- Si discuta la complessità computazionale dell'algoritmo di Bellman-Ford.
    - L'algoritmo di Bellman-Ford ha una complessità $O(N * A)$), in quanto viene fato un numero di iterazioni pari al numero N di nodi con un ciclo che esamina le etichette duali su tutti gli archi A. Qualora esistesse un ciclo negativo (abbiamo aggiornato almeno un'etichetta) oppure tutte le etichette sono stabili, l'algoritmo termina la sua esecuzione.

- Si discuta la complessità computazionale dell'algoritmo di Dijkstra per il problema del cammino minimo
    - Il prof riporta: "Si consiglia di riportare i passi dell'algoritmo di Dijkstra e, quindi, discuterne la complessità come fatto nelle dimostrazione della Proprietà 10 alla fine del paragrafo 6.1 delle dispense sugli algoritmi per problemi di cammino minimo. Quindi si può approfondire con qualche dettaglio sull'influenza della scelta di opportune strutture dati per migliorare l'efficienza, come dal paragrafo 6.2 delle stesse dispense."

Per la complessità:

- consideriamo una complessità lineare di analisi di tutti i nodi → $O(n)$
- cerchiamo il minimo e dobbiamo trovare un minimo dipendente dalla cardinalità di tutti gli elementi; *usando una lista*, avremo $n * O(n) = O(n^2)$
- controlliamo gli archi che escono da un solo nodo, con $n * O(n) = O(n^2)$
- la complessità sarebbe $O(n) + O(n^2) + O(n^2) = O(n^2)$
- calcolando invece la complessità ammortizzata (cioè, considerando tutte le iterazioni e nel caso peggiore; se le operazioni costose sono poco frequenti, compenso "ammortizzando" con operazioni poco costose in modo deterministico), consideriamo ogni volta gli archi che escono dall'etichetta ottima; controlliamo *al massimo una volta* tutti gli archi.
  Avremo quindi $O(m), m = |A|$

La complessità finale, usando S e $\bar{S}$ come liste è $O(|N| + |N|^2 + |A|) = O(|N|^2)$

Implementando invece S e $\bar{S}$ come heap, possiamo fare la ricerca di minimo in tempo costante, che ora ci costa almeno $O(n)$, dato che se fosse min-heap, sta nella radice.

- Per creare un heap (*create*), costerà $O(1)$.
- Per l'inserimento (*insert*), dovremo mantenere la proprietà di heap (facendo in modo che il nodo padre sia più grande/più piccolo di tutti i nodi figli e sia bilanciato, quindi con numero minimo di figli), con costo $O(\log(n))$
- Una volta trovato il minimo (*find-min*), dobbiamo anche eliminare il nodo radice e ristabilire tutto lo heap.
- Quindi, eliminare il minimo (*delete-min*) costerà $O(\log(n))$
- L'aggiornamento dell'etichetta (*decrease-key*) costerà $O(\log(n))$

La complessità finale, usando S e $\bar{S}$ come heap è $O(|A| \log(|N|))$

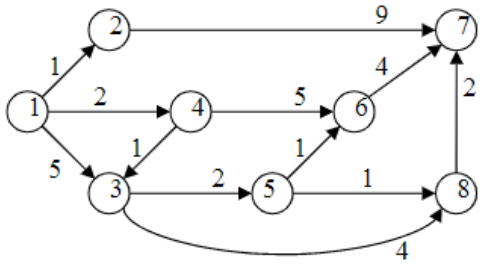
Quindi, avremo $O(n^2)\ vs\ O(mlog(n))$; tuttavia:

- se avessimo a che fare con un grafo completo/denso, meglio implementare con le liste;
- se avessimo a che fare con un grafo sparso, meglio implementare con gli heap.

Se usassimo un Fibonacci heap, si dimostra che la complessità sarà $O(|A| + |N| \log|\mathrm{N}|)$.

Simulazione Esame 2018/2019 – Esercizio 3 Grafi Raccolta "Esercizi vari"

3. Si vogliono determinare i cammini minimi composti da al più 4 archi sul seguente grafo:



- si scelga un algoritmo appropriato e si motivi la scelta;
- si calcolino i cammini minimi con al più quattro archi dal nodo 1 verso tutti gli altri nodi (i passi dell'algoritmo vanno riportati in una tabella e giustificati);
- si ricavi un cammino minimo di al più quattro archi da 1 a 7, descrivendo il procedimento adottato.

a) Nella scelta dell'algoritmo, notiamo che ci viene dato un massimo numero di archi da dover rispettare, pertanto si può applicare solo l'algoritmo di Bellman – Ford, l'unico che dà la possibilità di calcolo dei cammini minimi sulla base di un massimo numero di archi. Applicheremo Bellman – Ford fermandoci alla quarta iterazioni, con un numero di archi e iterazioni pari a $k \leq 4$

b) Ora, il calcolo dei cammini minimi, riportati in tabella (controllando ad ogni passo miglioramenti rispetto alla riga precedente e segnando in colonna apposita gli aggiornamenti effettuati; i predecessori sono segnati come semplici pedici senza parentesi):

Iterazione	Nodo 1	Nodo 2	Nodo 3	Nodo 4	Nodo 5	Nodo 6	Nodo 7	Nodo 8	Aggiornamenti
$Inizio$	$0_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	1
$h=1$	$0_\wedge$	1_1	5_1	2_1	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	2,3,4
$h=2$	$0_\wedge$	1_1	3_4	2_1	7_3	7_4	10_2	9_3	3,5,6,7,8
$h=3$	$0_\wedge$	1_1	3_4	2_1	5_3	7_4	10_2	7_3	5,8
$h=4$	$0_\wedge$	1_1	3_4	2_1	5_3	6_5	9_8	6_5	6,7,8

Le etichette di una riga sono ottenute controllando i vincoli duali su tutti gli archi uscenti dai nodi "aggiornati" della riga (iterazione) precedente secondo la $if\ \pi_j > \pi'_i + c_{ij}\ then\ \pi_j = \pi'_i + c_{ij}\ and\ p(j) = i$ dove (i,j) è uno degli archi uscenti da un nodo i aggiornato all'iterazione precedente, π_j è l'etichetta corrente (sulla riga corrente) del nodo j, π'_i è l'etichetta del nodo i all'iterazione (riga) precedente e c_{ij} è il costo dell'arco (i,j).

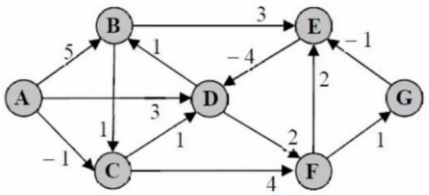
Grazie al fatto di utilizzare l'etichetta del nodo precedente, viene assicurata la scelta del cammino minimo con 4 archi (qua si può fare un qualsiasi esempio numerico dove si va a scegliere, prendendo ad esempio l'ultima iterazione, qui $h = 4$ un cammino che ha costo migliore considerando l'etichetta precedente e non quella corrente, qui $h = 5$, dimostrando la validità di quanto fatto).

c) Un cammino minimo con al massimo 4 archi da 1 a 7 viene fatta seguendo la catena dei predecessori, quindi, partendo dal nodo 7 (ultimo nodo in generale) con $h = 4$, si considera quindi:

$$7 \Leftarrow 8 \Leftarrow 3 \Leftarrow 4 \Leftarrow 1$$

Questo equivale al percorso $1 \Rightarrow 4 \Rightarrow 3 \Rightarrow 8 \Rightarrow 7$, con costo 9.

3. Nel seguente grafo, calcolare i cammini minimi dal nodo A verso **tutti** gli altri nodi.

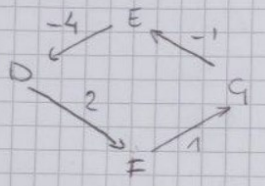


a. si scelga l'algoritmo da utilizzare e si motivi la scelta;
b. si applichi l'algoritmo scelto (riportare e **giustificare** i passi dell'algoritmo in una tabella);
c. si utilizzi la tabella del punto b per riportare, se possibile, l'albero e il grafo dei cammini minimi oppure, se esiste, un ciclo di costo negativo (descrivere il procedimento);
d. è possibile, con l'algoritmo scelto, ottenere un cammino minimo da A a E con al più 5 archi? Se sì, qual è? come si ottiene?

3.

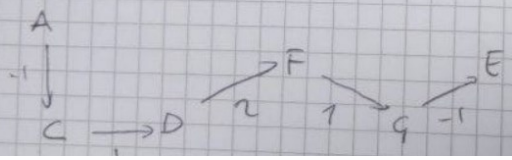
t	A	B	C	D	E	F	G	Flag	Agg
0	0_A	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	true	A
1	0_A	5_A	-1_A	3_A	$+\infty$	$+\infty$	$+\infty$	true	B, C, D
2	0_A	4_D	-1_A	0_C	8_B	3_C	$+\infty$	true	B, D, E, F
3	0_A	1_D	-1_A	0_C	7_B	2_D	4_F	true	B, ~~E~~, F, G
4	0_A	1_D	-1_A	0_C	4_B	2_D	3_F	true	E, G
5	0_A	1_D	-1_A	0_C	~~2_G~~	2_D	3_F	true	E
6	0_A	1_D	-1_A	~~-2_E~~	2_G	2_D	3_F	true	D

l'ultimo nodo da analizzare è D, percui ripercorrendo all'indietro la tabella trovo



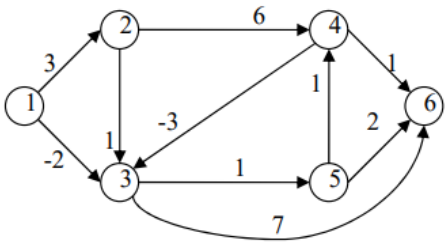
che è un ciclo di costo negativo.

Si è possibile, fermandosi all'iterazione 5 e ripercorrendo a ritroso la tabella si ottiene:



Carrellata di Esercizi da Raccolta “Esercizi vari”

- Si consideri il seguente grafo:



a. si scelga un algoritmo per determinare i cammini minimi dal nodo 1 verso tutti gli altri nodi e si motivi la scelta;
b. si applichi l’algoritmo scelto (riportare e **giustificare** i passi dell’algoritmo in una tabella);
c. L’algoritmo ha individuato un ciclo negativo? Giustificare la risposta.
d. Riportare l’albero e il grafo dei cammini minimi, oppure il ciclo negativo (in ogni caso, si descriva il procedimento utilizzato).

a) In questo grafo, utilizziamo Bellman-Ford, in quanto esistono archi con costi ridotti negativi.

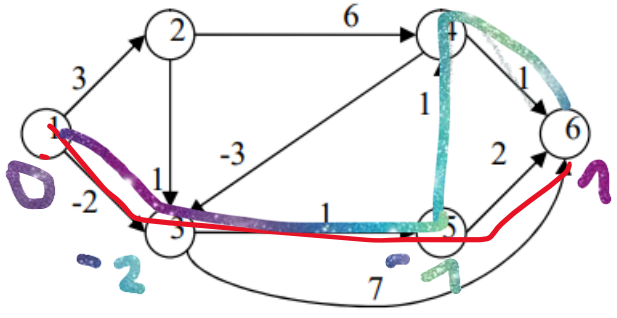
b) Ora, il calcolo dei cammini minimi, riportati in tabella (controllando ad ogni passo miglioramenti rispetto alla riga precedente e segnando in colonna apposita gli aggiornamenti effettuati; i predecessori sono segnati come semplici pedici senza parentesi). Dato che non siamo limitati dai max-hop, andremo a creare la tabella iterando un numero pari di volte al numero di nodi (quindi, facendoli tutti):

Iterazione	Nodo 1	Nodo 2	Nodo 3	Nodo 4	Nodo 5	Nodo 6	Aggiornamenti
$Inizio$	$0_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	1
$h=1$	$0_\wedge$	3_1	-2_1	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	2, 3
$h=2$	$0_\wedge$	3_1	-2_1	9_2	-1_3	5_3	4,5,6
$h=3$	$0_\wedge$	3_1	-2_1	0_5	-1_3	1_5	4, 6
$h=4$	$0_\wedge$	3_1	-2_1	0_5	-1_3	1_5	//
$h=5$	$0_\wedge$	3_1	-2_1	0_5	-1_3	1_5	//
$h=6$	$0_\wedge$	3_1	-2_1	0_5	-1_3	1_5	//

c) L’algoritmo non ha individuato un ciclo negativo in quanto non ha terminato con *flag_aggiornato=true*, cioè non ha aggiornato fino alla fine delle iterazioni.

d) Per individuare l’albero dei cammini minimi, si percorre a ritroso la catena dei predecessori partendo dall’ultima iterazione, quindi, $1-3-5-6$ con costo $0-2-1+1=-2$. Si evidenzia in rosso il cammino minimo.

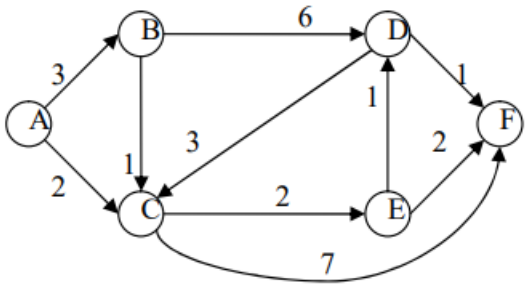
Per individuare il grafo dei cammini minimi, composto da *tutti* i cammini minimi, esamino il grafo e scrivendo le etichette ottime, verifico se esistono altri cammini minimi. In effetti, esistono facendo un altro percorso, passando per 4 e per 6. Si ottiene sempre costo 1.



Legenda:

- Rosso – Albero
- Arcobaleno - Grafo

- Si consideri il seguente grafo:



a. si scelga il miglior algoritmo tra quelli presentati per determinare i cammini minimi dal nodo A verso tutti gli altri nodi e si motivi la scelta;
b. si applichi l'algoritmo scelto (riportare e **giustificare** i passi dell'algoritmo in una tabella);
c. si disegni l'albero e il grafo dei cammini minimi, descrivendo il procedimento usato.

a) Il miglior algoritmo in questo contesto è Dijkstra, in quanto più efficiente e usabile in quanto tutti i costi ridotti sono positivi.

b) Ora, il calcolo dei cammini minimi, riportati in tabella. Si ricorda che:

- $\hat{v}$ rappresenta l'etichetta minima di ogni iterazione
- $\bar{S}$ rappresentano le etichette ancora da fissare
- Il segno * rappresenta l'etichetta fissata
- Il segno – rappresenta l'etichetta controllata ma non aggiornata
- Il segno x rappresenta l'etichetta non controllata perché il nodo è già fissato
- Gli spazi vuoti servono per indicare che non considero più l'etichetta nelle varie iterazioni in quanto fissata
- L'algoritmo termina quando non ci sono più nodi in $\bar{S}$

Iterazione	Nodo A	Nodo B	Nodo C	Nodo D	Nodo E	Nodo F	$\bar{S}$	$\hat{v}$
$Inizio$	$0_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	A,B,C,D,E,F	
$h=1$	*	3_A	2_A	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	B,C,D,E,F	A
$h=2$		-	*	9_B	4_C	9_C	B,D,E,F	C
$h=3$		*		5_E	-	6_E	D,E,F	B
$h=4$				-	*	-	D,F	E
$h=5$				*		-	F	D
$h=6$						*	$\emptyset$	F

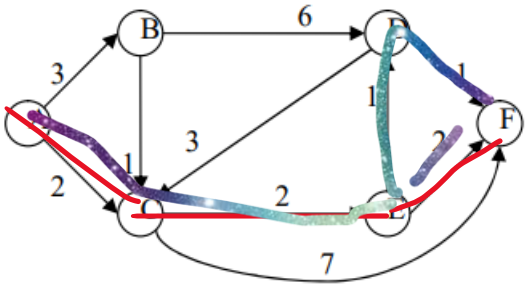
Ad ogni iterazione, percorriamo il grafo e scegliamo il percorso con costo minore. Ad ogni iterazioni, scegliamo e fissiamo un'etichetta che ha costo minore, scremando ad ogni iterazione quelle da controllare e avere sempre in mano il costo minimo. Anche in questo caso, come per gli algoritmi label correcting in caso di convergenza alla soluzione ottima, le etichette calcolate e i relativi puntatori rappresentano, rispettivamente, una soluzione duale ammissibile e i predecessori su dei cammini dall'origine ai diversi nodi. Questo funziona non avendo costi negativi.

c)

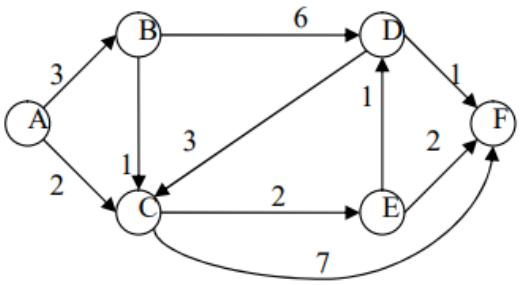
Pertanto, come abbiamo visto per gli algoritmi label correcting in caso di convergenza, è possibile derivare l'albero (risp. il grafo) dei cammini minimi attraverso i puntatori ai predecessori (risp. la verifica della saturazione dei vincoli duali sugli archi).

Concretamente, avremo che:

- In rosso riportiamo l'albero dei cammini minimi
- In arcobaleno riportiamo il grafo (composto come sempre da *tutti* i cammini minimi, quindi fissando le etichette ottime e scegliendo come cammino sia l'albero che tutte le altre etichette con costo $\leq$)



- Si consideri il seguente grafo:



a. si scelga il miglior algoritmo tra quelli presentati per determinare i cammini minimi CON MASSIMO 4 ARCHI dal nodo A verso tutti gli altri nodi e si motivi la scelta;
b. si applichi l'algoritmo scelto (riportare e **giustificare** i passi dell'algoritmo in una tabella);
c. si ricavi un cammino con al più 4 archi da A verso E e un cammino minimo con al più 3 archida A a F: DESCRIVERE IL PROCEDIMENTO.
d. è possibile, ricavare direttamente dalla tabella ottenuta albero e/o grafo dei cammini minimi? Giustificare la risposta.

a) Nella scelta dell'algoritmo, notiamo che ci viene dato un massimo numero di archi da dover rispettare, pertanto si può applicare solo l'algoritmo di Bellman – Ford, l'unico che dà la possibilità di calcolo dei cammini minimi sulla base di un massimo numero di archi. Applicheremo Bellman – Ford fermandoci alla quarta iterazioni, con un numero di archi e iterazioni pari a $k \leq 4$
b) Ora, il calcolo dei cammini minimi, riportati in tabella (controllando ad ogni passo miglioramenti rispetto alla riga precedente e segnando in colonna apposita gli aggiornamenti effettuati; i predecessori sono segnati come semplici pedici senza parentesi):

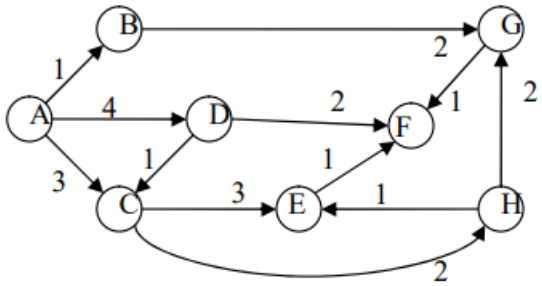
Iterazione	Nodo A	Nodo B	Nodo C	Nodo D	Nodo E	Nodo F	Aggiornamenti
$Inizio$	$0_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	A
$h=1$	$0_\wedge$	3_A	2_A	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	B, C
$h=2$	$0_\wedge$	3_A	2_A	9_B	4_C	9_C	D, E, F
$h=3$	$0_\wedge$	3_A	2_A	5_E	4_C	6_E	D, F
$h=4$	$0_\wedge$	3_A	2_A	5_E	4_C	6_E	F

Le etichette di una riga sono ottenute controllando i vincoli duali su tutti gli archi uscenti dai nodi "aggiornati" della riga (iterazione) precedente secondo la $if\ \pi_j > \pi'_i + c_{ij}\ then\ \pi_j = \pi'_i + c_{ij}\ and\ p(j) = i$ dove (i,j) è uno degli archi uscenti da un nodo i aggiornato all'iterazione precedente, π_j è l'etichetta corrente (sulla riga corrente) del nodo j, π'_i è l'etichetta del nodo i all'iterazione (riga) precedente e c_{ij} è il costo dell'arco (i,j).

c) Un cammino minimo con al più 4 archi da A verso E, seguendo la catena dei predecessori e considerano lo stesso E, viene dato da $A \Rightarrow C \Rightarrow E$ con costo 4
Un cammino minimo con al più 3 archi da A verso F, considerando lo stesso F e seguendo la catena dei predecessori, viene dato da $A \Rightarrow C \Rightarrow E \Rightarrow F$ con costo 10. Usando solo i predecessori, non abbiamo un albero dei cammini minimi.
Per l'individuazione di entrambi i cammini minimi, seguo come detto i predecessori partendo dal nodo indicato e diminuendo h di 1, considerando la diminuzione delle iterazioni ad ogni passo e guardando la riga precedente. Infatti, nel caso si risolva un problema di cammino minimo con un massimo numero di hop, non si può parlare di albero dei cammini minimi.

d) L'albero dei cammini minimi può essere ottenuto dalla tabella seguendo la catena dei predecessori (solo nel caso della prima richiesta), mentre per quanto riguarda il grafo, questo viene dato da tutti e soli i cammini minimi; in questo caso specifico, coincide con il percorso con al più 3 archi, in quanto quello da 4 non può essere considerato dati i motivi enunciati sopra.

- Si consideri il seguente grafo:



  a. si scelga un algoritmo appropriato e si motivi la scelta;
  b. si calcolino i cammini minimi dal nodo A verso tutti gli altri nodi (i passi dell'algoritmo vanno riportati in una tabella e giustificati);
  c. si disegni l'albero e il grafo dei cammini minimi, descrivendo il procedimento usato.

a) Il miglior algoritmo in questo contesto è Dijkstra, in quanto più efficiente e usabile in quanto tutti i costi ridotti sono positivi.

b) Ora, il calcolo dei cammini minimi, riportati in tabella. Si ricorda che:
- $\hat{v}$ rappresenta l'etichetta minima di ogni iterazione
- $\bar{S}$ rappresentano le etichette ancora da fissare
- Il segno * rappresenta l'etichetta fissata
- Il segno – rappresenta l'etichetta controllata ma non aggiornata
- Il segno x rappresenta l'etichetta non controllata perché il nodo è già fissato
- Gli spazi vuoti servono per indicare che non considero più l'etichetta nelle varie iterazioni in quanto fissata
- L'algoritmo termina quando non ci sono più nodi in $\bar{S}$

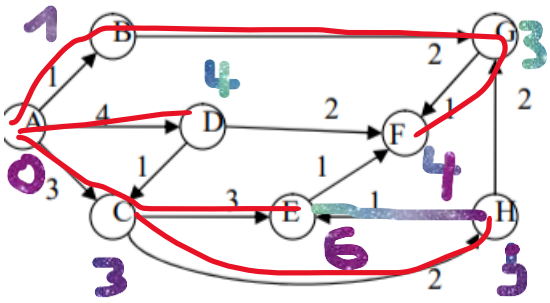
Iterazione	Nodo A	Nodo B	Nodo C	Nodo D	Nodo E	Nodo F	Nodo G	Nodo H	$\bar{S}$	$\hat{v}$
$Inizio$	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A,B,C,D,E,F,G,H	
$h=1$	*	1_A	3_A	4_A					B,C,D,E,F,G,H	A
$h=2$		*					3_B		C,D,E,F,G,H	B
$h=3$			*		6_C			5_C	D,E,F,G,H	C
$h=4$						4_G	*		D,E,F,H	G
$h=5$						*			D,E,H	F
$h=6$			x	*		x			E,H	D
$h=7$					$-$				G	H
$h=8$					*	x			$\emptyset$	G

Ad ogni iterazione, percorriamo il grafo e scegliamo il percorso con costo minore. Ad ogni iterazioni, scegliamo e fissiamo un'etichetta che ha costo minore, scremando ad ogni iterazione quelle da controllare e avere sempre in mano il costo minimo. Anche in questo caso, come per gli algoritmi label correcting in caso di convergenza alla soluzione ottima, le etichette calcolate e i relativi puntatori rappresentano, rispettivamente, una soluzione duale ammissibile e i predecessori su dei cammini dall'origine ai diversi nodi. Questo funziona non avendo costi negativi.

In questo esercizio, si presentavano più etichette con lo stesso costo minimo; ho scelto in tutti i casi l'etichetta con indice inferiore.

c)

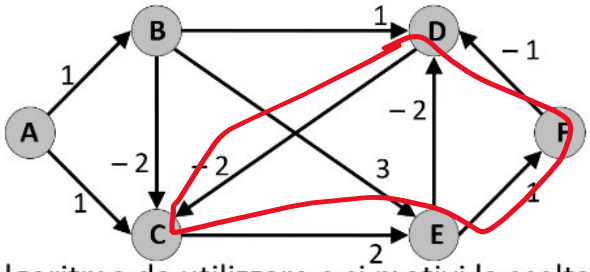
Pertanto, come abbiamo visto per gli algoritmi label correcting in caso di convergenza, è possibile derivare l'albero (risp. il grafo) dei cammini minimi attraverso i puntatori ai predecessori (risp. la verifica della saturazione dei vincoli duali sugli archi).



Concretamente, avremo che:

- In rosso riportiamo l'albero dei cammini minimi
- In arcobaleno riportiamo il grafo (composto come sempre da *tutti* i cammini minimi, quindi fissando le etichette ottime con lo stesso colore e scegliendo come cammino sia l'albero che tutte le altre etichette con costo $\leq$); in questo caso, grafo ed albero sono diversi.

3. Calcolare i cammini minimi **con al più 5 archi** dal nodo A verso **tutti** g
altri nodi.



a) si scelga l'algoritmo da utilizzare e si motivi la scelta;

b) si applichi l'algoritmo scelto (riportare e **giustificare** i passi dell'algoritmo in una tabella);

c) si riporti un cammino minimo con al più 5 archi da A verso C (giustificare)

d) è possibile, basandosi solo sulla tabella del punto b **e senza ulteriori calcoli**, determinare l'esistenza di un ciclo negativo?

a)

Nella scelta dell'algoritmo, notiamo che ci viene dato un massimo numero di archi da dover rispettare, pertanto si può applicare solo l'algoritmo di Bellman – Ford, l'unico che dà la possibilità di calcolo dei cammini minimi sulla base di un massimo numero di archi. Applicheremo Bellman – Ford fermandoci alla quarta iterazioni, con un numero di archi e iterazioni pari a $k \leq 5$

b)

Iterazione	Nodo A	Nodo B	Nodo C	Nodo D	Nodo E	Nodo F	Aggiornamenti
$Inizio$	$0_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	1
$h=1$	$0_\wedge$	1_A	1_A	$+\infty_\wedge$	$+\infty_\wedge$	$+\infty_\wedge$	B, C
$h=2$	$0_\wedge$	1_A	-1_B	2_B	3_C	$+\infty_\wedge$	C, D, E
$h=3$	$0_\wedge$	1_A	1_B	1_E	1_C	4_E	D, E, F
$h=4$	$0_\wedge$	1_A	-1_B	-1_E	1_C	2_E	D, F
$h=5$	$0_\wedge$	1_A	-3_D	-1_E	1_C	2_E	D

c) Un cammino minimo (con al più 5 archi) può essere individuato seguendo la catena dei predecessori

$F \Rightarrow E \Rightarrow C \Rightarrow B \Rightarrow A$ che equivale al percorso con costo $1 + (-2) + 2 + 1 = 2$

A livello di algoritmo, si termina con *flag_aggiornato=true*, pertanto esiste un ciclo negativo, in quanto ad ogni iterazione è stata aggiornata almeno un'etichetta.

d) Basandosi solo sulla tabella e senza ulteriori calcoli, è possibile determinare il ciclo negativo perché siamo arrivati "fino in fondo" con i calcoli.

e) È possibile disegnare albero e grafo dei cammini minimi? No, in quanto abbiamo etichette instabili (no soluzione ammissibile duale e quindi no cammini minimi).

Dimostriamo ad esempio esiste il ciclo $[C, D, E, F]$; come tale, partiamo dal nodo iniziale e seguiamo per i primi nodi quelli dettati dal percorso dei predecessori e poi seguiamo il percorso che vogliamo fare noi, in questo caso ciclo; come si vede, esiste un vertice che era già presente nel cammino e che, idealmente, lo migliora ad n iterazioni.

Risposte con limiti di archi (al più 5 archi)

e) È possibile disegnare albero e grafo dei cammini minimi? No, in quanto abbiamo etichette instabili (no soluzione ammissibile duale e quindi no cammini minimi). Fermandosi all'iterazione 5, vedo che c è stato aggiornato e, seguendo da questo la catena dei predecessori, arrivo a costruire l'albero dei cammini minimi.

Senza fare calcoli, non posso sapere se le etichette si stabilizzeranno e quindi dato che non faccio l'iterazione 6, non riesco a fare il grafo con cammini minimi.