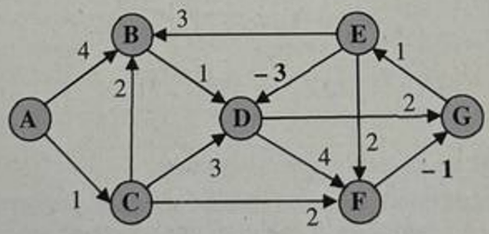
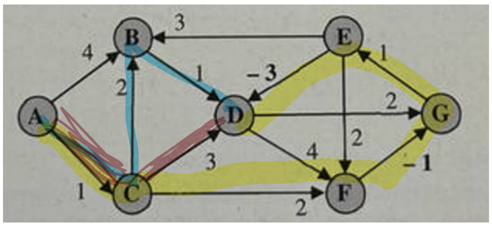
3. Nel seguente grafo, calcolare i cammini minimi dal nodo A verso **tutti** gli altri nodi.



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.
d. si riporti il cammino minimo da A a D, il cammino da A a D con al più 4 archi e con al più 5 archi.



ALBERO

ITER.	A	B	C	D	E	F	G	AGGIORNATI
0	0_Λ	∞	∞	∞	∞	∞	∞	A
1	0_Λ	4_A	1_A	∞	∞	∞	∞	B, C
2	0_Λ	3_C	1_A	4_C	∞	3_C	∞	B, D, F
3	0_Λ	3_C	1_A	4_C	∞	3_C	2_F	G
4	0_Λ	3_C	1_A	4_C	3_G	3_C	2_F	E
5	0_Λ	3_C	1_A	0_E	3_G	3_C	2_F	D
6	0_Λ	3_C	1_A	0_E	3_G	3_C	2_F	$\emptyset$
7	0_Λ	3_C	1_A	0_E	3_G	3_C	2_F	$\emptyset$

La tabella riporta al riga 0 di inizializzazione e una riga per ogni iterazione. All'iterazione h si controllano gli archi (i,j) uscenti da ciascun nodo i nella colonna *Aggiornati* alla riga h–1, e si aggiornano i costi e i predecessori del nodo j all'iterazione h qualora l'etichetta del nodo i all'iterazione h–1 più il costo dell'arco (i,j) sia strettamente minore dell'etichetta corrente del nodo j.

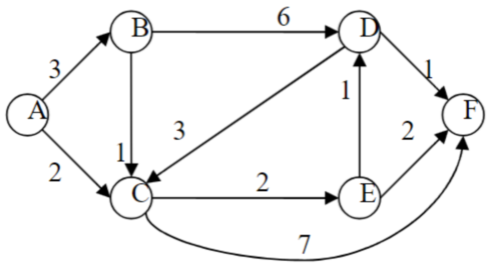
L'algoritmo si ferma qualora la lista dei nodi aggiornati sia vuota (convergenza delle etichette ai costi dei cammini minimi da A verso gli altri nodi) o, come in questo caso, venga completata l'iterazione con h uguale al numero di nodi avendo dei nodi aggiornati (presenza di un ciclo negativo).

- Si consideri il seguente grafo:

DIJKSTRA

→ PIÙ EFFICIENTE

→ PESI > 0



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;

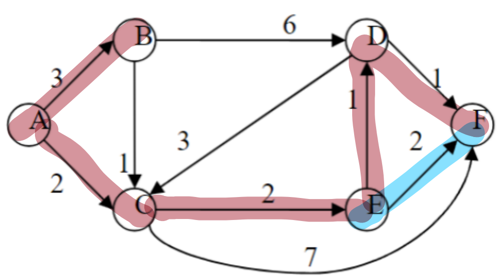
b. si applichi l'algoritmo scelto (riportare e **giustificare** i passi dell'algoritmo in una tabella);

c. si disegni l'albero e il grafo dei cammini minimi, descrivendo il procedimento usato.

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 $\bar{S}$

ITER.	A	B	C	D	E	F	$\bar{S}$	$\hat{v}$
0	$0_\wedge$	∞	∞	∞	∞	∞	A, B, C, D, E, F	∅
1	*	3_A	2_A	∞	∞	9_C	B, C, D, E, F	A
2	X	–	*	∞	4_C	–	B, D, E, F	C
3		*	X	5_E	–	6_E	D, E, F	B
4		X		–	*	–	D, F	E
5				*	X	–	F	D
6				X		*	∅	F



ALBERO = ROSSO

GRAFO = AZZURRO

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.

6. Si consideri il seguente modello di programmazione lineare relativo a un problema di produzione di un insieme di prodotti K, da realizzare con materie prime nell'insieme J, fornite da fornitori nell'insieme I. Sono definiti i parametri: P_k (prezzo di vendita del prodotto k), C_{ij} (costo unitario della materia prima j presso il fornitore i), F_i (costo fisso per fornirsi dal fornitore i), Q_{jk} (quantità di materia j consumata da un'unità di prodotto k), M_i (limite massimo agli acquisti dal fornitore i), e B (budget disponibile).

$$\max \sum_{k \in K} P_k x_k$$

$$\text{s.t.} \quad \sum_{i \in I, j \in J} C_{ij} y_{ij} + \sum_{i \in I} F_i z_i \leq B$$

$$\sum_{i \in I} y_{ij} \geq \sum_{k \in K} Q_{jk} x_k \quad , \forall j \in J$$

$$\sum_{j \in J} y_{ij} \leq M_i z_i \quad , \quad \forall i \in I$$

$$x_k \in \mathbb{Z}_+, y_{ij} \in \mathbb{R}_+, \; z_i \in \{0,1\},$$

$$\forall i \in I, \; j \in J, \; k \in k$$

a. Si traduca nel linguaggio **AMPL** il modello proposto (**file .mod**).

b. Si produca un **file .dat** di esempio per 3 fornitori, 2 materie prime e 3 prodotti.

c. Si scriva uno script di AMPL (**file .run**) che risolve l'istanza specificata e visualizza il valore della funzione obiettivo e delle variabili per una soluzione ottima.

Nell'ordine: file .mod

```
set I; #fornitori
set J; #materie prime
set K; #prodotti
param P{K}; #prezzo di vendita
param C{I,J}; #costo unitario materia prima
param F{I}; #costo fisso
param Q{J,K}; #quantità materia consumata
param M{I}; #limite massimo acquisti fornitore
param B; #budget
var x{K} integer >=0;
var y{I,J} >=0;
var z{I} binary;

maximize ricavo: sum{k in K} P[k]*x[k];
s.t. limite{i in I, j in J}: C[i,j]*y[i,j] + sum{i in I} F[i]*z[i] <= B;
s.t. quantita{j in J}: sum{i in I} y[i,j] >= sum{k in K} Q[j,k] * x[k];
s.t. acquisti{i in I}: sum{j in J} y[i,j] <= M[i] * z[i];
```

File .dat

```
#3 fornitori, 2 materie prime, 3 prodotti
set I := forn1 forn2 forn3;
set J := mat1 mat2;
set K := prod1 prod2 prod3;

param P := prod1 10 prod2 20 prod3 30;
param F := forn1 20 forn2 30 forn3 40;
param M := forn1 100 forn2 200 forn3 300;
param B := 5000;

param C (tr):   forn1    forn2    forn3 :=
mat1            10       20       30
mat2            40       50       60
;

param Q (tr):   mat1     mat2  :=
prod1           5        10
prod2           15       20
prod3           25       30
;
```

TRASPOSTO

I

NORMALMENTE

C[I,J] J

C: → J

↓

I

J

Q[J,K]

K

File .run:

```
reset;
option solver cplex;
model test.mod;
data file.dat;
solve;
 display ricavo, x, y, z;
```

Un mulino produce due tipi di semola normale e integrale a partire da tre tipi di granaglie: A, B e C. Per produrre un quintale di semola normale, sono necessari 0.5 quintali di granaglia A, 0.4 di granaglia B e 0.3 di granaglia C; per un quintale di semola integrale, sono necessari 0.3 quintali di granaglia A, 0.7 di B e 0.4 di C. Il mulino si serve da tre fornitori. Ciascun fornitore mette a disposizione un lotto di acquisto, le cui caratteristiche sono riportate nella seguente tabella:

$(0.5 + 0.4 + 0.3) \leq 3 + 5 + 8$

$S = \{1, 2\}$

N, I

Lotto	Granaglia A	Granaglia B	Granaglia C	Costo	% impurità
1	3 q	5 q	8 q	100 €	1.0 %
2	4 q	9 q	3 q	140 €	2.0 %
3	7 q	2 q	2 q	120 €	1.5 %

Il mulino dispone di 10 000 € per approvvigionarsi di granaglie e vuole massimizzare il numero di quintali di semola prodotta complessivamente, considerando che:

- si possono acquistare al massimo 5 unità di lotto 3;
- la semola normale deve essere almeno il doppio della semola integrale e non più del quadruplo;
- le granaglie del lotto 1 e del lotto 2 sono incompatibili e pertanto non possono essere contemporaneamente acquistate;
- l'impurità media delle scorte di granaglia di tipo A deve essere inferiore allo 1.6%.

$100 x_{A1} + 140 x_{A2} + 120 x_{A3} \leq 10000$

x_i = QUINTALI DI GRANAGLIA DI TIPO $I \in \{A, B, C\}$

$\max \; x_A + x_B + x_C$ (?)

→ QUASI!

NORMALE → $0.5 x_A + 0.4 x_B + 0.3 x_C$

INTEGRALE → $0.3 x_A + 0.7 x_B + 0.4 x_C$

x_{IS} → $I = \{A, B, C\}$

$S = \{1, 2\}$

NORMALE, INTEGRALE

$\max \; (0.5 x_{A1} + 0.3 x_{A2} \; \ldots)$ ...

di quintali di semola prodotta complessivamente, considerando

- si possono acquistare al massimo 5 unità di lotto 3;

$K = \{1, 2, 3\}$ → y_{KI} ⟷ $(y_{3A} + y_{3B} + y_{3C}) \leq 5$

$I = \{A, B, C\}$

- la semola normale deve essere almeno il doppio della semola integrale e non più del quadruplo;

$$\underline{X_{1J} \leq M(Z_{1J})}$$

ATTIVAZIONE

$$\left[2 \cdot X_{1J} \leq X_{2J} \leq 4 \cdot X_{1J} \right]$$

- le granaglie del lotto 1 e del lotto 2 sono incompatibili e pertanto non possono essere contemporaneamente acquistate;

$$Y_{KJ} \rightarrow B_{KJ}$$

$$\rightarrow Y_{1J} \leq 1 - Y_{2J}$$

$$\equiv$$

$$Y_{KJ} \leq M(B_{KJ})$$

$$Y_{1J} + Y_{2J} \leq 1$$

- l'impurità media delle scorte di granaglia di tipo A deve essere inferiore allo 1.6%.

Lotto	Granaglia A	Granaglia B	Granaglia C	Costo	% impurità
1	3 q	5 q	8 q	100 €	1.0 %
2	4 q	9 q	3 q	140 €	2.0 %
3	7 q	2 q	2 q	120 €	1.5 %

$$\left[\frac{0.1 X_{A1} + 0.2 X_{A2} + 0.15 X_{A3}}{3} \leq 0.16 \right]$$

F.O

$1.5\, X_{MB} + 2.5\, X_{MA} + 1.5\, X_{SA} + 2.5\, X_{MR} + 1.5\, X_{SR}$

Es. 1) Si vogliono usare avanzi stoffa per produrre mascherine, vedi tabella:

Tipo	colore	misura	prezzo €	quantità min / qtà max
1	bianco	M	1,5	$15 \le X_{MB} \le 40$
2	azzurro	M	2,5	$10 \le X_{MA} \le 30$
3	azzurro	S	1,5	$5 \le \dots \le 15$
4	rosa	M	2,5	$10 \le \dots \le 30$
5	rosa	S	1,5	$5 \le \dots \le 15$

7 pt.

Ogni mascherina ha 1 strato del colore indicato + secondo strato bianco.
La qtà di stoffa disp. è equivalente a 100 strati di misura M per mascherine bianche, 40 mascherine M rosa e 40 masch. M azzurre.
Le mascherine S consumano la metà della stoffa rispetto alle M.
Scrivere modello PL che massimizzi i ricavi tenendo conto che:
- è stabilita una quantità minima e massima di mascherine (vedi tabella)
- produrre mascherine di minimo 2 e massimo 4 tipi
- la differenza tra il numero complessivo di mascherine rosa e azzurre deve essere massimo 7 $[(X_{MA} + X_{SA}) - (X_{MR} + X_{SR})] \le 7$

100 MB / 40 MR / 40 MA

$|Z| - Z^+$

$X_{MB} \le 100, \quad X_{MR} \le 40, \quad X_{MA} \le 40$

$$Z_{ij} \begin{cases} 1 & \text{se } X_{ij} \\ 0 & \text{no} \end{cases}$$

$$2 \le Z_{ij} \le 4$$

$$\Updownarrow$$

$$X_{ij} \le M(Z_{ij})$$ → ATTIVAZIONE

DOMINI:

$X_{ij} \in R_+, \quad Z_{ij} \in \{0,1\}, \quad \dots\dots$