origine, destinazione diversa

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

s.t.

$$\sum_{(i,v)\in A} x_{iv} - \sum_{(v,j)\in A} x_{vj} = \begin{cases} -1, & v = s; \\ +1, & v = d; \\ 0, & v \in N \setminus \{s, d\}. \end{cases}$$

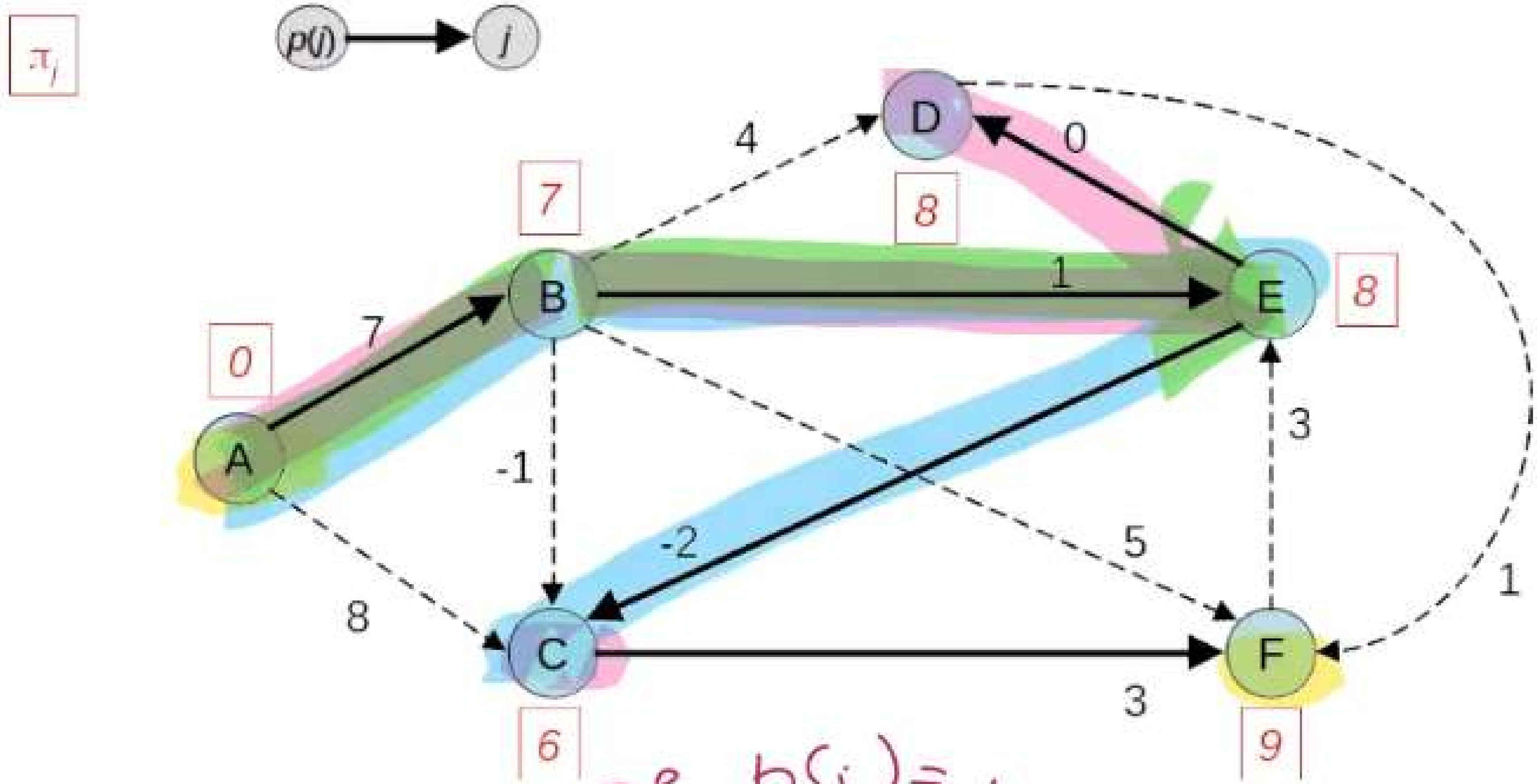
$$x_{ij} \in \mathbb{R}_+ \quad \forall (i,j) \in A$$

$$\max \quad \pi_d - \pi_s$$

$$\text{s.t.} \quad \pi_j - \pi_i \leq c_{ij} \quad \forall (i,j) \in A$$

$$\pi_v \in \mathbb{R} \quad \forall v \in N$$

Esempio: etichette ammissibili e ottime



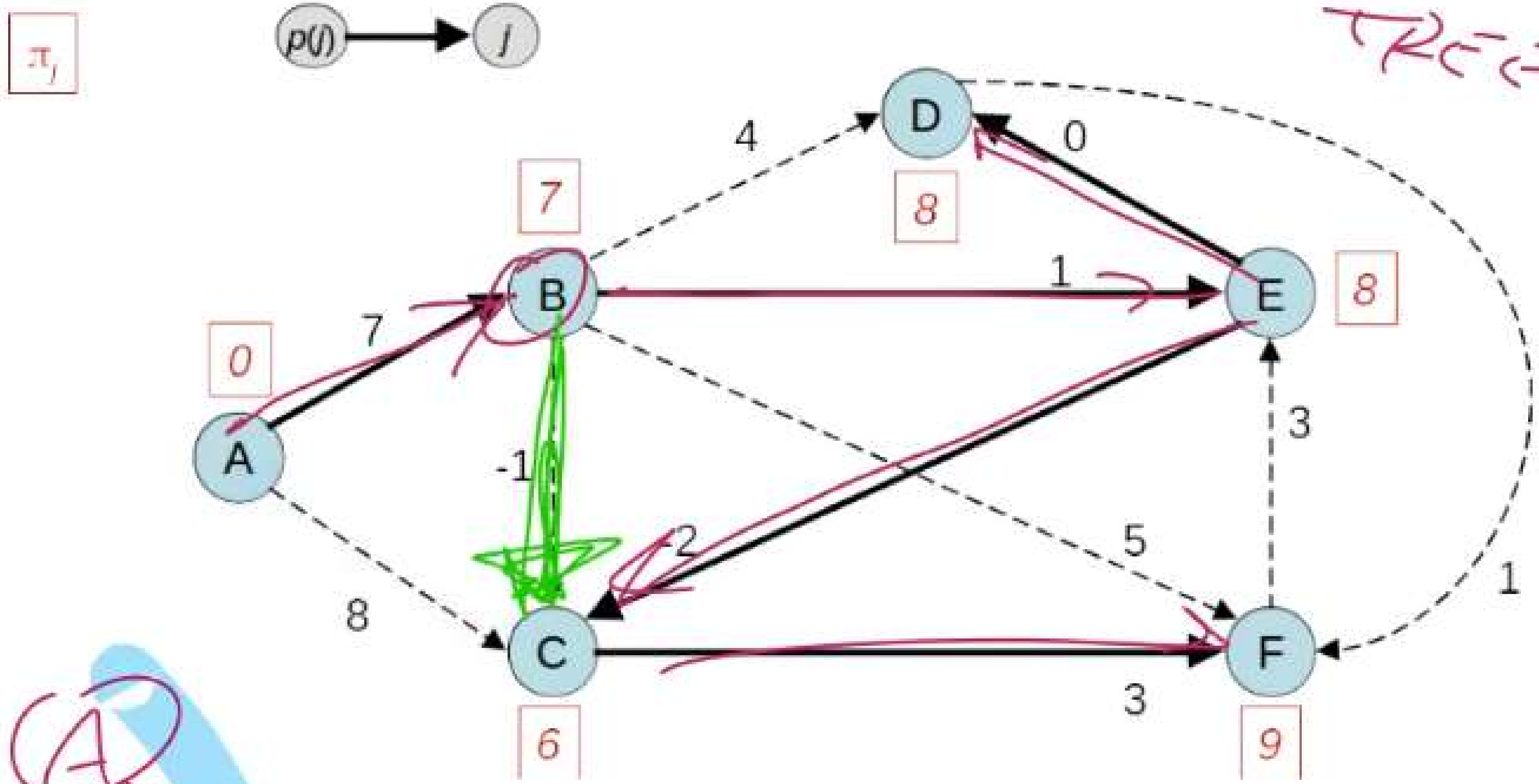
$- x \; p(j) =$

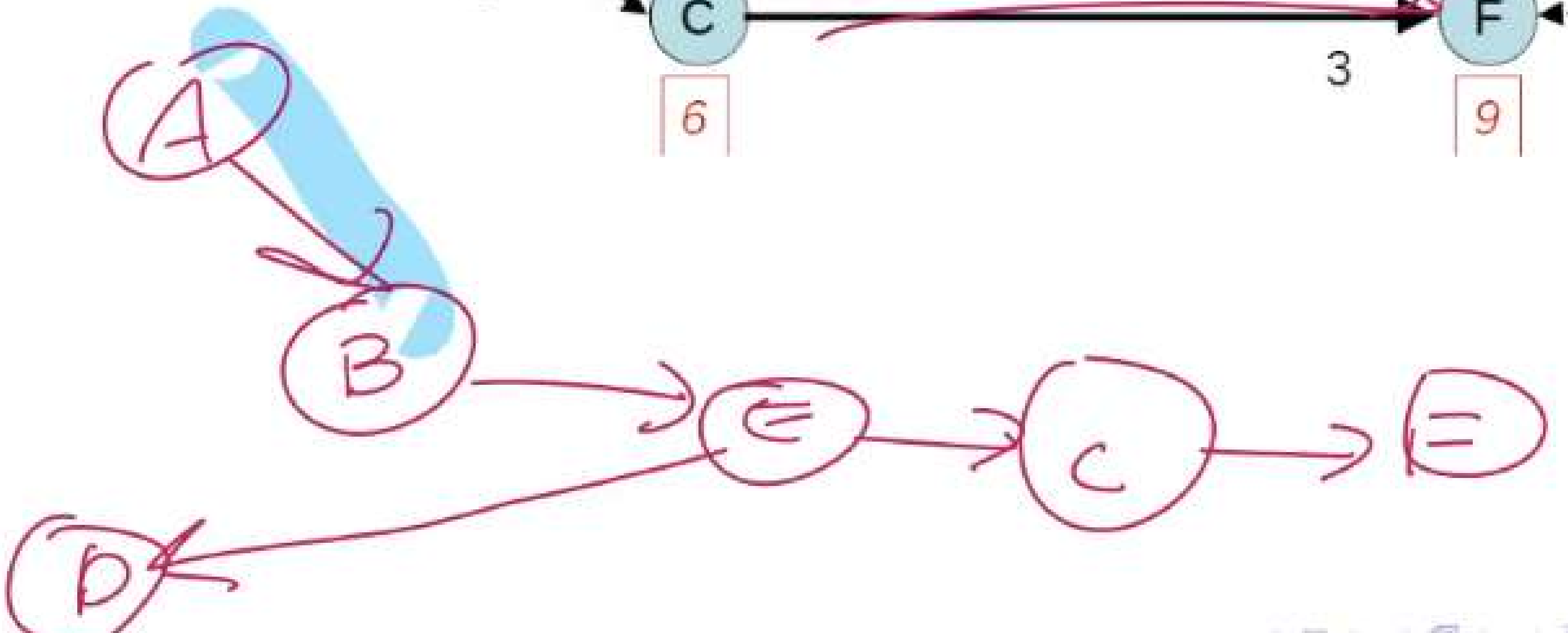
Soluzione primale: $x_{ij} = 1$ su cammino $[A, B, E, C, F]$, 0 altrimenti

$$\sum_{(i,j)\in A} c_{ij}\, x_{ij} = \pi_F - \pi_A \quad \Rightarrow \quad x \text{ ottima!}$$

Esempio

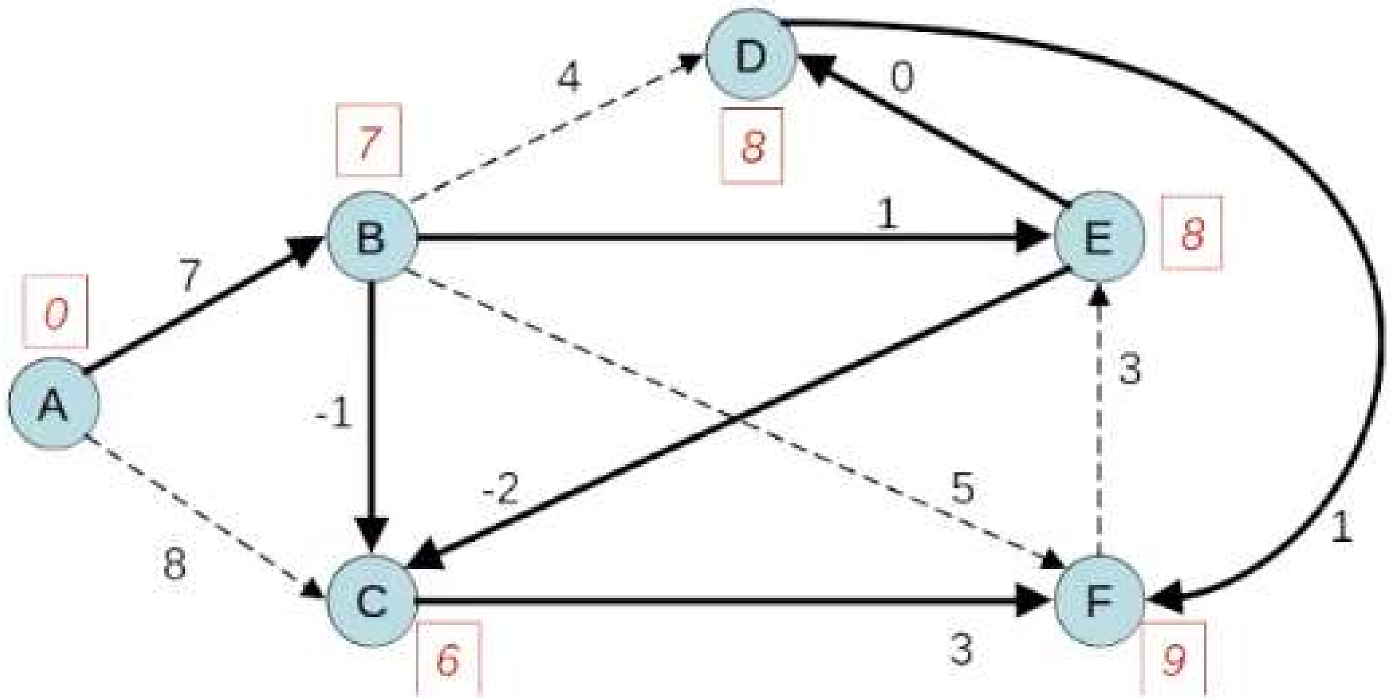
SHORTEST PATH TREE





Grafo dei cammini minimi (cammini minimi alternativi)

$$G_{s,\pi} = (N, A_{s,\pi}) \quad \text{dove} \quad A_{s,\pi} = \{(i,j) \in A : \pi_j = \pi_i + c_{ij}\}.$$



$$\text{Cammino } [s, \dots, j] \text{ su } G_{s,\pi} \iff \text{cammino minimo da } s \text{ a } j$$

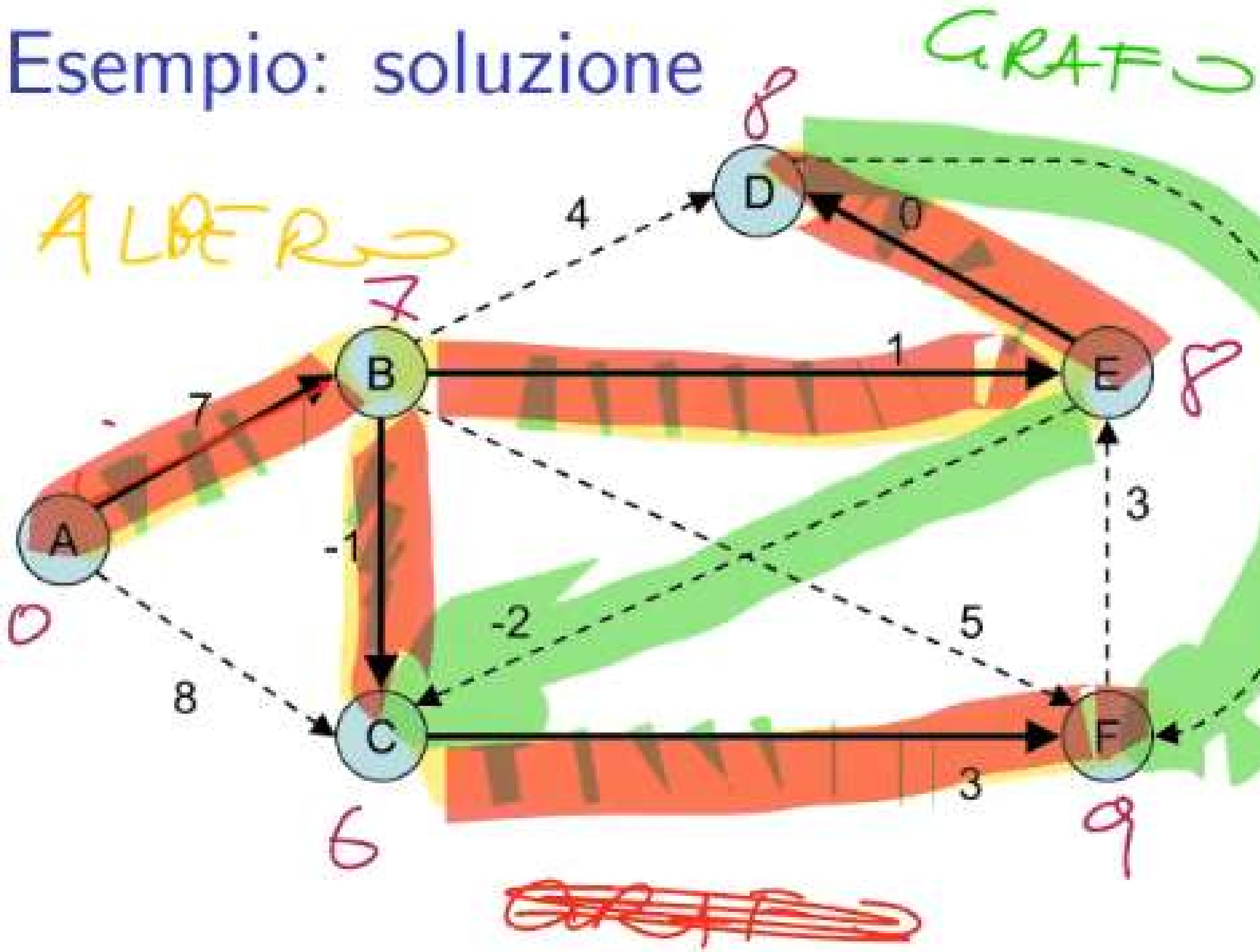
cammino individua soluzioni ammissibili primale e duale con

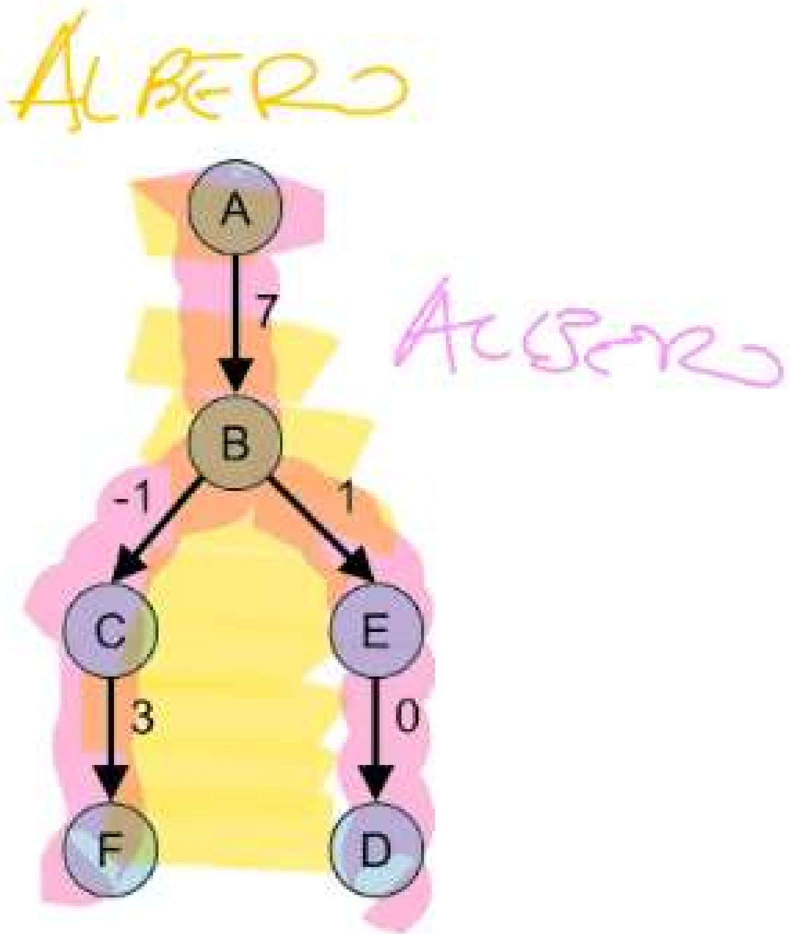
$$(\pi_j - \pi_i - c_{ij})\, x_{ij} = 0, \ \forall\, (i,j) \in A$$

Algoritmo di Bellman-Ford (label correcting)

```
π_s := 0; set p(s) = Λ;
for each v ∈ N − s { set π_v := +∞, set p(v) := Λ }
for h = 1 to |N| {
    set π' := π; set flag_aggiornato := false;
    for all ( (i,j) ∈ A : π_j > π'_i + c_ij ) do {
        set π_j := π'_i + c_ij
        set p(j) := i;
        set flag_aggiornato := true;
    }
    if (not flag_aggiornato) then { STOP: π è ottima }
}
STOP: ∃ ciclo di costo negativo.
```

Esempio: soluzione





iterazione	nodo A	nodo B	nodo C	nodo D	nodo E	nodo F	flag
$h = 0$	$0_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	*true*
$h = 1$	$0_{(\wedge)}$	$+7_{(A)}$	$+8_{(A)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	*true*
$h = 2$	$0_{(\wedge)}$	$+7_{(A)}$	$+6_{(B)}$	$+11_{(B)}$	$+8_{(B)}$	$\cancel{+12_{(B)}} + 11_{(C)}$	*true*
$h = 3$	$0_{(\wedge)}$	$+7_{(A)}$	$+6_{(B)}$	$+8_{(E)}$	$+8_{(B)}$	$+9_{(C)}$	*true*
$h = 4$	$0_{(\wedge)}$	$+7_{(A)}$	$+6_{(B)}$	$+8_{(E)}$	$+8_{(B)}$	$+9_{(C)}$	*false*

Proprietà: cammini minimi e **max-hop**

Lemma

Alla fine dell'iterazione con $h = \bar{h}$, π_j rappresenta il costo di un **cammino minimo da s a j con al più $\bar{h}$ archi**

Per induzione:

- $h = 1$: verificati **tutti** i modi di arrivare da s con 1 solo arco, ✓
- $h = k$

- π' ottime per $k - 1$ archi (ipotesi induttiva)
- verificati **tutti** i modi di raggiungere nodi con *un arco in più*, ✓

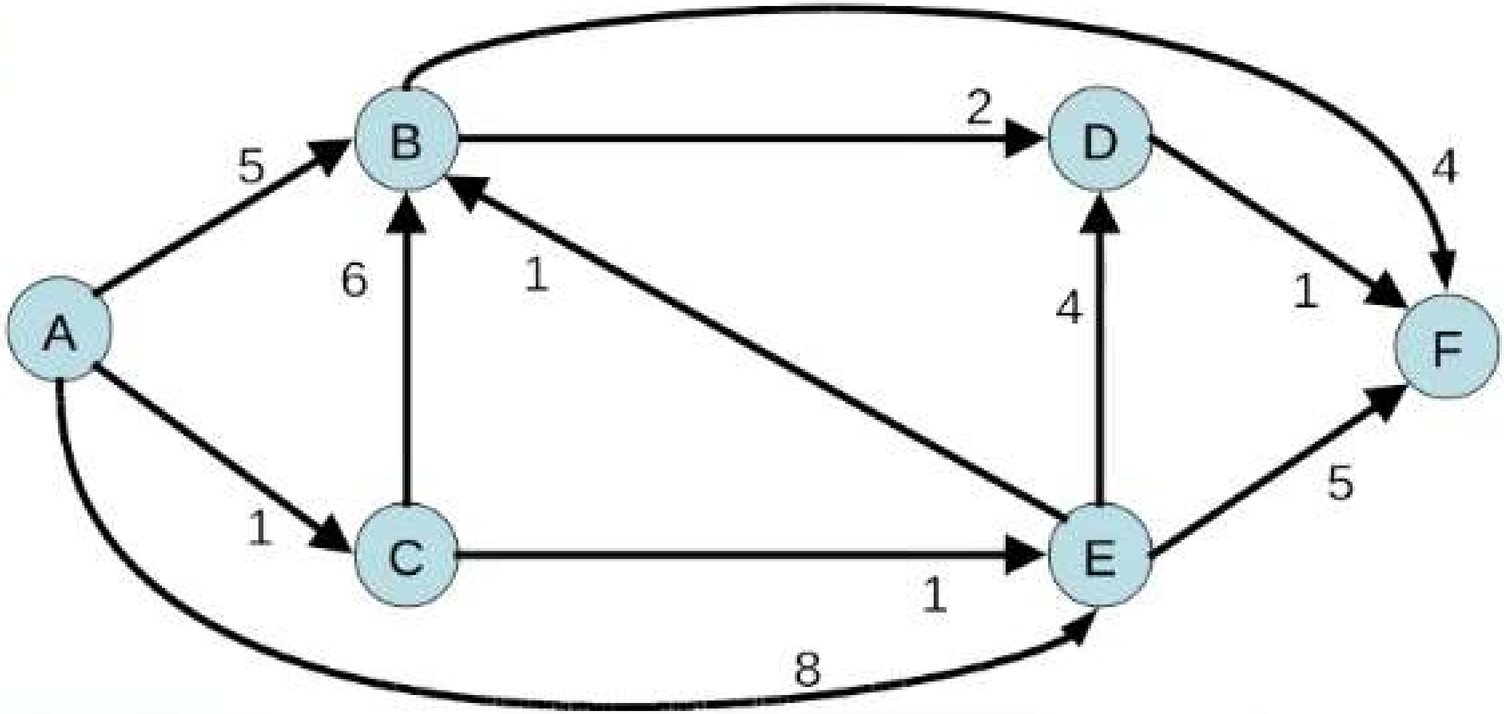
Convergenza, correttezza e complessità

- **Senza cicli negativi**, cammino minimo ha al più $|N| - 1$ archi
  $\Rightarrow$ `flag_aggiornato` rimane `false` entro iterazione $h = |N|$

- **Con cicli negativi** `flag_aggiornato` diventa `true` all'iterazione $h = |N|$ (un ciclo negativo individuato seguendo a ritroso i puntatori a partire da uno dei nodi con etichetta aggiornata)

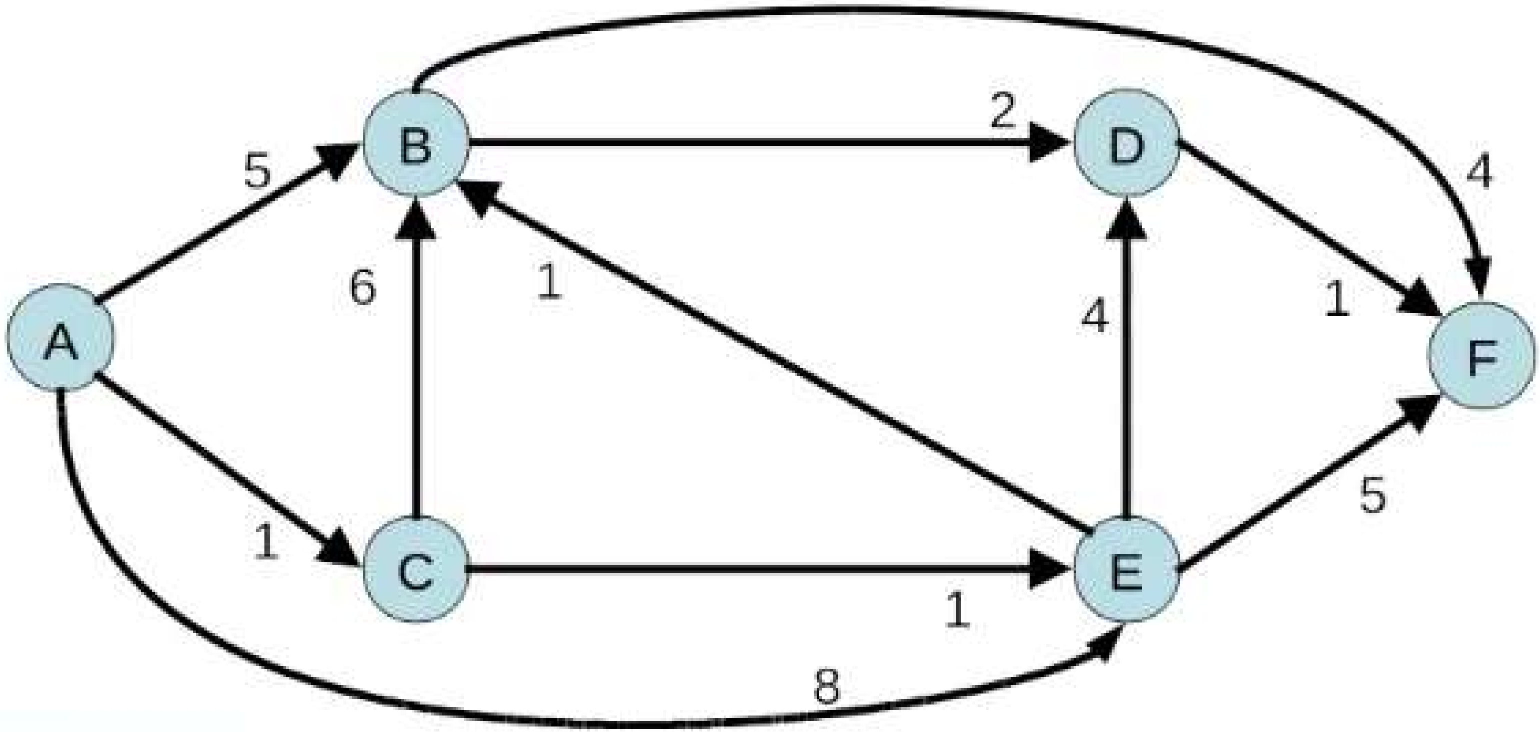
Dato un grafo pesato $G = (N, A)$ e un nodo origine $s \in N$, l'algoritmo di Bellman-Ford fornisce la soluzione ottima del problema del cammino minimo dal nodo origine s verso tutti gli altri nodi o individua un ciclo di costo negativo in $O(|N| \cdot |A|)$

Esempio



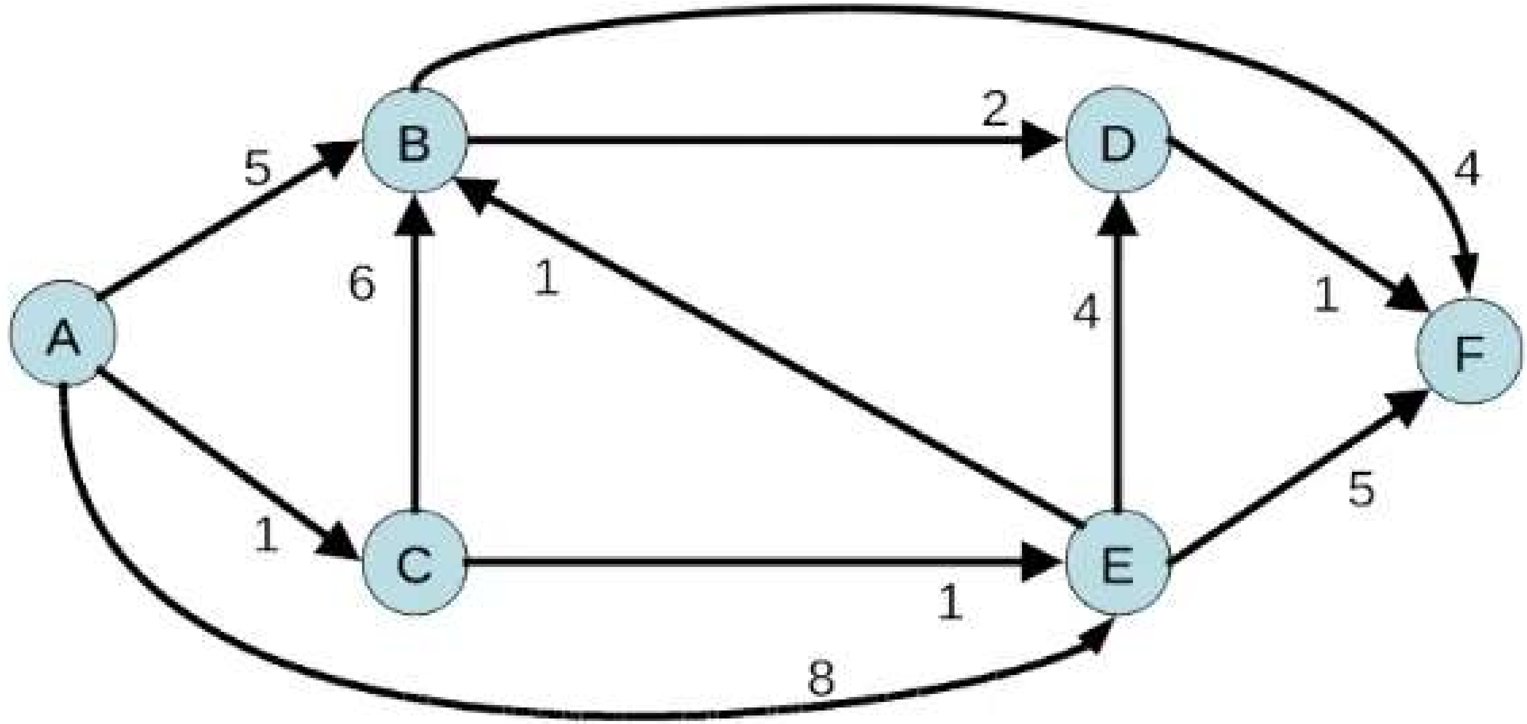
iterazione	nodo A	nodo B	nodo C	nodo D	nodo E	nodo F	*agg.*
$h = 0$	0 (∧)	$+\infty$ (∧)	$+\infty$ (∧)	$+\infty$ (∧)	$+\infty$ (∧)	$+\infty$ (∧)	true
$h = 1$	0 (∧)	5 (*A*)	1 (*A*)	$+\infty$ (∧)	8 (*A*)	$+\infty$ (∧)	
$h = 2$	0 (∧)	5 (*A*)	1 (*A*)	7 (*B*)	2 (*C*)	9 (*B*)	
$h = 3$	0 (∧)	3 (*E*)	1 (*A*)	6 (*E*)	2 (*C*)	~~8 (*D*)~~ 7 (*E*)	
$h = 4$	0 (∧)	3 (*E*)	1 (*A*)	5 (*B*)	2 (*C*)	7 (*E*)	
$h = 5$	0 (∧)	3 (*E*)	1 (*A*)	5 (*B*)	2 (*C*)	6 (*D*)	
$h = 6$	0 (∧)	3 (*E*)	1 (*A*)	5 (*B*)	2 (*C*)	6 (*D*)	

Esempio



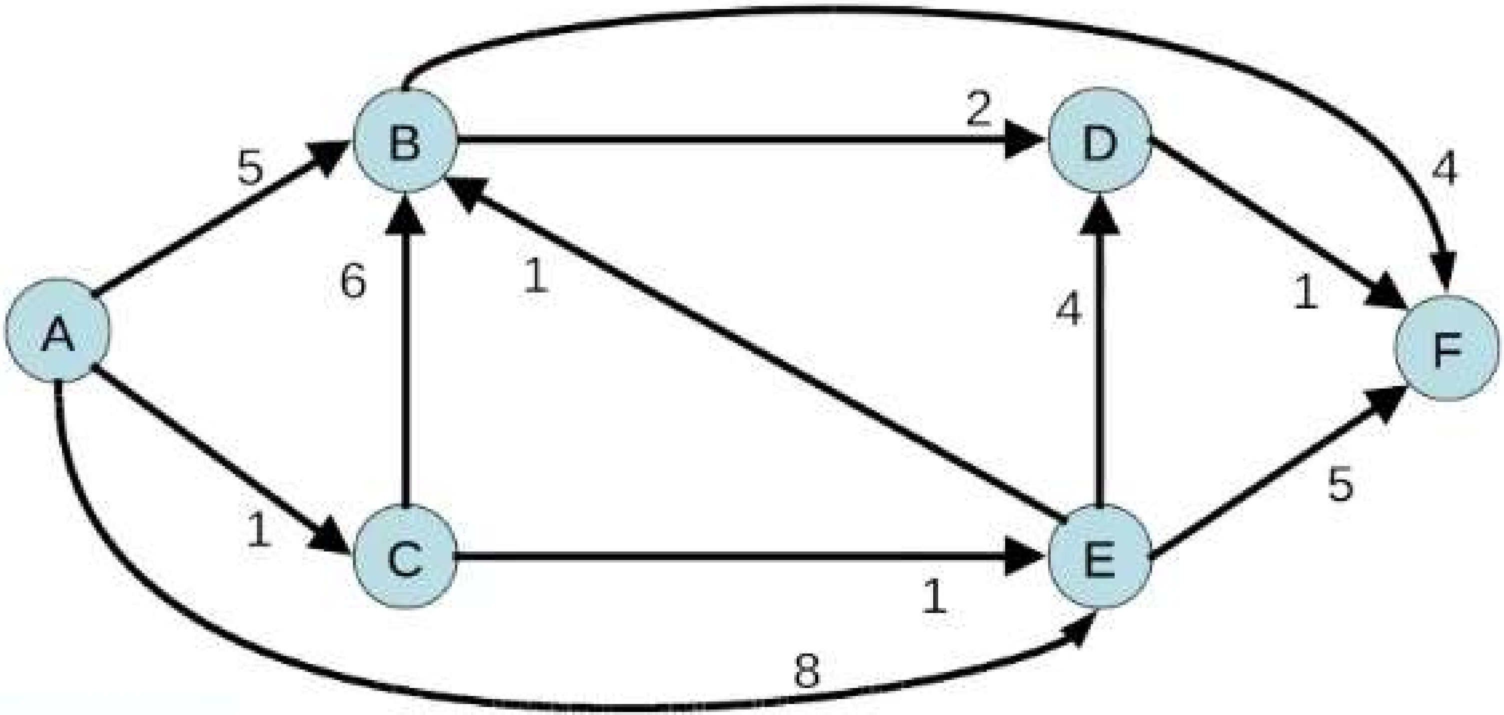
iterazione	nodo A	nodo B	nodo C	nodo D	nodo E	nodo F	*agg.*
$h=0$	$0_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	true
$h=1$	$0\ \wedge$	$5A$	$1\ A$	$\infty\ \wedge$	$8\ A$	$+\infty\ \wedge$	true
$h=2$	$0_{(\wedge)}$	$5_{(A)}$	$1_{(A)}$	$7\ B$	$2\ c$	$9\ B$	true
$h=3$	$0_{(\wedge)}$	$3_{(E)}$	$1_{(A)}$	$6_{(E)}$	$2_{(C)}$	$\cancel{8_{(D)}}\ 7_{(E)}$	
$h=4$	$0_{(\wedge)}$	$3_{(E)}$	$1_{(A)}$	$5_{(B)}$	$2_{(C)}$	$7_{(E)}$	
$h=5$	$0_{(\wedge)}$	$3_{(E)}$	$1_{(A)}$	$5_{(B)}$	$2_{(C)}$	$6_{(D)}$	
$h=6$	$0_{(\wedge)}$	$3_{(E)}$	$1_{(A)}$	$5_{(B)}$	$2_{(C)}$	$6_{(D)}$	

Esempio



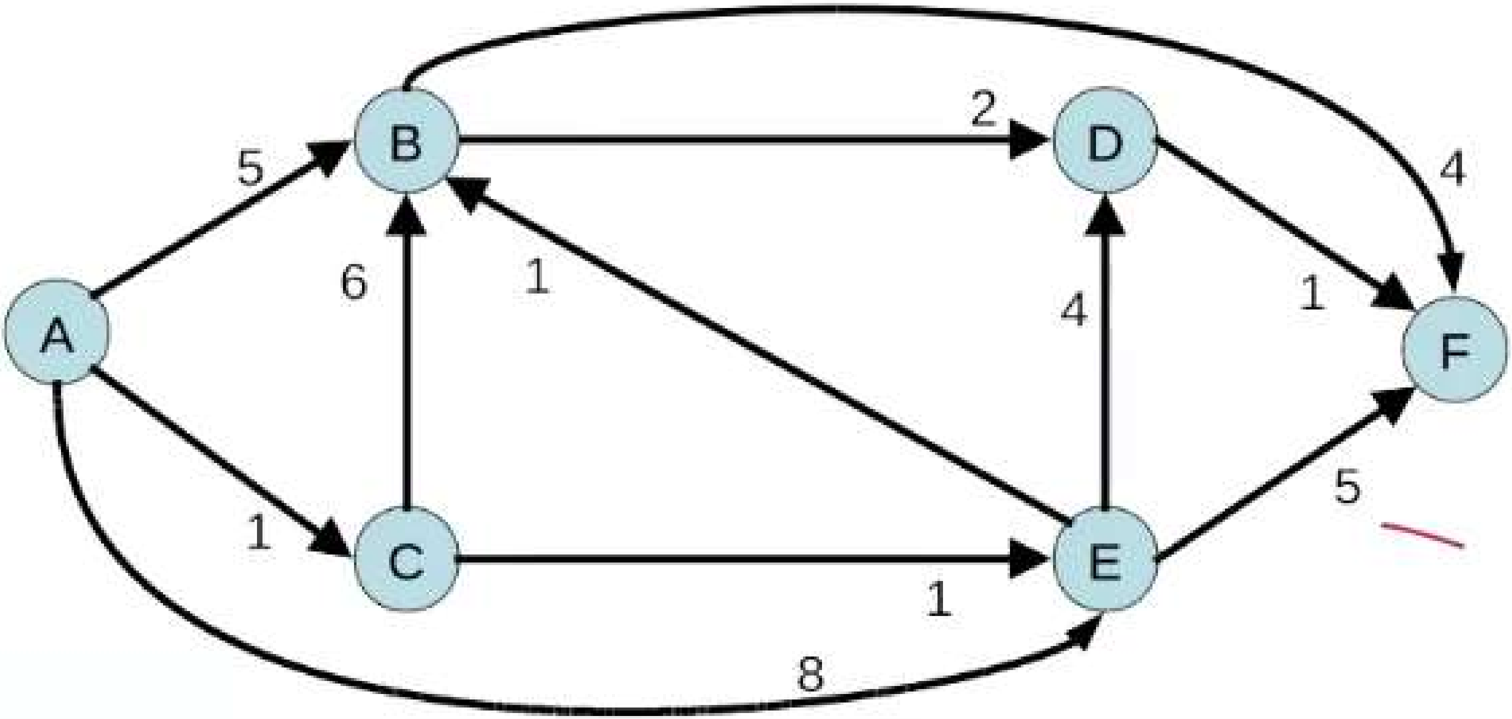
iterazione	nodo A	nodo B	nodo C	nodo D	nodo E	nodo F	*agg.*
$h = 0$	$0_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	
$h = 1$	$0_{(\wedge)}$	$5_{(A)}$	$1_{(A)}$	$+\infty_{(\wedge)}$	$8_{(A)}$	$+\infty_{(\wedge)}$	
$h = 2$	$0_{(\wedge)}$	$5_{(A)}$	$1_{(A)}$	$7_{(B)}$	$2_{(C)}$	$9_{(B)}$	DEF
$h = 3$	$0_{(\wedge)}$	3 E	$1_{(A)}$	6 E	$2_{(C)}$	~~8 D~~ 7 E	BDE
$h = 4$	$0_{(\wedge)}$	$3_{(E)}$	$1_{(A)}$	$5_{(B)}$	$2_{(C)}$	$7_{(E)}$	
$h = 5$	$0_{(\wedge)}$	$3_{(E)}$	$1_{(A)}$	$5_{(B)}$	$2_{(C)}$	$6_{(D)}$	
$h = 6$	$0_{(\wedge)}$	$3_{(E)}$	$1_{(A)}$	$5_{(B)}$	$2_{(C)}$	$6_{(D)}$	

Esempio



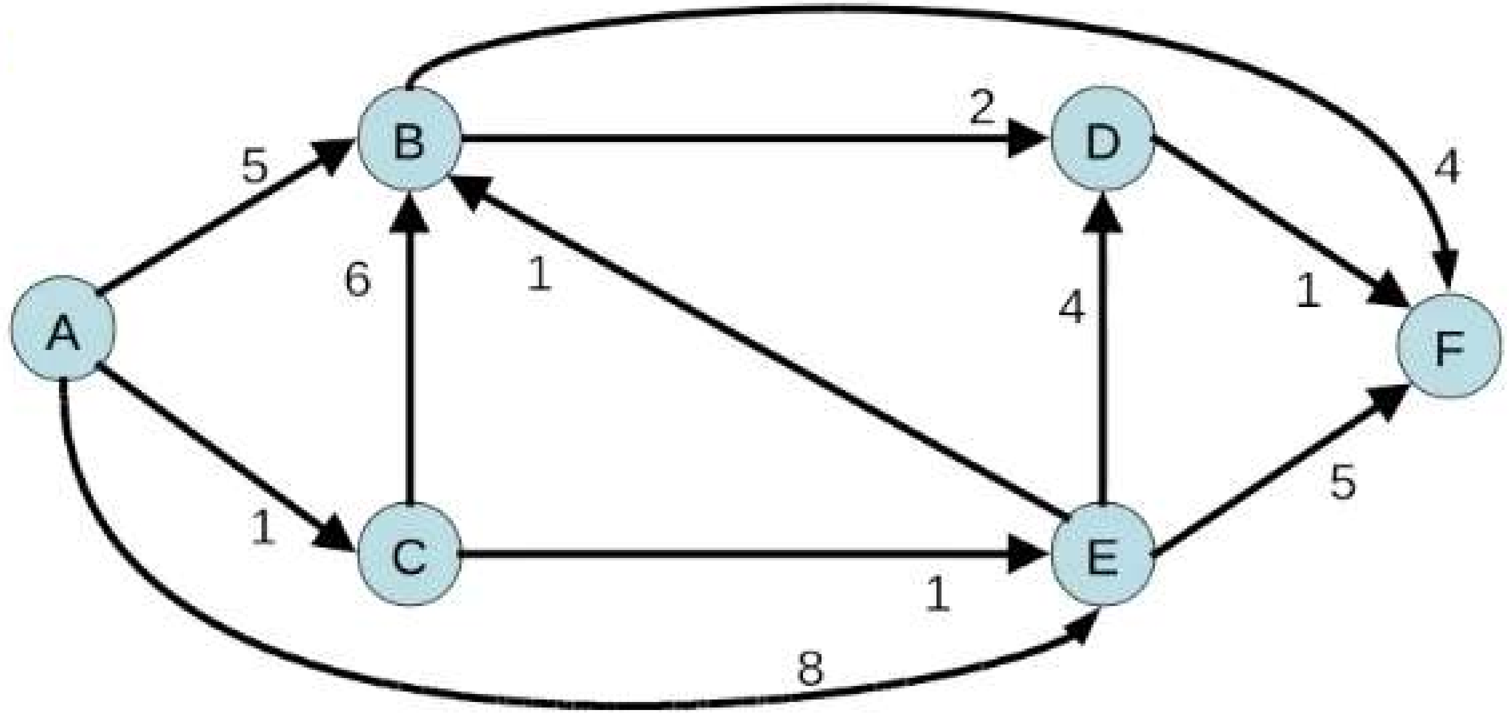
iterazione	nodo A	nodo B	nodo C	nodo D	nodo E	nodo F	*agg.*
$h = 0$	$0_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	
$h = 1$	$0_{(\wedge)}$	$5_{(A)}$	$1_{(A)}$	$+\infty_{(\wedge)}$	$8_{(A)}$	$+\infty_{(\wedge)}$	
$h = 2$	$0_{(\wedge)}$	$5_{(A)}$	$1_{(A)}$	$7_{(B)}$	$2_{(C)}$	$9_{(B)}$	
$h = 3$	$0_{(\wedge)}$	$3_{(E)}$	$1_{(A)}$	$6_{(E)}$	$2_{(C)}$	$\cancel{8_{(D)}}\ 7_{(E)}$	~~B~~ ~~D~~ ~~E~~
h = 4	0 ∧	3 E	1 A	5 B	2 C	7 E	D
$h = 5$	$0_{(\wedge)}$	$3_{(E)}$	$1_{(A)}$	$5_{(B)}$	$2_{(C)}$	$6_{(D)}$	
$h = 6$	$0_{(\wedge)}$	$3_{(E)}$	$1_{(A)}$	$5_{(B)}$	$2_{(C)}$	$6_{(D)}$	

Esempio



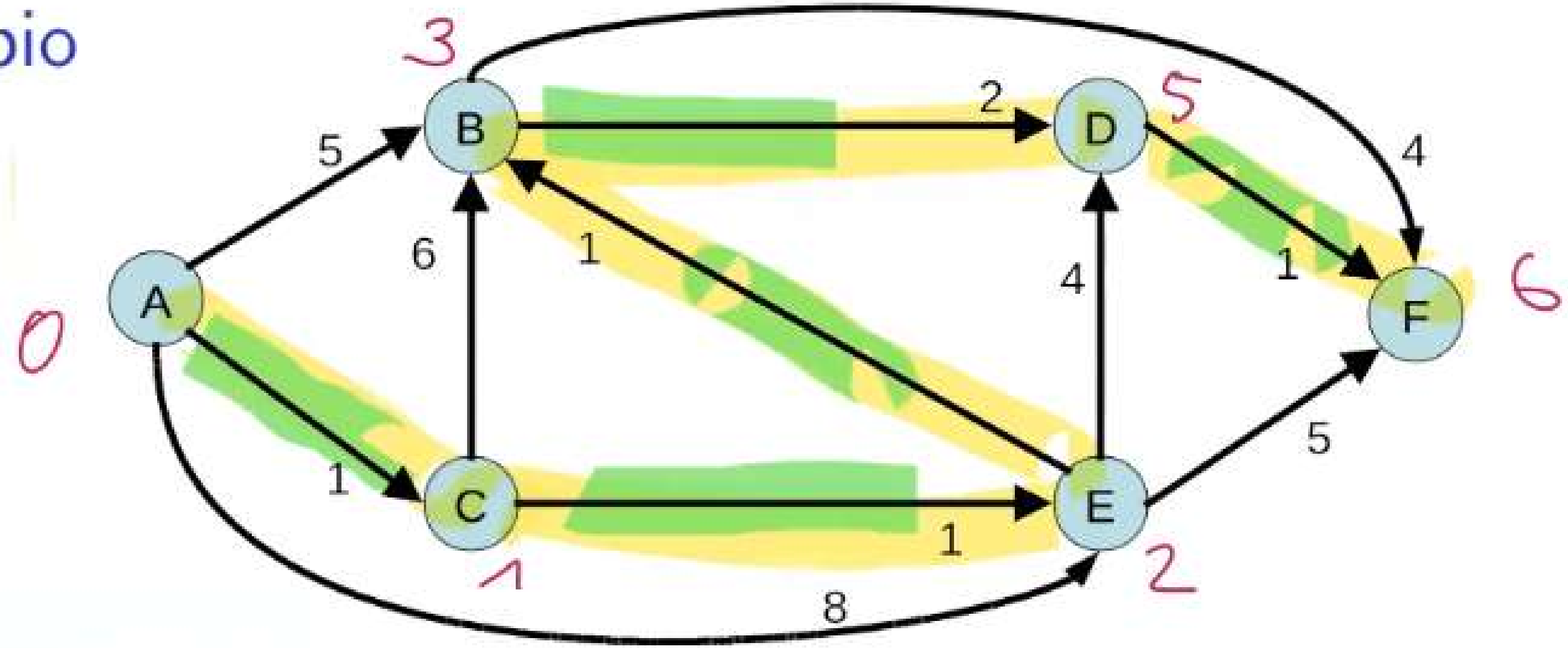
iterazione	nodo A	nodo B	nodo C	nodo D	nodo E	nodo F	*agg.*
$h = 0$	$0_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	
$h = 1$	$0_{(\wedge)}$	$5_{(A)}$	$1_{(A)}$	$+\infty_{(\wedge)}$	$8_{(A)}$	$+\infty_{(\wedge)}$	
$h = 2$	$0_{(\wedge)}$	$5_{(A)}$	$1_{(A)}$	$7_{(B)}$	$2_{(C)}$	$9_{(B)}$	
$h = 3$	$0_{(\wedge)}$	$3_{(E)}$	$1_{(A)}$	$6_{(E)}$	$2_{(C)}$	$\cancel{8_{(D)}}\ 7_{(E)}$	
$h = 4$	$0_{(\wedge)}$	$3_{(E)}$	$1_{(A)}$	$5_{(B)}$	$2_{(C)}$	$7_{(E)}$	D
h=5	0 ∧	3 E	1 A	5 B	2 C	6 D	F
$h = 6$	$0_{(\wedge)}$	$3_{(E)}$	$1_{(A)}$	$5_{(B)}$	$2_{(C)}$	$6_{(D)}$	

Esempio



iterazione	nodo A	nodo B	nodo C	nodo D	nodo E	nodo F	*agg.*
$h = 0$	$0\ (\wedge)$	$+\infty\ (\wedge)$	$+\infty\ (\wedge)$	$+\infty\ (\wedge)$	$+\infty\ (\wedge)$	$+\infty\ (\wedge)$	
$h = 1$	$0\ (\wedge)$	$5\ (A)$	$1\ (A)$	$+\infty\ (\wedge)$	$8\ (A)$	$+\infty\ (\wedge)$	
$h = 2$	$0\ (\wedge)$	$5\ (A)$	$1\ (A)$	$7\ (B)$	$2\ (C)$	$9\ (B)$	
$h = 3$	$0\ (\wedge)$	$3\ (E)$	$1\ (A)$	$6\ (E)$	$2\ (C)$	$\cancel{8\ (D)}\ 7\ (E)$	
$h = 4$	$0\ (\wedge)$	$3\ (E)$	$1\ (A)$	$5\ (B)$	$2\ (C)$	$7\ (E)$	
$h = 5$	$0\ (\wedge)$	$3\ (E)$	$1\ (A)$	$5\ (B)$	$2\ (C)$	$6\ (D)$	F
$h = 6$	$0\ \wedge$	$3\ E$	$1\ A$	$5\ B$	$2\ C$	$6\ D$	$\emptyset$

Esempio



iterazione	nodo A	nodo B	nodo C	nodo D	nodo E	nodo F	*agg.*
$h = 0$	$0_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	
$h = 1$	$0_{(\wedge)}$	$5_{(A)}$	$1_{(A)}$	$+\infty_{(\wedge)}$	$8_{(A)}$	$+\infty_{(\wedge)}$	
$h = 2$	$0_{(\wedge)}$	$5_{(A)}$	$1_{(A)}$	$7_{(B)}$	$2_{(C)}$	$9_{(B)}$	
$h = 3$	$0_{(\wedge)}$	$3_{(E)}$	$1_{(A)}$	$6_{(E)}$	$2_{(C)}$	$\cancel{8_{(D)}}\ 7_{(E)}$	
$h = 4$	$0_{(\wedge)}$	$3_{(E)}$	$1_{(A)}$	$5_{(B)}$	$2_{(C)}$	$7_{(E)}$	
$h = 5$	$0_{(\wedge)}$	$3_{(E)}$	$1_{(A)}$	$5_{(B)}$	$2_{(C)}$	$6_{(D)}$	
$h = 6$	$0_{(\wedge)}$	$3_{(E)}$	$1_{(A)}$	$5_{(B)}$	$2_{(C)}$	$6_{(D)}$	

Implementazione mediamente più efficiente

π_s := 0; set $p(s) = \wedge$; Aggiornati := $\{s\}$

for each $v \in N - s$ { set π_v:=$+\infty$, set $p(v)$:= $\wedge$ }

for h = 1 to $|N|$ {

set π' := π; set Aggiornati' := Aggiornati;

set Aggiornati := $\emptyset$;

for all ($(i,j) \in A : i \in$ Aggiornati' $\wedge\, \pi_j > \pi'_i + c_{ij}$) do {

set π_j := $\pi'_i + c_{ij}$

set $p(j)$:= i;

set Aggiornati := Aggiornati $\cup$ $\{j\}$;

}

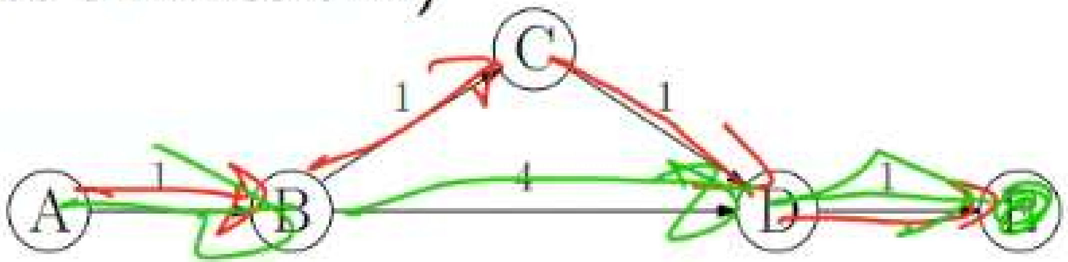
if (Aggiornati = $\emptyset$) then { STOP: π è ottima }

}

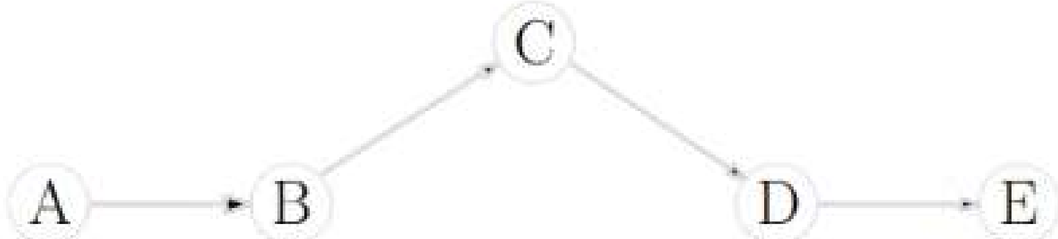
STOP: $\exists$ ciclo di costo negativo.

Cammini minimi con al più $\bar{h}$ archi

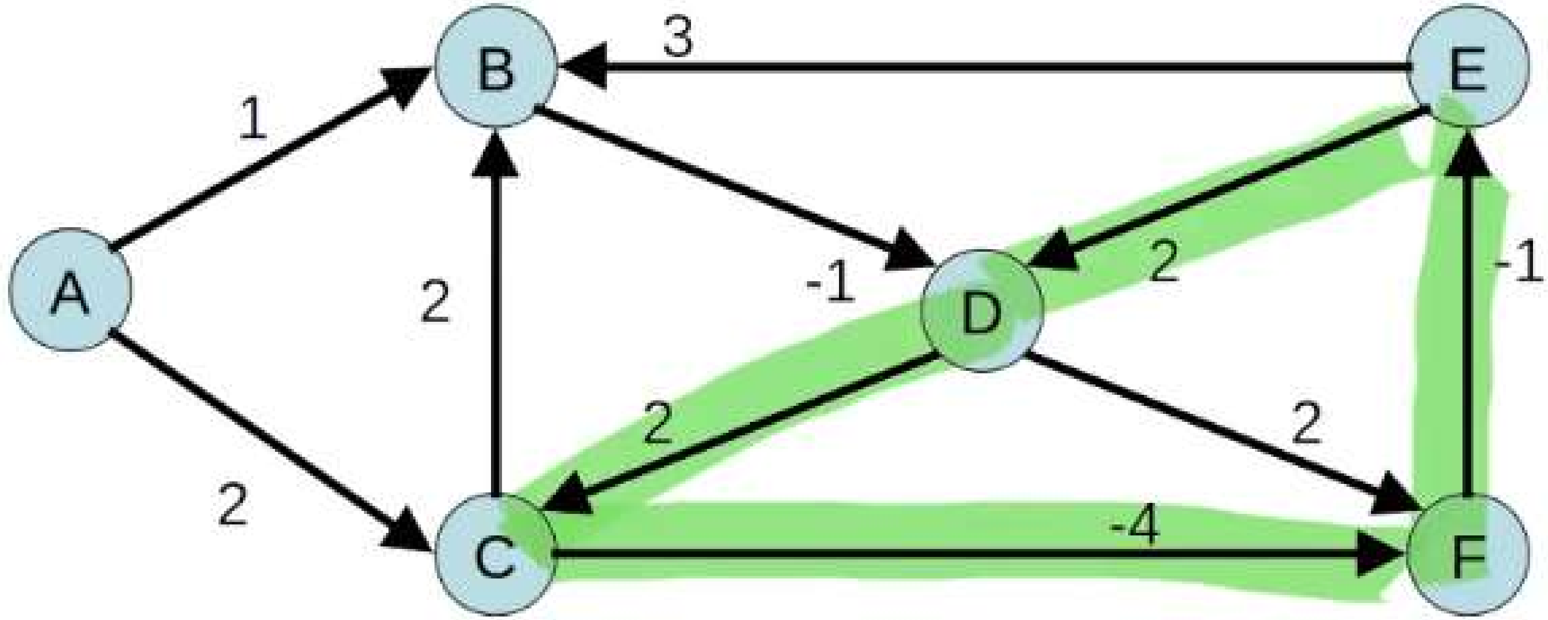
- **Fermarsi all'iterazione** $h = \bar{h}$ (esempi, definiti anche con cicli negativi)
- Cammini individuati saltando alla riga precedente
- Alberi e grafi dei cammini minimi non sono definiti (etichette non necessariamente ammissibili!)



iterazione	nodo A	nodo B	nodo C	nodo D	nodo E	Aggiornati
$h = 0$	$0_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	A
$h = 1$	$0_{(\wedge)}$	$1_{(A)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	B
$h = 2$	$0_{(\wedge)}$	$1_{(A)}$	$2_{(B)}$	$5_{(B)}$	$+\infty_{(\wedge)}$	C, D
$h = 3$	$0_{(\wedge)}$	$1_{(A)}$	$2_{(B)}$	$3_{(C)}$	$6_{(D)}$	D, E



Esempio



iteraz.	nodo A	nodo B	nodo C	nodo D	nodo E	nodo F	Agg.i
$h=0$	$0_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	A
$h=1$	$0_{(\wedge)}$	$+1_{(A)}$	$+2_{(A)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	B, C
$h=2$	$0_{(\wedge)}$	$+1_{(A)}$	$+2_{(A)}$	$0_{(B)}$	$+\infty_{(\wedge)}$	$-2_{(C)}$	D, F
$h=3$	$0_{(\wedge)}$	$+1_{(A)}$	$+2_{(A)}$	$0_{(B)}$	$-3_{(F)}$	$-2_{(C)}$	E
$h=4$	$0_{(\wedge)}$	$0_{(E)}$	$+2_{(A)}$	$-1_{(E)}$	$-3_{(F)}$	$-2_{(C)}$	B, D
$h=5$	$0_{(\wedge)}$	$0_{(E)}$	$+1_{(D)}$	$-1_{(E)}$	$-3_{(F)}$	$-2_{(C)}$	C
$h=6$	$0_{(\wedge)}$	$0_{(E)}$	$+1_{(D)}$	$-1_{(E)}$	$-3_{(F)}$	$-3_{(C)}$	F

Albero, grafo, ciclo?

Nota storica

Condizioni di Bellman: ottimalità dei cammini minimi

Dato un grafo $G = (N, A)$ con costi c sugli archi e un nodo origine $s \in N$, e date delle etichette π_v per ogni nodo $v \in N$, queste rappresentano i costi dei cammini minimi da s verso v *se e solo se* π_v è la lunghezza di un cammino ammissibile da s a v e $\pi_j - \pi_i \leq c_{ij}$, $\forall\ (i, j) \in A$.

- dimostrabili a partire dal principio di sub-ottimalità
- principio sfruttato da Bellman-Ford per derivare il loro algoritmo come applicazione di *programmazione dinamica*

Algoritmo di Dijkstra (label setting): $\boxed{c_{ij} \geq 0 \,,\ \forall\, (i,j) \in A}$

0) π_s := 0; set $p(s)$:= $\wedge$; set $S := \emptyset$; set $\bar{S} := N$;

for each $v \in N \setminus \{s\}$ { set π_v:=+∞, set $p(v)$:= $\wedge$ }

1) set $\hat{v}$:=$\arg\min_{i \in \bar{S}} \{\pi_i\}$

set $S := S \cup \{\hat{v}\}$; set $\bar{S} := \bar{S} \setminus \{\hat{v}\}$;

if $\bar{S} = \emptyset$ then STOP: $\boxed{\pi \text{ è ottimo.}}$

2) for all ($j \in \Gamma_{\hat{v}} \cap \bar{S} : \pi_j > \pi_{\hat{v}} + c_{\hat{v}j}$) do {

set $\pi_j := \pi_{\hat{v}} + c_{\hat{v}j}$

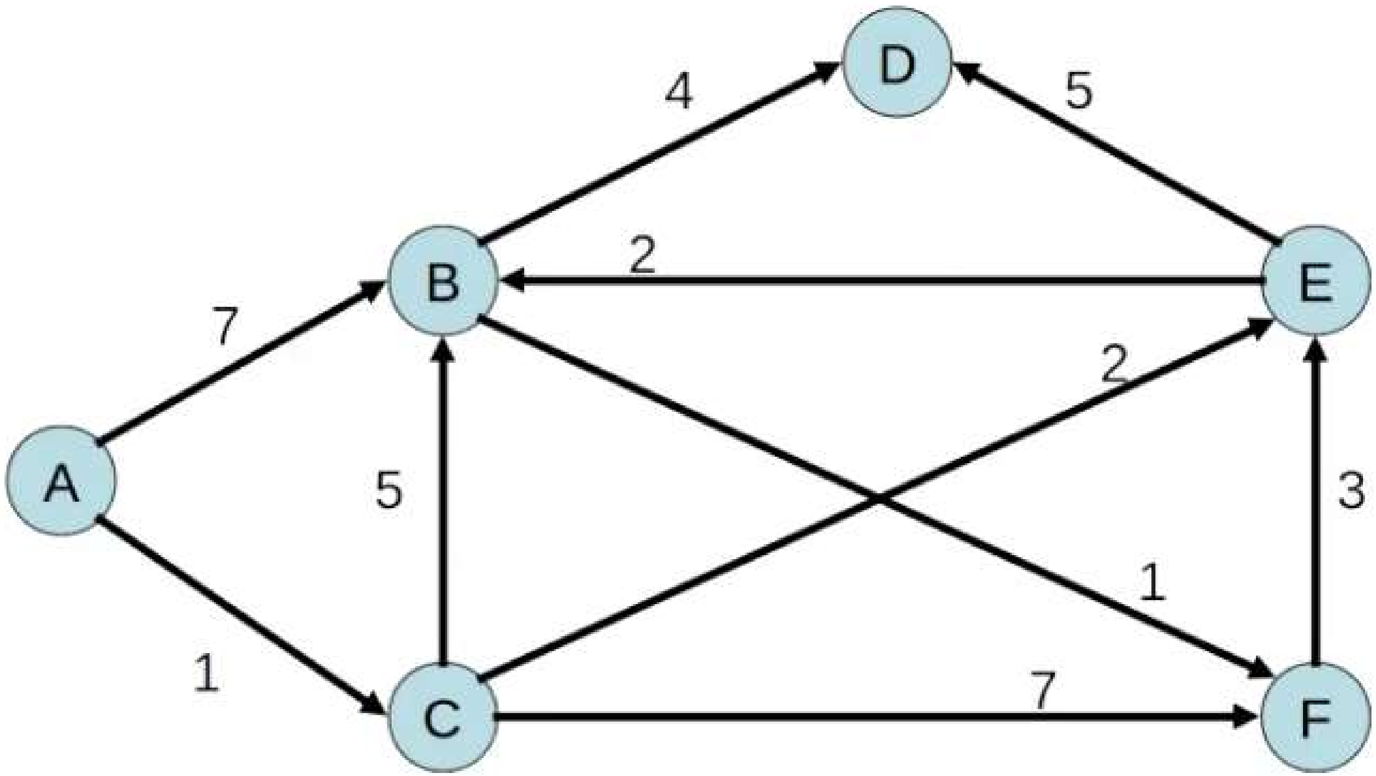
set $p(j) := \hat{v}$;

}

go to 1)

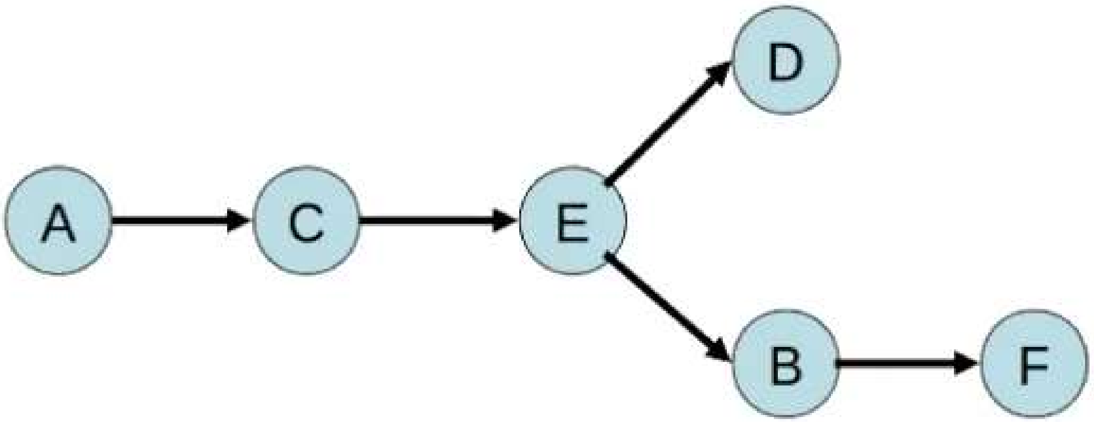
$\Gamma_{\hat{v}} : (\hat{v}, j)$

Esempio



it.	nodo A	nodo B	nodo C	nodo D	nodo E	nodo F	$\bar{S}$	$\hat{v}$
0	$0_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	A, B, C, D, E, F	
1	$*$	$7_{(A)}$	$1_{(A)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	B, C, D, E, F	A
2		$6_{(C)}$	$*$		$3_{(C)}$	$8_{(C)}$	B, D, E, F	C
3		$5_{(E)}$		$8_{(E)}$	$*$		B, D, F	E
4		$*$		$-$		$6_{(B)}$	D, F	B
5					$\times$	$*$	D	F
6				$*$			$\emptyset$	D

Esempio



it.	nodo A	nodo B	nodo C	nodo D	nodo E	nodo F	$\bar{S}$	$\hat{v}$
0	$0_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	A, B, C, D, E, F	
1	$*$	$7_{(A)}$	$1_{(A)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	B, C, D, E, F	A
2		$6_{(C)}$	$*$		$3_{(C)}$	$8_{(C)}$	B, D, E, F	C
3		$5_{(E)}$		$8_{(E)}$	$*$		B, D, F	E
4		$*$		$-$		$6_{(B)}$	B, D	B
5					$\times$	$*$	D	F
6				$*$			$\emptyset$	D

$*$: etichetta fissata.

$-$: etichetta controllata ma non aggiornata.

$\times$: etichetta non controllata perché il nodo è già fissato.

Convergenza e correttezza

- Ad ogni iterazione un nodo passa in S: converge in $|N|$ iterazioni!

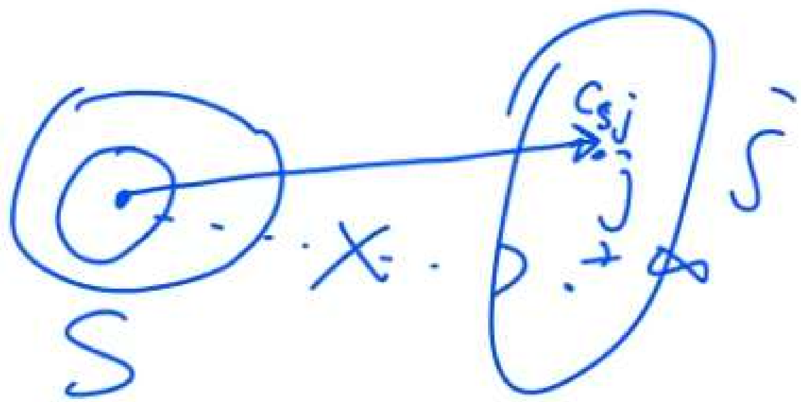
Lemma. Alla fine di ogni iterazione: per ogni nodo $i \in N$:

- $i \in S$: π_i è costo di un cammino minimo da s a i
  $p(i)$ è il predecessore di i su tale cammino
  tale cammino utilizza solo nodi in S;
- $i \in \bar{S}$: π_i è costo cammino minimo da s a i *con solo nodi in S*
  $p(i)$ è il predecessore di i su tale cammino.

IT. 1

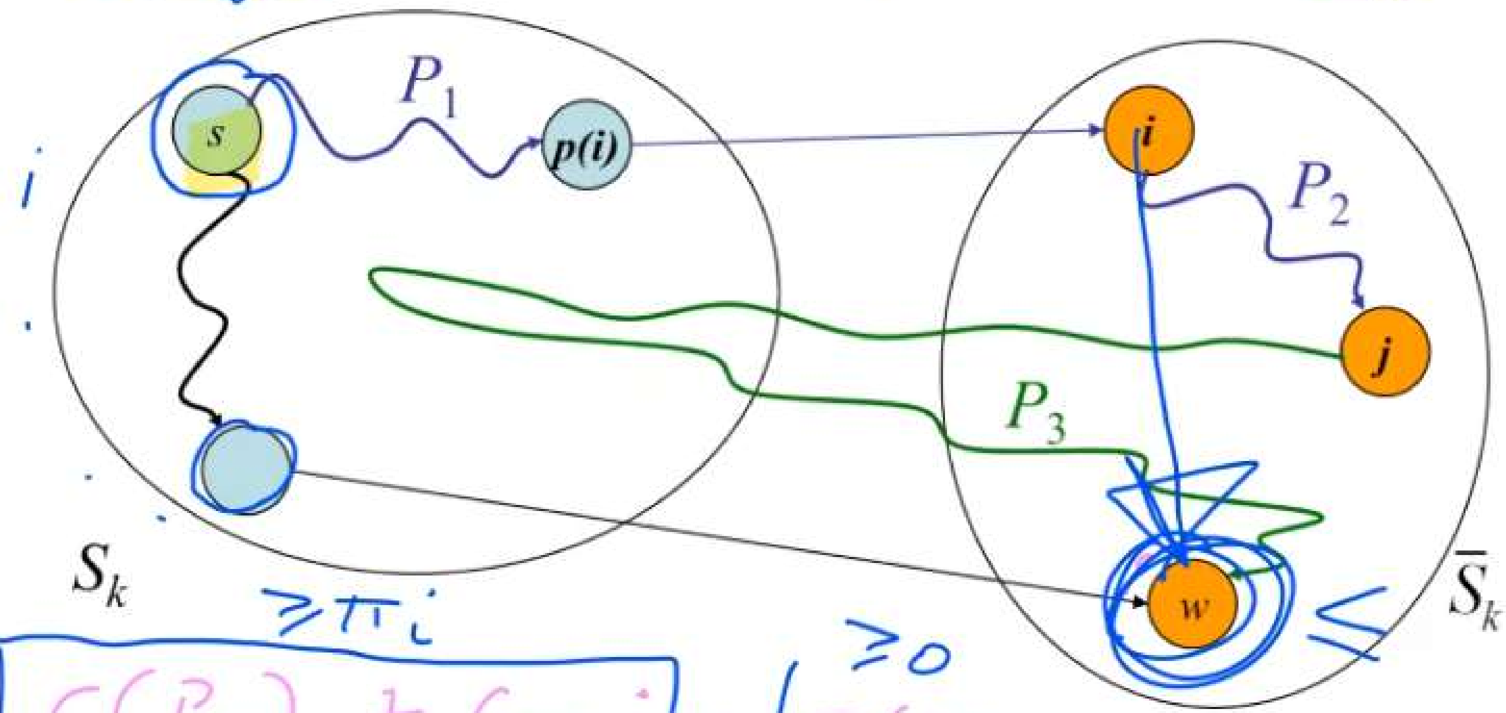
Dimostrazione per induzione: iterazione 1

- $j \in \Gamma_s$: $\pi_j = c_{sj}$ e $p(j) = s$ ✓
- $j \notin \Gamma_s$: $\pi_j = +\infty$ e $p(j) = \wedge$ ✓



Dimostrazione: fine iterazione $k+1$, nodi in S

- $S_{k+1} = S_k + w$ $\qquad$ $w = v$!
- S_k ✓(non sono cambiati etichette e puntatori)
- Verifica: $\pi_w = \pi^*_w$, $\quad p(w)$ predecessore, $\quad$ solo nodi in S_{k+1}



$$\pi_w = \overbrace{\boxed{c(P_1) + c_{p(i)i}}}^{\geq \pi_i} + \overbrace{c(P_2) + c(P_3)}^{\geq 0}$$

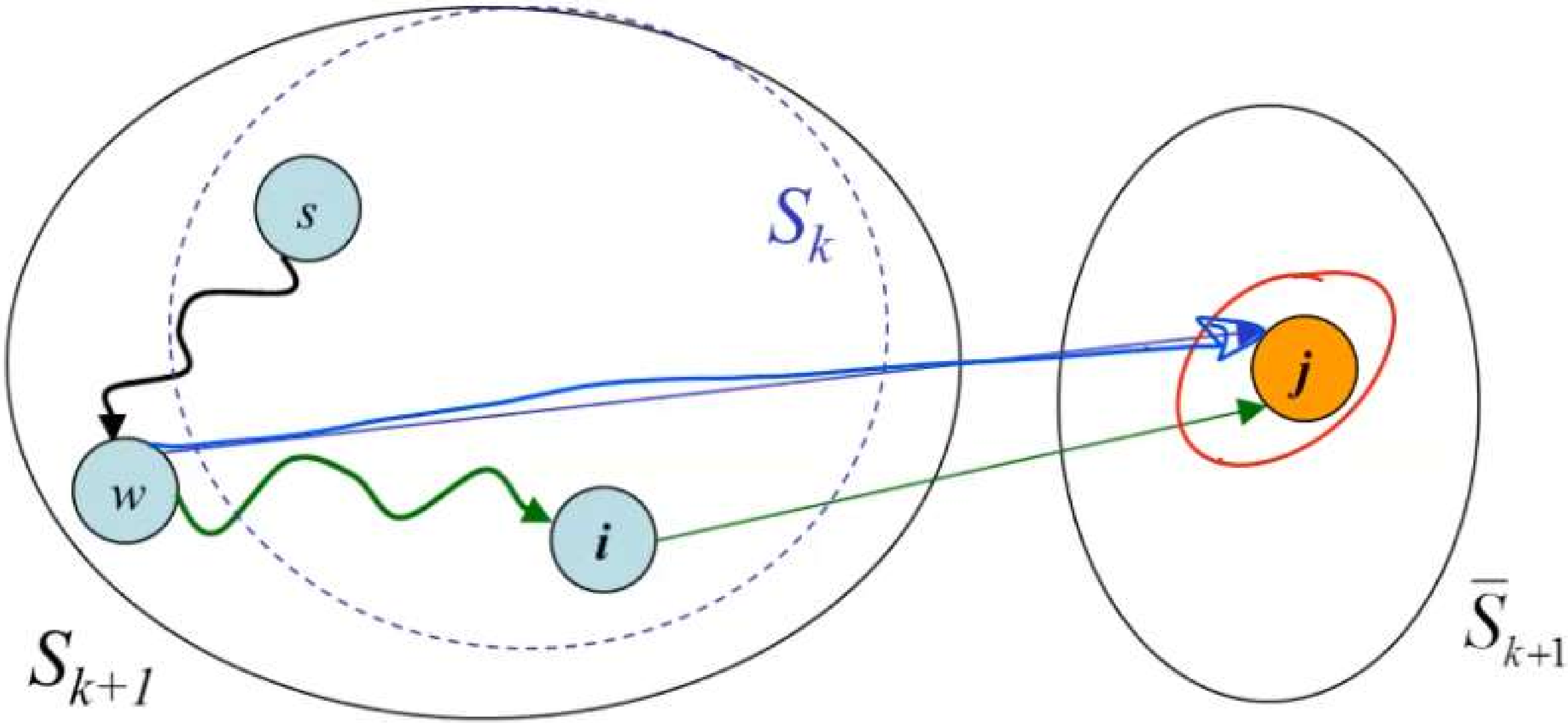
- Se $\pi_w > \pi^*_w$, devo usare nodo $i \in \bar{S}_k$ (hp induttiva su $w \in \bar{S}_k$)
- ma $\boxed{\pi_w \leq \pi_i}$ e $c(P_2) + c(P_3) \geq 0$: assurdo!

$= c(P_1) + c_{p(i)i}$

$\pi_i \leq c(P_1) + c_{p(i)i}$

Dimostrazione: fine iterazione $k+1$, nodi in $\bar{S}$

- Passando da w, migliora il costo di cammino minimo con soli nodi in $S \ni w$?



min solo S_K

- $j \notin \Gamma_w$: $c(s \rightsquigarrow w) + c(w \rightsquigarrow i) \geq \pi_i$, $\quad \pi_i + c_{ij} \geq \pi_j$. $\quad \pi_j$ rimane ✓
- $j \in \Gamma_w$: anche $s \rightsquigarrow w \to j$. $\quad \pi_j$ controllato (e aggiornato) ✓

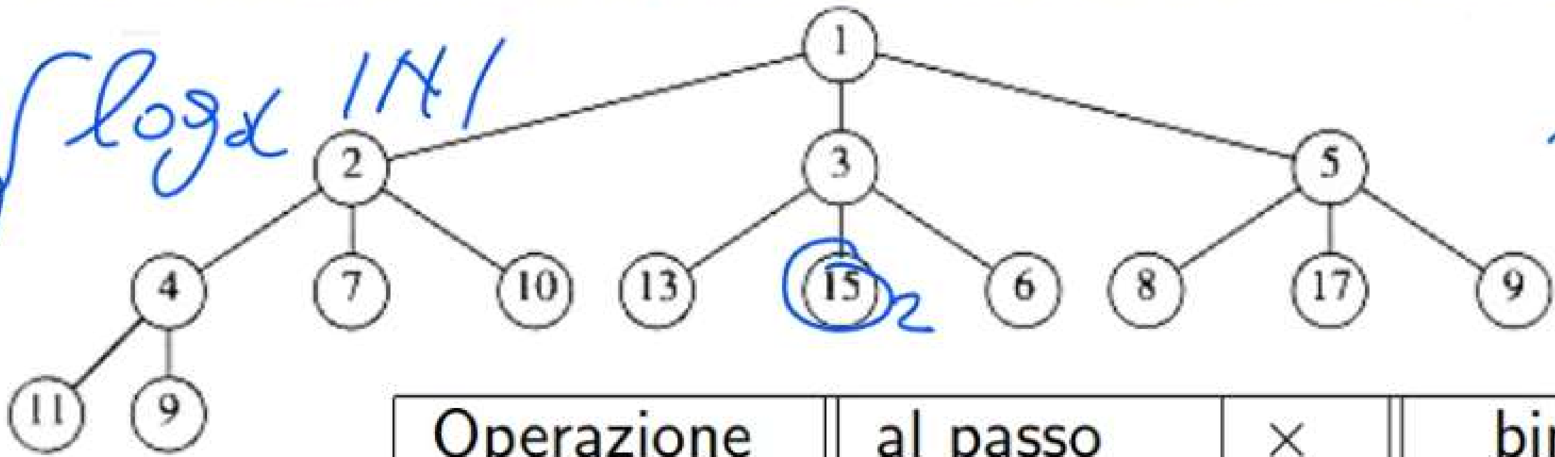
Correttezza e complessità

Dati $G = (N, A)$, c_{ij} e s, l'algoritmo di Dijkstra risolve il problema del cammino minimo dall'origine s verso tutti gli altri nodi in tempo $O(|N|^2)$

- Alla fine, tutti i nodi in S ✓
- Inizializzazione: $O(|N|)$
- Ricerca minimo: $|N| \cdot O(N) = O(|N|^2)$
- Aggiornamenti etichette: **in tutto** (complessità ammortizzata) $O(|A|)$ (ogni arco controllato al massimo **una volta!**)
- in tutto $O(|N| + |N|^2 + |A|) = O(|N|^2)$

Implementazioni efficienti: heap per ricerca minimo

$\log_d |N|$



$\sim \bar{S}$

d-heap

Operazione	al passo	$\times$	binario
create	init	1	$O(1)$
insert	init	$\|N\|$	$O(\log \|N\|)$
find-min	trova $\hat{v}$	$\|N\|$	$O(1)$
delete-min	aggiorna S	$\|N\|$	$O(\log \|N\|)$
decrease-key	aggiorna π	$\|A\|$	$O(\log \|N\|)$

INIT

1

2

- Complessità Dijkstra con heap binario: $O(|A| \log |N|)$
- Con *Fibonacci heap* si arriva a implementazioni $O(|A| + |N| \log |N|)$

$\bar{S}$ lista $\to O(|N^2|)$

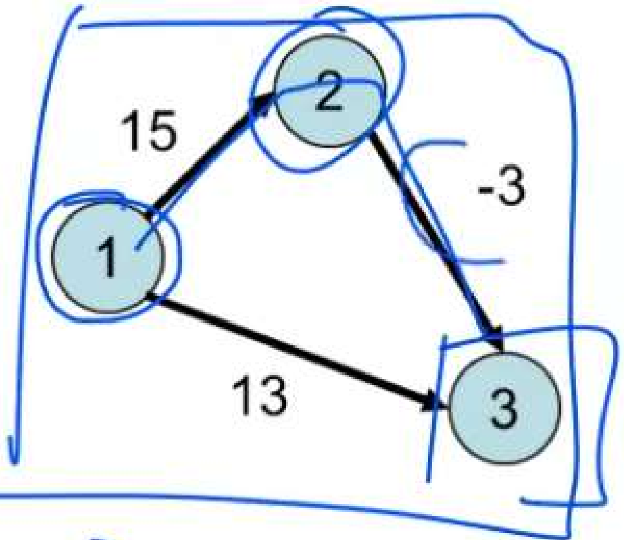
$\bar{S}$ d-heap $\to O(|A| \log_d |N|)$

Applicabilità

Dijkstra applicabile solo se $\boxed{c_{ij} \geq 0}$ **per tutti gli archi** $(i,j) \in A$

$\nexists$ arco < 0

$\exists$ arco < 0



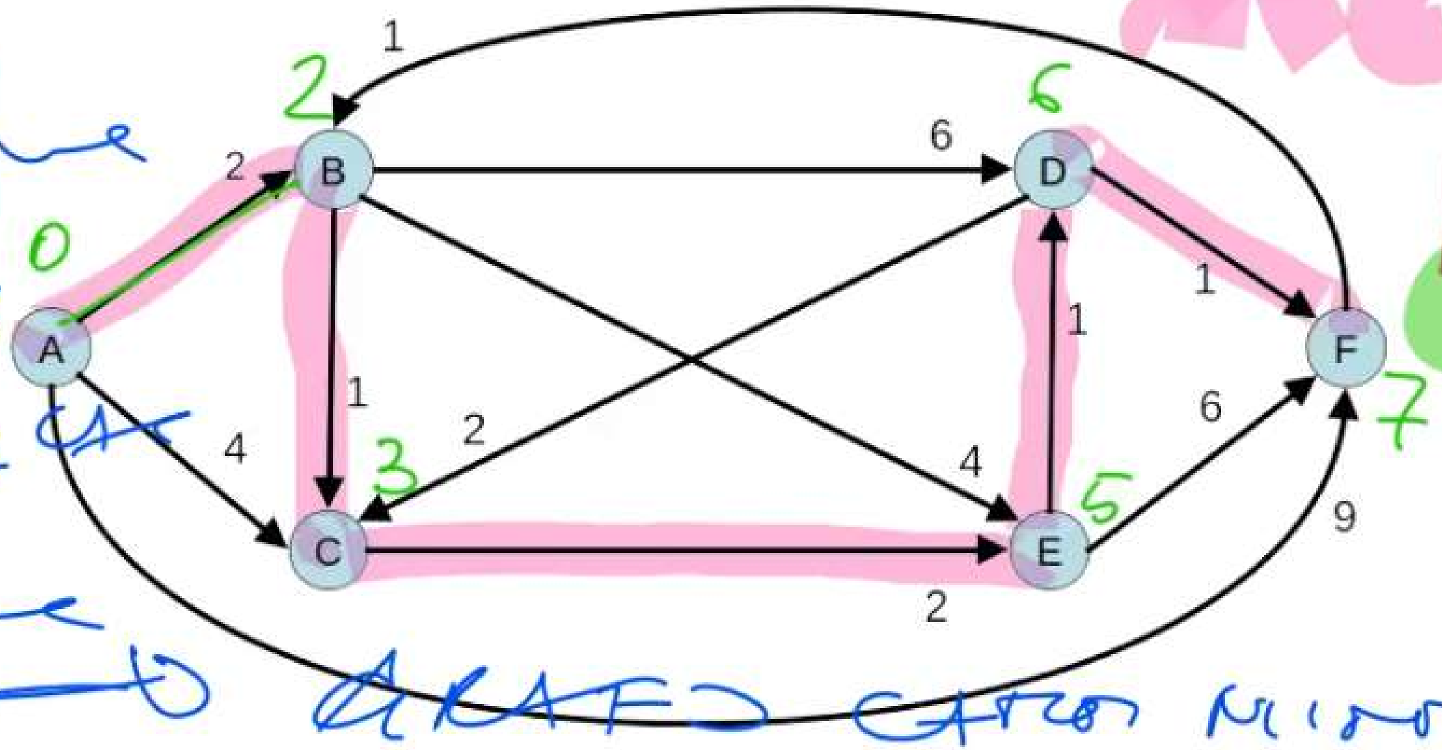
1 2 3

0 0

1 15 13

δ: 3

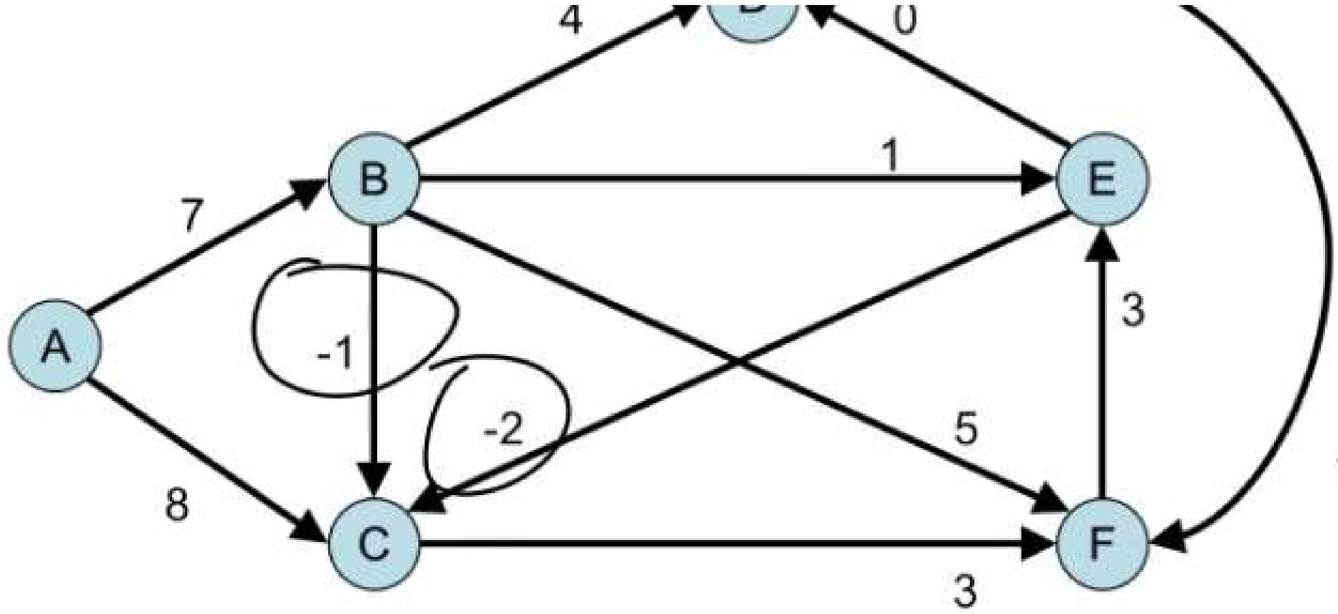
Esercizio



it.	nodo A	nodo B	nodo C	nodo D	nodo E	nodo F	$\bar{S}$	$\hat{v}$
0	$0_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	A, B, C, D, E, F	//
1	$*$	$2_{(A)}$	$4_{(A)}$	$+\infty_{(\wedge)}$	$+\infty_{(\wedge)}$	$9_{(A)}$	B, C, D, E, F	A
2		$*$	$3_{(B)}$	$8_{(B)}$	$6_{(B)}$		C, D, E, F	B
3			$*$		$5_{(C)}$		D, E, F	C
4				$6_{(E)}$	$*$	$-$	D, F	E
5			$\times$	$*$		$7_{(D)}$	F	D
6		$\times$				$*$	$\emptyset$	F

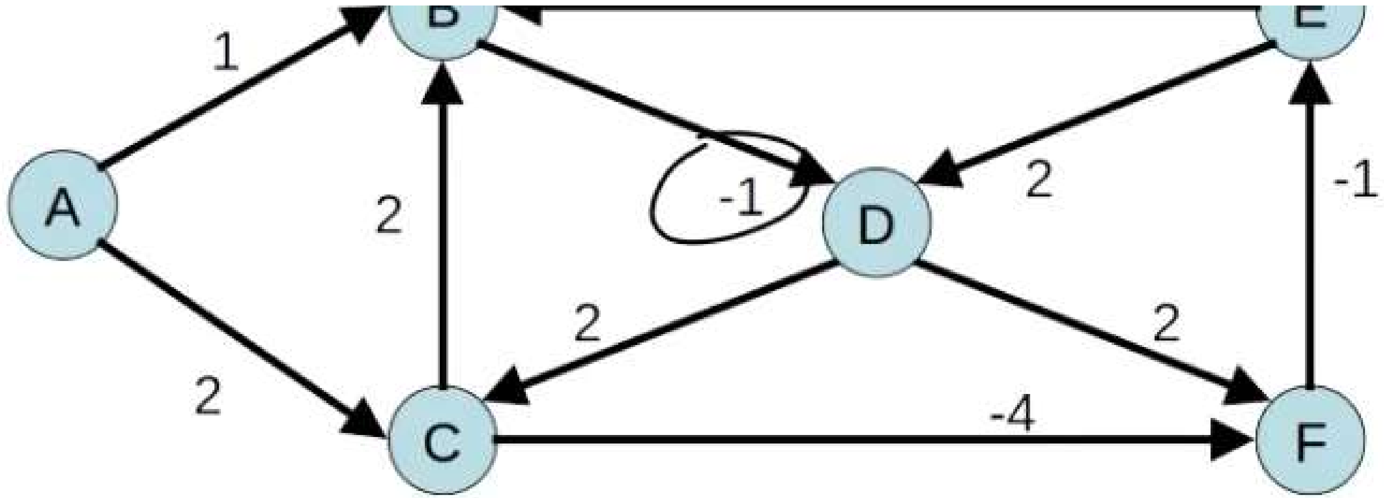
$*$: fissata. $-$: controllata ma non aggiornata. $\times$: non controllata.

Esercizi...



- Trovare i cammini minimi da A verso tutti gli altri nodi: posso usare Bellman-Ford? Posso usare Dijkstra? Quale conviene usare? BF
- Trovare i cammini minimi da A verso tutti gli altri nodi con al massimo 3 archi: quale algoritmo uso? BF

Esercizi...



- Trovare i cammini minimi sa A verso tutti gli altri nodi: posso usare Bellman-Ford? Posso usare Dijkstra? Quale conviene usare?
- Trovare i cammini minimi da A verso tutti gli altri nodi con al massimo 3 archi: quale algoritmo uso?