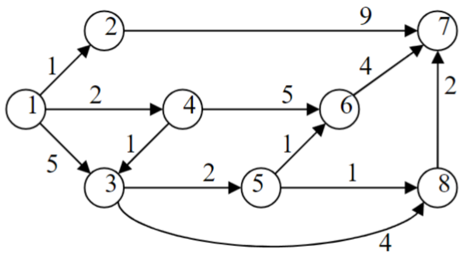
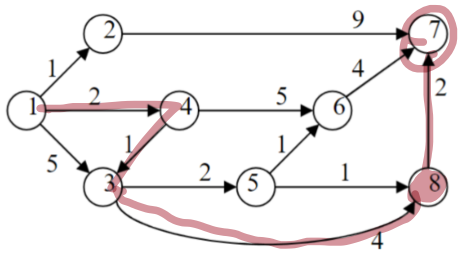
- (*3) Si vogliono determinare i cammini minimi composti da al più 4 archi sul seguente grafo:




BELLMAN-FORD

- si scelga un algoritmo appropriato e si motivi la scelta; → DIJKSTRA
- 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.




| ITER. | A | B | C | D | E | F | G (7) | H |
|---|---|---|---|---|---|---|---|---|
| 0 | $0_1$ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 1 | $0_1$ | $1_A$ | $5_A$ | $2_A$ | ∞ | ∞ | ∞ | ∞ |
| 2 | $0_1$ | $1_A$ | $3_D$ | $2_A$ | $7_C$ | $7_D$ | $10_B$ | $9_C$ |
| 3 | $0_1$ | $1_A$ | $3_D$ | $2_A$ | $5_C$ | $7_D$ | $10_B$ | $7_C$ |
| 4 | $0_1$ | $1_A$ | $3_D$ | $2_A$ | $5_C$ | $6_E$ | $9_H$ | $6_E$ |

"Si fa infine notare che, in presenza del massimo numero di archi, non è possibile parlare di albero dei cammini minimi, come mostrato dal seguente Esempio 11."

WIGLONS SAYS....

Utilizzando i puntatori ottenuti mell'ultima iterazione, si ottiene l'albero in Figura 16, che *non* è un albero di cammini minimi con al massimo 3 archi, visto che il cammino da A a E ne contiene 4.

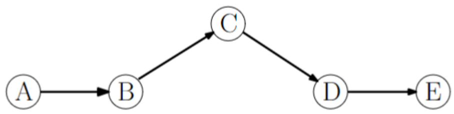



Figure 16: Un albero di cammini per l'Esempio 11.

In ogni caso, i puntatori ai nodi predecessori determinati durante le varie iterazioni possono essere utilizzati per costruire, a ritroso, dei cammini minimi vincolati dal massimo numero di archi dall'origine verso un nodo specifico. Si tratta di partire dal nodo des-

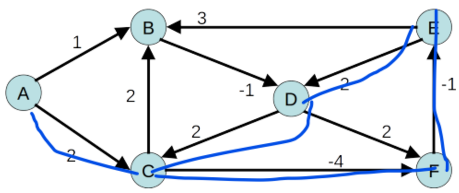
Il grafo dei cammini minimi, invece, include TUTTI i possibili cammini che hanno costo minimo, anche se raggiungono lo stesso nodo di destinazione attraverso percorsi diversi ma di uguale costo totale.

L'albero dei cammini minimi è una struttura che:

1. È definibile solo quando non ci sono vincoli sul numero di archi
2. Rappresenta l'insieme dei cammini minimi dall'origine a tutti gli altri nodi ottenuti seguendo la catena dei predecessori
3. È un sottografo del grafo originale che non contiene cicli

Il grafo dei cammini minimi invece:

1. Include tutti i possibili cammini di costo minimo tra coppie di nodi
2. Può esistere anche con vincoli sul numero di archi
3. È un sovrainsieme dell'albero (quando quest'ultimo esiste)

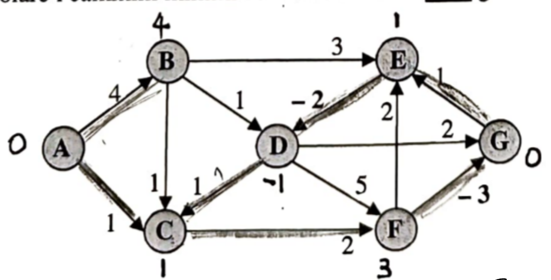



Se salendo ritorno a un nodo già visitato (partendo dai predecessori) e deve succedere che ritorna nel colonna del nodo già preso con i predecessori

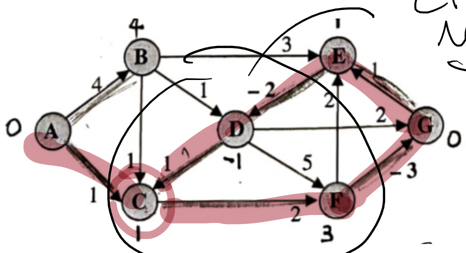
Figure 13: Grafo per l'Esempio 8.

| iterazione | nodo A | nodo B | nodo C | nodo D | nodo E | nodo F | Aggiornati |
|---|---|---|---|---|---|---|---|
| $h = 0$ | $0\ (\wedge)$ | $+\infty\ (\wedge)$ | $+\infty\ (\wedge)$ | $+\infty\ (\wedge)$ | $+\infty\ (\wedge)$ | $+\infty\ (\wedge)$ | $A$ |
| $h = 1$ | $0\ (\wedge)$ | $+1\ (A)$ | $+2\ (A)$ | $+\infty\ (\wedge)$ | $+\infty\ (\wedge)$ | $+\infty\ (\wedge)$ | $B, C$ |
| $h = 2$ | $0\ (\wedge)$ | $+1\ (A)$ | $+2\ (A)$ | $0\ (B)$ | $+\infty\ (\wedge)$ | $-2\ (C)$ | $D, F$ |
| $h = 3$ | $0\ (\wedge)$ | $+1\ (A)$ | $+2\ (A)$ | $0\ (B)$ | $-3\ (F)$ | $-2\ (C)$ | $E$ |
| $h = 4$ | $0\ (\wedge)$ | $0\ (E)$ | $+2\ (A)$ | $-1\ (E)$ | $-3\ (F)$ | $-2\ (C)$ | $B, D$ |
| $h = 5$ | $0\ (\wedge)$ | $0\ (E)$ | $+1\ (D)$ | $-1\ (E)$ | $-3\ (F)$ | $-2\ (C)$ | $C$ |
| $h = 6$ | $0\ (\wedge)$ | $0\ (E)$ | $+1\ (D)$ | $-1\ (E)$ | $-3\ (F)$ | $-3\ (C)$ | $F$ |

**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; → BF
b. si applichi l'algoritmo scelto (riportare e **giustificare** i passi dell'algoritmo in una tabella);
c. con le informazioni disponibili dalla tabella, e descrivendo il procedimento, si riporti l'albero e il grafo dei cammini minimi se possibile, oppure si individui un ciclo di lunghezza negativa.



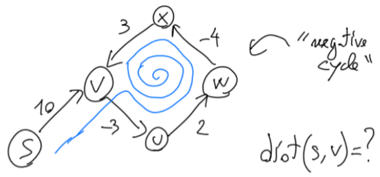
CICLO NEGATIVO

| ITER. | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| ∅ | $0_\Lambda$ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 1 | $0_\Lambda$ | $4_A$ | $1_A$ | ∞ | ∞ | ∞ | ∞ |
| 2 | $0_\Lambda$ | $4_A$ | $1_A$ | $5_B$ | $7_B$ | $3_C$ | ∞ |
| 3 | $0_\Lambda$ | $4_A$ | $1_A$ | $5_B$ | $5_F$ | $3_C$ | $0_F$ |
| 4 | $0_\Lambda$ | $4_A$ | $1_A$ | $5_B$ | $1_G$ | $3_C$ | $0_F$ |
| 5 | $0_\Lambda$ | $4_A$ | $1_A$ | $-1_B$ | $1_G$ | $3_C$ | $0_F$ |

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 6 | $0_\Lambda$ | $4_A$ | $0_D$ | $-1_E$ | $1_G$ | $3_C$ | $0_F$ |
| 7 | $0_\Lambda$ | $4_A$ | $0_D$ | $-1_E$ | $1_G$ | $3_C$ | $0_F$ ] → ∅ |

La ragione è formalmente rigorosa: un ciclo negativo implica che possiamo ridurre indefinitamente il costo di un cammino percorrendo ripetutamente il ciclo. Questo significa che:

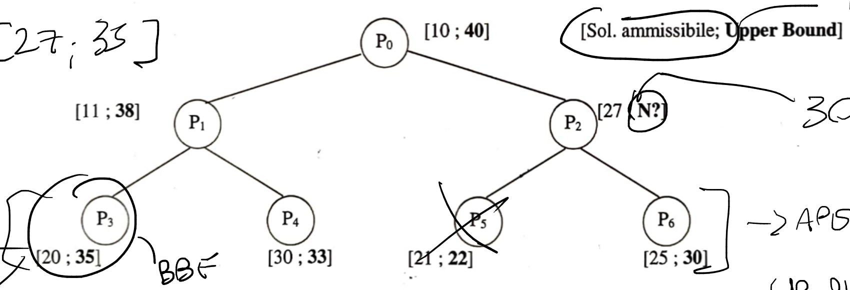
1. Non esistono cammini minimi definiti, poiché ogni cammino può essere migliorato (ridotto nel costo) aggiungendo un altro giro del ciclo negativo
2. Le etichette dei nodi non si stabilizzano, come si può vedere nella tabella dell'immagine dove continuano ad essere aggiornate anche all'ultima iterazione
3. L'algoritmo di Bellman-Ford rileva questa situazione quando, dopo n iterazioni (dove n è il numero di nodi), trova ancora etichette che possono essere migliorate




**6.** Si consideri il seguente sviluppo di un albero di Branch and Bound per un problema di ottimizzazione:

INCUMBENT

OPT → [27; 35]




$30 \le N \le 40$

→ APERTO

BBF

UB DIMINUISCONO

Si risponda, dando giustificazione, alle seguenti domande:

a. Perché si può stabilire che, per opportuni valori di N, si tratta di un problema di massimo? Qual è un possibile valore N per l'upper bound del nodo P₂?
b. Qual è l'intervallo più piccolo che contiene sicuramente il valore ottimo della funzione obiettivo?
c. Quali sono i nodi attualmente aperti? È possibile chiudere dei nodi? Se sì, quali e perché?
d. Quale nodo sarà sviluppato per primo in una strategia Best Bound First? — BBF
e. Si supponga che lo sviluppo di cui al punto precedente porti a due nodi figli, di cui uno è relativo ad un insieme di soluzioni vuoto. Si dia un esempio di valori di upper bound e soluzione ammissibile relativi al secondo nodo che consentano di riconoscere subito la soluzione ottima del problema.

NEW OPT [33; 33]

$UB \le 35$; $S.A \ge 27$

$S.A \le 35$

↓

[33 ; 33]

**5.** Si consideri il seguente tableau del simplesso:

ILLIMITATO

$\forall \le 0$

$B = [x_7 \; x_2 \; x_1]$

BLAND

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ | $z$ | $b$ |
|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | $\boxed{-2}$ | −3 | −4 | 0 | 0 | −1 | 10 |
| $x_7$ | 0 | 0 | $\boxed{16}$ | 2 | −1 | 0 | $\boxed{1}$ | 0 | 0 |
| $x_2$ | 0 | 1 | 60 | 8 | 30 | 1 | 0 | 0 | 60 |
| $x_1$ | $\boxed{1}$ | 0 | −2 | −1 | 40 | 23 | 0 | 0 | 80 |

PIVOT

PRIM DUALE

S S

FO

DUALITÀ FORTE

[PRIM. ILLIM.] [DUALE INAMM.]

DUALITÀ DEBOLE

Si dica, **senza eseguire operazioni di pivot** e fornendo una **giustificazione teorica delle risposte**:

a. riusciamo a individuare una soluzione di base corrispondente? Quale? È ottima? → COSTI RIDOTTI NEGATIVI
b. perché non è consentita l'operazione di pivot sull'elemento evidenziato nel cerchio (8)?
c. considerando le variabili ordinate per indice crescente, quale sarà il cambio base secondo le regole del simplesso e applicando la regola di Bland? $x_7$ ESCE, $x_3$ ENTRA
d. su quali elementi è possibile effettuare il pivot secondo le regole del simplesso (indipendentemente da regole anticiclo)? $x_3, x_4, x_5$ e RAPP. MINIMI DI CIASCUNO
e. Per ciascuna delle operazioni di pivot individuate, qual è il relativo valore della funzione obiettivo che si otterrebbe se si eseguisse tale operazione?

$$(-1)\cdot 10 + (-2)\cdot\left(\frac{0}{16}\right) = -10$$ (E COSÌ PER TUTTE)