Esame di Ricerca Operativa - 11 febbraio 2015
Facoltà di Scienze MM.FF.NN. - Verona
- CORREZIONE -

Problema 1 (5 punti):

La Coloraben mira ad affermarsi nella vendita di tinte e smalti. La disponibilità di vernici base in magazzino, per le vendite del prossimo mese, è la seguente: 550 kg di bianco, 150 kg giallo, 90 kg di rosso e 70 kg di verde. Ogni barattolo messo sul mercato contiene 500 grammi di una tinta ottenuta miscelando le quattro vernici base. La Coloraben propone quattro tipi di tinte, descritte di seguito:

prodotto	composizione	profitto (lire/scatola)
tinta 1	solo bianco	260
tinta 2	non piú del 50% di bianco almeno il 10% di rosso almeno il 15% di giallo	400
tinta 3	solo giallo	510
tinta 4	almeno il 30% di giallo almeno il 20% di rosso almeno il 30% di verde	520

Quindi un barattolo di tinta 2 potrebbe ad esempio essere composto al 45% di bianco, 10% di rosso, 20% di giallo, e 25% di verde.

Supponendo che tutto quanto miscelato venga venduto, formulare come PL il problema di massimizzare il profitto della Coloraben.

svolgimento.

Il problema può essere formulato introducendo le seguenti variabili:

- x_{B1} = quantità di bianco (in kg) utilizzata per produrre tinta 1;
- x_{B2} = quantità di bianco (in kg) utilizzata per produrre tinta 1;
- x_{R2} = quantità di rosso (in kg) utilizzata per produrre tinta 2;
- x_{V2} = quantità di verde (in kg) utilizzata per produrre tinta 2;
- x_{G2} = quantità di giallo (in kg) utilizzata per produrre tinta 2;
- x_{G3} = quantità di giallo (in kg) utilizzata per produrre tinta 2;
- x_{B4} = quantità di bianco (in kg) utilizzata per produrre tinta 4;
- x_{R4} = quantità di rosso (in kg) utilizzata per produrre tinta 4;
- x_{V4} = quantità di verde (in kg) utilizzata per produrre tinta 4;
- x_{G4} = quantità di giallo (in kg) utilizzata per produrre tinta 4;
- y_1 = numero di barattoli di tinta 1 prodotti;
- y_2 = numero di barattoli di tinta 2 prodotti;
- y_3 = numero di barattoli di tinta 3 prodotti;
- y_4 = numero di barattoli di tinta 4 prodotti.

Stiamo supponendo per semplicità che le variabili y_i non siano vincolate ad essere intere. L'obiettivo é quello di massimizzare i ricavi sulla vendita dei quattro tipi di confezioni ossia

$$\max R \quad = \quad 260\, y_1 + 400\, y_2 + 510\, y_3 + 520\, y_4\,,$$

nel rispetto dei seguenti vincoli:

vincoli di non negativitá

$$y_1, y_2, y_3, y_4, x_{B1}, x_{B2}, x_{R2}, x_{V2}, x_{G2}, x_{G3}, x_{B4}, x_{R4}, x_{V4}, x_{G4} \geq 0.$$

vincoli sulla composizione

$$\begin{aligned}
x_{B1} &= 0,5\, y_1 \\
x_{B2} + x_{R2} + x_{G2} + x_{V2} &= 0,5\, y_2 \\
x_{G3} &= 0,5\, y_3 \\
x_{B4} + x_{R4} + x_{G4} + x_{V4} &= 0,5\, y_4 \\
x_{B2} &\leq 0,25\, y_2 \\
x_{R2} &\geq 0,05\, y_2 \\
x_{G2} &\geq 0,075\, y_2 \\
x_{R4} &\geq 0,1\, y_4 \\
x_{R4} &\geq 0,15\, y_4 \\
x_{G4} &\geq 0,15\, y_4
\end{aligned}$$

disponibilitá di materie prime

$$\begin{aligned}
x_{B1} + x_{B2} + x_{B4} &\leq 550 \\
x_{G2} + x_{G3} + x_{G4} &\leq 150 \\
x_{R2} + x_{R4} &\leq 90 \\
x_{V2} + x_{V4} &\leq 70
\end{aligned}$$

Ovviamente i vincoli di non negativitá $y_1, y_2, y_3, y_4 \geq 0$ possono essere omessi. Introducendo il vincolo di interezza per le sole 4 variabili y_1, y_2, y_3 e y_4 otteniamo soluzioni intere ottime che possono essere messe in pratica senza arrotondamenti (con conseguente rischio di perdita di precisione nella soluzione del modello matematico intero).

Problema 2 (1+1+1+1+1+2=8 punti):
Si formuli come un modello di Programmazione Lineare Intera (PLI) il seguente modello classico della Ricerca Operativa.

Knapsack

INPUT: Due numeri naturali n, B ed un insieme di n oggetti descritti ciascuno da una coppia valore/peso, (v_i, p_i) per ogni $i = 1, \ldots, n$.

OUTPUT: Trovare un sottoinsieme S degli oggetti assegnati in input, a somma dei pesi non eccedente il budget assegnato B, e massimizzando il valore totale raccolto.

A(**(1pt)**) Fornire una formulazione di PLI per KNAPSACK.

B(**(1pt)**) L'esistenza della formulazione di PLI di cui al punto precedente, ti consente di dedurre quali delle seguenti affermazioni? (Specificando il perchè): (1) la PLI è NP-hard in senso forte; (2) la PLI è NP-hard in senso debole; (3) esiste un algoritmo pseudo-polinomiale per la PLI; (4) esiste un algoritmo polinomiale per KNAPSACK; (5) esiste un algoritmo pseudo-polinomiale per KNAPSACK.

C(**(1pt)**) quali affermazioni tra (1-5) sono note valere pur non essendo di per se deducibili dalla mappatura di KNAPSACK in PLI da te realizzata?

D(**(1pt)**) Si consideri il rilassamento di Programmazione Lineare (PL) ottenuto rimuovendo i vincoli di interezza dalla formulazione di PLI sopra. Fornire un esempio di istanza dove l'unica soluzione ottima di questo rilassamento è frazionaria.

E(**(1pt)**) Fornire un esempio di istanza dove una soluzione ottima di questo rilassamento è frazionaria ed un'altra è intera.

F(**(1pt)**) Puoi fornire un esempio di istanza dove una soluzione ottima di questo rilassamento ha almeno due componenti frazionarie? Perchè?

G(**(2pt)**) Puoi fornire un esempio di istanza dove ogni soluzione ottima di questo rilassamento ha almeno due componenti frazionarie? Perchè?

svolgimento.

A(**(1pt)**) Introduciamo una variabile booleana x_i per ogni $i = 1, \ldots, n$. Nelle nostre intenzioni, $x_i = 1$ significa che l'oggetto i-esimo viene preso nella soluzione S. Se $x_i = 0$ l'oggetto non viene utilizzato. Si perviene alla seguente formulazione di PLI.

$$\max \ \sum_{i=1}^{n} v_i x_i$$
$$\begin{cases} \sum_{i=1}^{n} p_i x_i & \leq \ B \\ x_i & \leq \ 1 \text{ per ogni } i = 1, \ldots, n \\ x_i & \geq \ 0 \text{ per ogni } i = 1, \ldots, n \\ x_i & \quad \text{intero per ogni } i = 1, \ldots, n \end{cases}$$

B(**(1pt)**) Poichè KNAPSACK è noto essere NP-hard in senso debole, e lo abbiamo mappato nella PLI, allora anche la PLI è NP-hard in senso debole. Con la nostra mappatura abbiamo quindi riscontrato la veridicità della (2).

C(**(1pt)**) la (1) e la (5) sono anche esse vere.

D(**(1pt)**) $n = 1$, $B = 1$, $(v_1, p_1) = (1, 2)$. Unica soluzione ottima: $x_1 = \frac{1}{2}$.

E(**(1pt)**) $n = 3$, $B = 2$, $(v_1, p_1) = (4, 4)$, $(v_2, p_2) = (2, 2)$. Soluzione ottima frazionaria: $x_1 = \frac{1}{2}$, $x_2 = 0$. Soluzione ottima intera: $x_1 = 0$, $x_2 = 1$.

F(**(1pt)**) $n = 2$, $B = 1$, $(v_1, p_1) = (v_2, p_2) = (1, 1)$. Soluzione ottima con due componenti frazionarie: $x_1 = x_2 = \frac{1}{2}$.

G(**(2pt)**) Il problema di PL è in forma standard, quindi ha almeno una soluzione ottima di base. Esso presenta $2n$ vincoli di box ($0 \leq x_i \leq 1$, $i = 1, \ldots, n$) ed un solo vincolo tecnico (ed istanza-specifico) oltre questi. In una soluzione ottima di base il numero di componenti frazionarie non può eccedere il numero di vincoli non di box, in questo caso 1. Questo perchè una soluzione di base è caratterizzata dai vincoli tight per essa. Ogni vincolo di box congela una variabile ad un valore intero, ed in un certo senso (pensiero induttivo) la toglie dal tavolo. Volendo chiarire ulteriormente: certamente il vincolo istanza-specifico sarà tight per ogni soluzione ottima (tutti i pesi e tutti i valori sono positivi, e la somma dei pesi eccede B). Quindi è possibile che una delle variabili, ove non placcata da uno dei suoi due vincoli di box, venga placcata ad un valore frazionario da questo vincolo. In pratica, ogni soluzione di base sarà pertanto o intera o con una sola componente frazionaria. Il secondo caso si presenterè se e solo se quella soluzione di base è non degenere. (Perchè?) Ritornando a noi: visto che il teorema fondamentale della programmazione lineare ci consente di dedurre l'esistenza di una soluzione ottima di base per questo problema in forma standard, abbiamo almeno una soluzione ottima con al più una componente frazionaria. Potrò assegnare anche oltre i due punti su questo esercizio in base alla qualità e chiarezza delle argomentazioni.

Problema 3 (6 punti):

Sia $B = 36$ la capacità del mio zaino. Si supponga di voler trasportare un sottoinsieme dei seguenti elementi a massima somma dei valori, soggetti al vincolo che la somma dei pesi non ecceda B.

nome	A	B	C	D	E	F	G	H	I	L	M	N
peso	2	13	14	6	13	3	11	16	4	46	41	44
valore	11	63	60	33	30	13	60	66	20	66	60	20

3.1(1pt) quanto vale la somma massima dei valori di elementi trasportabili (con somma dei pesi al più $B = 36$)? Quali elementi devo prendere?

3.2 (1pt) e nel caso $B = 33$?

3.3 (1pt) e nel caso $B = 28$?

3.4 (1pt) e nel caso $B = 26$?

3.5 (2pt) e se l'oggetto G non fosse più disponibile, quale sarebbe allora la soluzione ottima per $B = 26, 28, 33, 36$?

svolgimento. Per lo svolgimento di questo esercizio di PD si segua la solita traccia: un eventuale preprocessing per eliminare oggetti che non possano far parte di una soluzione ottima in base as un semplice criterio euristico (in questo caso gli ultimi 3 oggetti hanno un peso eccessivo, quindi di fatto non possono far parte di alcuna soluzione) seguito da una programmazione dinamica. La tabella di programmazione prende avvio con una riga tutta a 0, dove si assume che nessuno degli oggetti sia disponibile, e quindi, riga dopo riga, aggiunge un nuovo oggetto dell'istanza in input a quelli considerati disponibili. Ciascuna riga prevede $36 + 1$ colonne, labellate con le possibili capacità intere dello zaino, da 0 a 36. Per ognuna di queste capacità $B' \leq B$, la riga specifica il valore della soluzione ottima per uno zaino di capacità B' ed ove gli oggeti disponibili siano quelli previsti dalla riga. Se avremo l'accortezza che l'aggiunta dell'oggetto G all'insieme (monotonicamente crescente) di oggetti disponibili avvenga solo all'ultima riga, ci risparmieremo di ricalcolare tutta la tabella per rispondere all'ultima domanda. Riportiamo solamente i risultati finali.

Con oggetto G disponibile:

B	max val	peso	quali prendere
36	187 = 60+20+33+11+63	36 = 11+4+6+2+13	G,I,D,A,B
33	169 = 60+33+13+63	33 = 11+6+3+13	G,D,F,B
28	143 = 60+20+63	28 = 11+4+13	G,I,B
26	137 = 60+20+33+11+13	26 = 11+4+6+2+3	G,I,D,A,F

Senza oggetto G:

B	max val	peso	quali prendere
36	169 = 13+33+63+60	36 = 3+6+13+14	F,D,B,C
33	156 = 33+63+60	33 = 6+13+14	D,B,C
28	140 = 11+13+20+33+63	28 = 2+3+4+6+13	A,F,I,D,B
26	129 = 13+20+33+63	26 = 3+4+6+13	F,I,D,B

Problema 4 (4 punti):
Con riferimento ai **Problemi 2,3**. Nel Problema 2 si era fatto riferimento ad un modello di PLI ed al suo rilassamento (problema di PL) per il KNAPSACK. Nel Problema 3 si è fornita un'istanza di KNAPSACK. Chiediamo qui l'applicazione di quei modelli a questa istanza specifica.

4.1 (1pt) scrivere il modello di PLI ed il suo rilassamento di PL per questa istanza.

4.2 (3pt) fornire soluzione ottima di base per il problema di PL.

svolgimento.

4.1 (1pt) scrivere il modello di PLI ed il suo rilassamento di PL per questa istanza.

Applicando il modello di cui al punto (a) del Problema 2, all'istanza offerta nel Problema 3, otteniamo:

$$\max\ 11x_A + 63x_B + 60x_C + 33x_D + 30x_E + 13x_F + 66x_G + 60x_H + 20x_I + 66x_L + 60x_M + 20x_N$$
$$\begin{cases} 2x_A + 13x_B + 14x_C + 6x_D + 13x_E + 3x_F + 16x_G + 11x_H + 4x_I + 46x_L + 41x_M + 44x_N \ \leq 36 \\ x_i \ \leq 1 \text{ per ogni } i = 1, \dots, n \\ x_i \ \geq 0 \text{ per ogni } i = 1, \dots, n \\ x_i \text{ intero per ogni } i = 1, \dots, n \end{cases}$$

4.2 (1pt) fornire soluzione ottima di base per il problema di PL.

Se non ci sono vincoli di interezza, ossia se posso prendere frazioni arbitrarie dei vari oggetti, mi converrà indugiare su quegli oggetti per cui il rapporto $\frac{v}{p}$, chiamiamolo *profitability*, é massimo. Di fatto, mi rendo conto che a questo punto un approccio greedy trova una soluzione ottima:

1. ordina gli oggetti per profitability non crescente;
2. considera gli oggetti in tale ordine, e di ogni oggetto prendine la massima frazione (≤ 1) che fitti nello zaino residuo.

In pratica, possiamo terminare la scansione degli oggetti ordinati non appena lo zaino è pieno. L'ultimo oggetto considerato è il solo oggetto del quale può essere stata presa una frazione x non-intera $0 < x < 1$. In questo caso la soluzion prodotta ha una componente frazionaria, altrimenti è tutta intera. In ogni caso è una soluzione ottima di base.

Computo la profitability:

nome	A	B	C	D	E	F	G	H	I	L	M	N
peso	2	13	14	6	13	3	11	16	4	46	41	44
valore	11	63	60	33	30	13	60	66	20	66	60	20
prof.	5.5	4.846	4.285	5.5	2.307	4.333	5.454	4.125	5	1.434	1.463	0.454

Riordino le colonne e poi computo le frazioni x da sinistra verso destra, tenendo aggiornata la capacità residua dello zaino:

nome	A	D	G	I	B	F	C	H	E	M	L	N
peso	2	6	11	4	13	3	14	16	13	41	46	44
valore	11	33	60	20	63	13	60	66	30	60	66	20
prof.	5.5	5.5	5.454	5	4.846	4.333	4.285	4.125	2.307	1.463	1.434	0.454
x	1	1	1	1	1	0	0	0	0	0	0	0
B resid.	34	28	17	13	0	0	0	0	0	0	0	0

In realtà la soluzione ottima e di base ricostruita per la formulazione di PL è intera. Di fatto essa corrisponde con la soluzione che avevamo reperito con la programmazione dinamica. Ciò è dovuto al fatto che questa occupava interamente lo zaino!

Problema 5 (7 punti):

Si consideri la seguente sequenza di numeri naturali (la prima riga serve solo ad indicizzarla).

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
65	57	55	50	58	47	36	30	59	39	13	54	33	45	20	18	56	53	61	38	19	51	35	26	52

5.1(1pt) trovare una sottosequenza decrescente che sia la più lunga possibile. Specificare quanto è lunga e fornirla.

5.2(1pt) una sequenza è detta decrescere con un possibile ripensamento (Z-sequenza), se esiste un indice i tale che ciascuno degli elementi della sequenza, esclusi al più il primo e l'i-esimo, è strettamente minore dell'immediatamente precedente nella sequenza. Trovare la più lunga Z-sequenza che sia una sottosequenza della sequenza data. Specificare quanto è lunga e fornirla.

5.3(1pt) trovare la più lunga sottosequenza decrescente che includa l'elemento di valore 59. Specificare quanto è lunga e fornirla.

5.4(1pt) trovare una sottosequenza decrescente che sia la più lunga possibile ma eviti di utilizzare i primi 4 elementi. Specificare quanto è lunga e fornirla.

5.5(1pt) trovare una sottosequenza decrescente che sia la più lunga possibile ma eviti di utilizzare gli elementi dal 13-esimo a 16-esimo. Specificare quanto è lunga e fornirla.

5.6(2pt) fornire un minimo numero di sottosequenze crescenti tali che ogni elemento della sequenza data ricada in almeno una di esse. Specificare quante sono e fornirle.

svolgimento. Dapprima compilo la seguente tabella di programmazione dinamica.

Decrescente

⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒	⇒
9	8	7	6	6	5	4	3	6	4	1	5	3	4	2	1	5	4	4	3	1	3	2	1	1
65	57	55	50	58	47	36	30	59	39	13	54	33	45	20	18	56	53	61	38	19	51	35	26	52
1	2	3	4	2	5	6	7	2	6	8	4	7	6	8	9	3	5	2	7	9	6	8	9	6
⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐	⇐

Decrescente

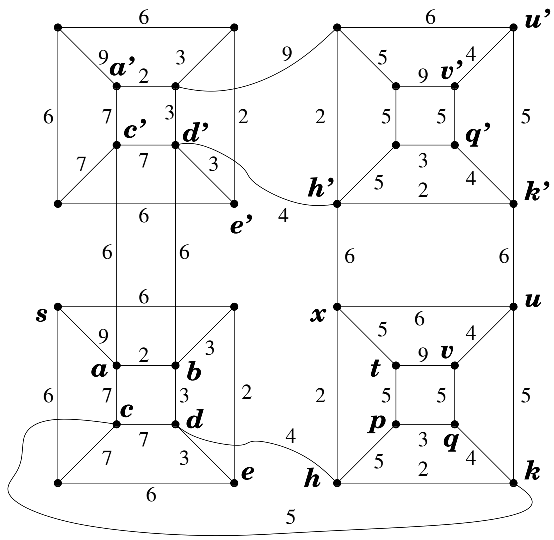
Possiamo ora fornire le seguenti risposte.

tipo sottosequenza	opt val	soluzione ottima
decrescente	9	65, 57, 55, 50, 47, 36, 30, 20, 18
Z-sequenza	14	65, 57, 55, 50, 47, 36, 30, 20, 18, 56, 53, 38, 35, 26
decrescente con 59	7	65, 59, 54, 45, 38, 35, 26
evita i primi 4	6	58, 47, 36, 30, 20, 18
evita da 13-mo a 16-mo	9	65, 57, 55, 50, 47, 39, 38, 35, 26
minima copertura	9	$\underbrace{65}_{1};\ \underbrace{57, 58, 59, 61}_{2};\ \underbrace{55, 56}_{3};\ \underbrace{50, 54}_{4};\ \underbrace{47, 53}_{5};\ \underbrace{36, 39, 45, 51, 52}_{6};\ \underbrace{30, 33, 38}_{7};\ \underbrace{13, 20, 35}_{8};\ \underbrace{18, 19, 26}_{9}$

Dove per il penultimo punto (5) si é osservato dalla tabella di DP (ultima riga) che:

per raccogliere 8 elementi sul solo lato sinistro, l'ultimo vale massimo 13,
per raccogliere 7 elementi sul solo lato sinistro, l'ultimo vale massimo 30,
per raccogliere 6 elementi sul solo lato sinistro, l'ultimo vale massimo 39,
per raccogliere 5 elementi sul solo lato sinistro, l'ultimo vale massimo 47,
per raccogliere 4 elementi sul solo lato sinistro, l'ultimo vale massimo 54,
per raccogliere 3 elementi sul solo lato sinistro, l'ultimo vale massimo 55,
per raccogliere 2 elementi sul solo lato sinistro, l'ultimo vale massimo 59,
per raccogliere 1 elementi sul solo lato sinistro, esso vale massimo 65,

e si sono poi ordinatamente combinate queste osservazioni con analoghe osservazioni concernenti le migliori (non-dominate) scelte relative al come giocarsi il lato destro, sempre come lette dalla tabella (prima riga).
Infine, per l'ultimo punto (6) ho costruito la sequenza crescente i-esima collocando in essa tutti quei numeri della sequenza in input tali che la massima lunghezza di una sequenza decrescente terminante in essi, come calcolata nell'ultima riga della tabella di PD, era precisamente i.



Problema 6 (14 punti):
Si consideri il grafo, con pesi sugli archi, riportato in figura.

6.1.**(1pt)** Dire, certificandolo, se il grafo è planare oppure no. In ogni caso, disegnare il grafo in modo da minimizzare il numero di incroci tra archi.

6.2.**(1pt)** Dire, certificandolo, se il grafo ottenuto da G sostituendo l'arco $h'x$ con un arco $q'x$ è planare oppure no.

6.3.**(1+1pt)** Trovare l'albero dei cammini minimi dal nodo s. Esprimere la famiglia di tali alberi.

6.4.**(2pt)** Trovare un albero ricoprente di peso minimo.

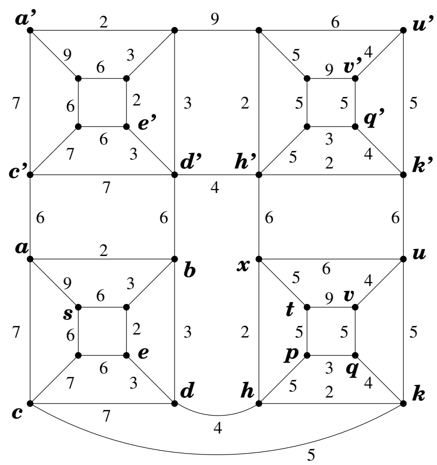
6.5.**(2pt)** Trovare tutti gli alberi ricoprenti di peso minimo. (Dire quanti sono e specificare con precisione come generarli).

6.6.**(2pt)** Trovare un massimo flusso dal nodo s al nodo t.

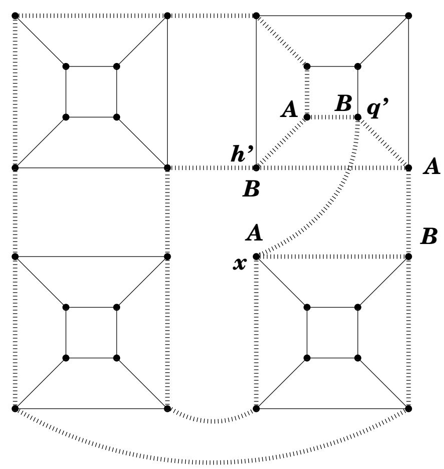
6.7.**(2pt)** Certificare l'ottimalità del flusso massimo dal nodo s al nodo t.

6.8.**(1+1pt)** Fornire (con certificato di ottimalità) il flusso massimo dal nodo s al nodo q.

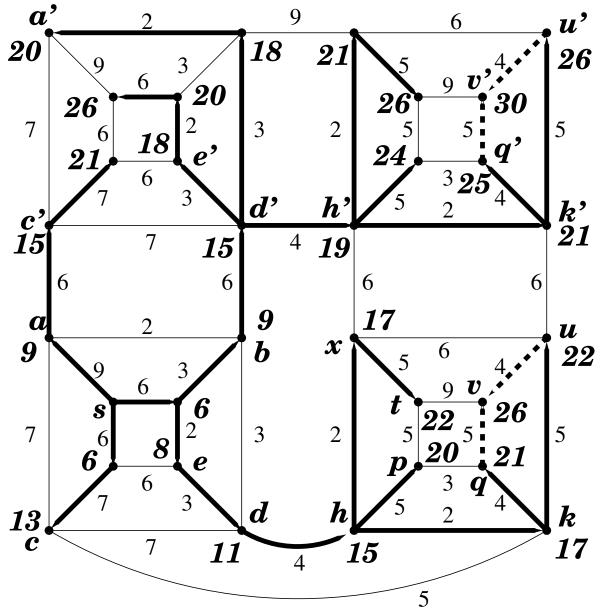
risposte.
Il fatto che G sia planare è messo in evidenza dal planar embedding fornito in figura.



Per la ricerca di alberi ricoprenti di peso minimo e di flussi massimi converrà lavorare sul planar embedding. E forse anche per osservare che il grafo modificato non è planare. Il certificato è la suddivisione del $K_{3,3}$ esibita in figura.

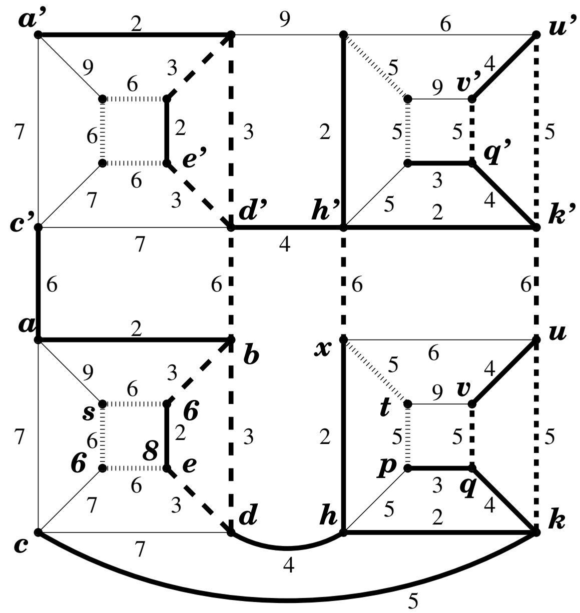


Un albero dei cammini minimi dal nodo s a tutti gli altri nodi del grafo è rappresentato in figura dagli archi in linea spessa (sia tratteggiata che continua).

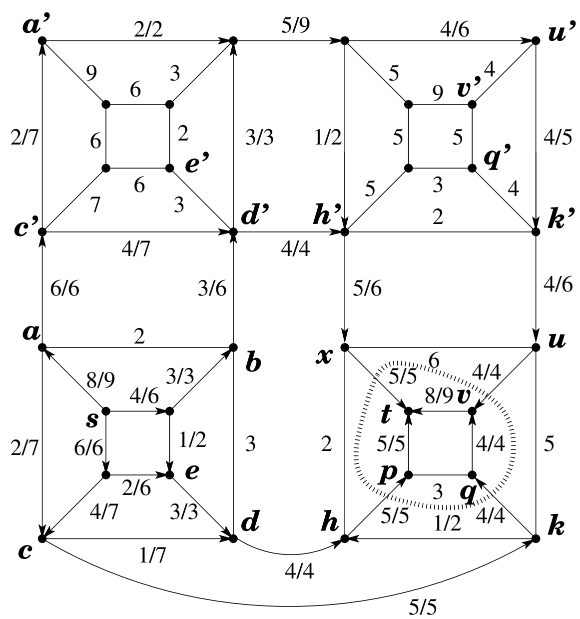


Ovviamente ogni arco del grafo non contenuto nell'albero dei cammini minimi (ossia ogni arco in linea non spessa) può essere rimosso senza allontanare alcun nodo dal nodo s. Inoltre, anche i due archi in linea spessa ma tratteggiata possono essere rimossi poichè sostituibili con altri archi (sempre in linea tratteggiata). Ci sono quindi $2^2 = 4$ alberi di cammini minimi dal nodo s.

La seguente figura, con due quadrati affiancati nella parte alta e due nella parte passa, esprime la famiglia degli alberi ricoprenti di peso minimo. Ci sono $2^4 3^5$ alberi ricoprenti di perso minimo e ciascuno di essi include i 17 archi in linea spessa, più 2 qualsiasi dei 3 archi di peso 3 ed in linea tratteggiata nella parte alta (3 scelte), più 2 qualsiasi dei 3 archi di peso 3 ed in linea tratteggiata nella parte bassa (3 scelte), più 2 qualsiasi dei 3 archi di peso 6 ed in linea sfumata spessa nella parte alta (3 scelte), più 2 qualsiasi dei 3 archi di peso 6 ed in linea sfumata spessa nella parte bassa (3 scelte), più 1 qualsiasi dei 2 archi di peso 5 ed in linea sfumata spessa (più sfumata) nella parte alta (2 scelte), più 1 qualsiasi dei 2 archi di peso 5 ed in linea sfumata spessa (più sfumata) nella parte bassa (2 scelte), più 1 qualsiasi dei 2 archi di peso 5 ed in linea tratteggiata nella parte alta (2 scelte), più 1 qualsiasi dei 2 archi di peso 5 ed in linea tratteggiata nella parte bassa (2 scelte), più 1 qualsiasi dei 3 archi verticali di peso 6 e che collegamo parte bassa e parte alta (3 scelte).



La seguente figura esibisce un flusso massimo (non esibisco tutti i passaggi che ho dovuto compiere per ottenerlo) ed un taglio (minimo) che ne dimostra l'ottimalità.



Il flusso ha valore 18 e satura l'insieme degli archi che attraversano la curva tratteggiata portandosi dal lato di s al lato di t. Questi 4 archi costituiscono pertanto un minimo s,t-taglio, anch'esso di valore 18 e che certifica pertanto l'ottimalità del flusso proposto. (Sia il flusso che il taglio sarebbero stati più immediati a vedersi e verificarsi nel planar embedding. Puoi provare a rappresentarteli lì).

Il massimo flusso da s a q a valore 12 e la stella di q è un taglio che ne certifica l'ottimalitá. A parte questa strozzatura, vi è altrimenti ampio margine nell'inviare flusso da s a q ed evitiamo quindi di fornire descrizione di una tale soluzione ammissibile di valore 12 (cosa che ovviamente voi non potete mai fare: se volete totalizzare i rispettivi punti dovete innanzitutto fronirmi le soluzioni/certificati!).