# Esame di Ricerca Operativa - 19 giugno 2014
# Facoltà di Scienze MM.FF.NN. - Verona
# - CORREZIONE -

**Problema 1 (3+2=5 punti):**

Tre raffinerie producono GPL partendo da oli provenienti da Taranto e Brindisi. Giornalmente, i tre impianti sono in grado di processare le seguenti quantitá di materia grezza (in ettolitri):

| Raffineria 1 | Raffineria 2 | Raffineria 3 |
|---|---|---|
| 150 | 80 | 210 |

A Taranto e Brindisi vi é una disponibilità giornaliera di 130 e di 310 ettolitri rispettivamente. I costi di trasporto sono come indicati nella seguente tabella (Euro/ettolitro)

| | Raffineria 1 | Raffineria 2 | Raffineria 3 |
|---|---|---|---|
| Taranto | 10 | 15 | 20 |
| Brindisi | 8 | 14 | 7 |

Vincoli logistici ed ambientali impongono che non si possa trasportare piú di 50 ettolitri di grezzo da Taranto alla Raffineria 1. In modo similare, non risulta possibile trasportare piú di 50 ettolitri di grezzo da Brindisi alla Raffineria 2. Inoltre, un accordo di cartello impedisce di processare oltre i 400 ettolitri di grezzo nelle 3 raffinerie, complessivamente. Nel rispetto di tale limite, intendiamo spingere al massimo sulla produzione. Si consideri il problema di minimizzare il costo totale di trasporto degli oli grezzi alle raffinerie.

(**(3pt)**) Fornire un modello di PL per tale problema specifico.

(**(2pt)**) Fornire un modello di PL che, piú in generale, si riferisca ad un numero $m$ arbitrario di porti ed ad un numero $n$ arbitrario di impianti di raffinamento.

**svolgimento.**

(**(2pt)**) Le variabili di decisione sono la quantità di olio grezzo da inviare, definite come da seguente tabella.

| | Raffineria 1 | Raffineria 2 | Raffineria 3 |
|---|---|---|---|
| Taranto | $x_{1,1}$ | $x_{1,2}$ | $x_{1,3}$ |
| Brindisi | $x_{2,1}$ | $x_{2,2}$ | $x_{2,3}$ |

Il problema è quindi quello di minimizzare il costo totale del trasporto.

$$\min C = 10\,x_{1,1} + 15\,x_{1,2} + 20\,x_{1,3} + 8\,x_{2,1} + 14\,x_{2,2} + 7\,x_{2,3}\,,$$

nel rispetto dei seguenti vincoli:

**vincoli di non negativitá**

$$x_{1,1}, x_{1,2}, x_{1,3}, x_{2,1}, x_{2,2}, x_{2,3} \geq 0.$$

**vincoli di capacitá delle linee**

$$x_{1,1}, x_{2,2} \leq 50.$$

**vincoli sulla capacitá dei porti**

$$x_{1,1} + x_{1,2} + x_{1,3} \leq 130.$$
$$x_{2,1} + x_{2,2} + x_{2,3} \leq 310.$$

**vincoli sulla potenzialitá delle singole raffinerie**

$$x_{1,1} + x_{2,1} \leq 150.$$
$$x_{1,2} + x_{2,2} \leq 80.$$
$$x_{1,3} + x_{2,3} \leq 210.$$

**vincolo di cartello**

$$x_{1,1} + x_{1,2} + x_{1,3} + x_{2,1} + x_{2,2} + x_{2,3} \leq 400.$$

**vincolo di impiego al massimo della potenzialitá di produzione** Questo era il vincolo piú ostico da gestire (ho dato punteggio pieno a chi ha toppato solo qui e punti extra a chi ha gestito piú o meno bene anche questo). Un modo di gestirlo era osservare che il vincolo di cartello in effetti lavora (ad esempio, la soluzione ammissibile $x_{1,1} = 0$, $x_{1,2} = 80$, $x_{1,3} = 50$, $x_{2,1} = 150$, $x_{2,2} = 0$, $x_{2,3} = 160$, certifica che il livello di produzione puó essere spinto a saturare il vincolo di cartello). Pertanto, il fatto che si intende spingere la produzione al massimo puó in questo caso essere espresso tramite il seguente vincolo.

$$x_{1,1} + x_{1,2} + x_{1,3} + x_{2,1} + x_{2,2} + x_{2,3} \geq 400.$$

Ho pertanto dato 2 punti extra a chi ha imposto il vincolo $x_{1,1}+x_{1,2}+x_{1,3}+x_{2,1}+x_{2,2}+x_{2,3} = 400$. Volendo trattare la questione piú in generale, si sarebbe sostanzialmete dovuto risolvere il problema in due fasi, determinando prima il livello di operativitá massimo per le raffinerie nel loro complesso, e quindi apporre come vincolo quello di non scendere al di sotto di tale livello di operativitá massimo come precedentemente determinato. (Del tutto incidentalmente, questo modello poteva essere visto come un caso particolare del flusso massimo di costo minimo in una rete). Questo modo di operare avrebbe dato le migliori performances computazionali, ma puó lasciare l'amaro in bocca di una soluzione "incompleta" e quindi non essere considerato soddisfacente in certe accezioni o contesti (se questa soluzione deve essere adottata in seno ad una cornice piú generale, ad esempio, assumendo anche di uscire da un contesto di riferimento teorico e di considerare semplicemente una specifica applicazione pratica che ci era stata commessa, la soluzione da noi offerta finirebbe col richiedere un minimo di know-how/insight all'operatore coinvolto nell'utilizzo del modello risultante non in soluzione unica ma in due fasi). Un tale inconveniente, se valutato antipatico o addirittura invalidante nel contesto di riferimento, poteva essere facilmente risolto semplicemente duplicando l'insieme delle variabili in modo da avere sia le $x_{i,j}$ che le $x'_{i,j}$, facendo sottostare entrambe ai vincoli visti sopra, apponendo quindi il vincolo:

$$x_{1,1} + x_{1,2} + x_{1,3} + x_{2,1} + x_{2,2} + x_{2,3} \geq x'_{1,1} + x'_{1,2} + x'_{1,3} + x'_{2,1} + x'_{2,2} + x'_{2,3}\,,$$

e ritoccando la funzione obiettivo come

$$\min 10\, x_{1,1} + 15\, x_{1,2} + 20\, x_{1,3} + 8\, x_{2,1} + 14\, x_{2,2} + 7\, x_{2,3} - M\, x'_{1,1} + M\, x'_{1,2} + M\, x'_{1,3} + M\, x'_{2,1} + M\, x'_{2,2} + M\, x$$

dove $M$ è un qualsiasi valore scelto sufficientemente grande. Una soluzione piú elegante per accoppiare il problema nelle $x$ e quello nelle $x'$ poteva essere il seguente trucco di validitá generale per mettere il valore dell'ottimo del secondo problema (quello nelle $x'$) nel vincolo di cartello del primo (quello nelle $x$). Vorremo cioé poter scrivere direttamente:

$$x_{1,1} + x_{1,2} + x_{1,3} + x_{2,1} + x_{2,2} + x_{2,3} \geq \max\{x'_{1,1} + x'_{1,2} + x'_{1,3} + x'_{2,1} + x'_{2,2} + x'_{2,3}\}.$$

Sarebbe bello poterlo fare vero? Ebbene, il linguaggio della PL é sufficientemente generale da consentire di esprimere anche vincoli come questo entro se stesso e senza esplosioni esponenziali. La cosa puó lasciare un attimo increduli in prima battuta perché é ovvio si possa esprimere vincoli del tipo $x_{1,1} + x_{1,2} + x_{1,3} + x_{2,1} + x_{2,2} + x_{2,3} \geq \min\{x'_{1,1} + x'_{1,2} + x'_{1,3} + x'_{2,1} + x'_{2,2} + x'_{2,3}\}$, di fatto semplicemente scrivendo il vincolo $x_{1,1} + x_{1,2} + x_{1,3} + x_{2,1} + x_{2,2} + x_{2,3} \geq x'_{1,1} + x'_{1,2} + x'_{1,3} + x'_{2,1} + x'_{2,2} + x'_{2,3}$ per raccordare i due problemi, ma in prima battuta sembra difficile gestire un operatore di max in questo contesto. L'idea per rappresentare questo tipo di vincoli é quella di scrivere il duale del problema nelle $x'$ il che ci regalerá un rovesciamento di prospettiva tra max e min e consentira' di combinare felicemente le formulazioni dei due problemi in un'unica formulazione da consegnare scatola chiusa (sghicia boton) all'utenza finale.

(**(2pt)**) Le variabili di decisione sono le $mn$ variabili $x_{i,j}$, $i = 1, \ldots, m$, $j = 1, \ldots, n$ definite da

$x_{i,j}$ = la quantità di olio grazzo da inviare direttamente dal porto $i$ alla raffineria $j$.

Dove si assuma di avere prestabilito una matrice $m \times n$ di costi di trasporto $C$, con

$C_{i,j}$ = costo del trasportare un ettolitro di olio grezzo direttamente dal porto $i$ alla raffineria $j$,

il problema è quello di minimizzare il costo totale del trasporto

$$\min c \quad = \quad \sum_{i=1}^{m} \sum_{j=1}^{n} C_{i,j} x_{i,j} \, ,$$

nel rispetto dei seguenti vincoli:

**vincoli di non negativitá**

$$x_{i,j} \geq 0 \text{ per ogni } i = 1, \ldots, m, \ j = 1, \ldots, n.$$

**vincoli di capacitá sulle linee di trasporto**

$$x_{i,j} \leq UB_{i,j} \text{ per ogni } i = 1, \ldots, m, \ j = 1, \ldots, n,$$

dove $UB_{i,j}$ é un limite superiore al numero di ettolitri di olio grezzo direttamente trasportabile dal porto $i$ alla raffineria $j$.

**vincoli sulla capacitá dei porti**

$$\sum_{j=1}^{n} x_{i,j} \leq P_i \text{ per ogni } i = 1, \ldots, m,$$

dove $P_i$ é la disponibilitá di grezzo al porto $i$.

**vincoli di fabbisogno industriale**

$$\sum_{i=1}^{m} x_{i,j} \geq F_j \text{ per ogni } j = 1, \ldots, n,$$

dove $F_j$ esprime il fabbisogno dell'impianto $j$.

**vincolo di cartello**

$$\sum_{i=1}^{m} \sum_{j=1}^{n} x_{i,j} \leq C,$$

dove $C$ é il limite complessivo sull'utilizzo di grezzo imposto dal cartello.

**vincolo di impiego al massimo della potenzialitá di produzione** Prova a coniugare le indicazioni fornite trattando l'istanza particolare nel seno del contesto generale, sperimentando quale generalitá minima di approccio sia richiesta e cosa ti riesca di fare.

---

**Problema 2 (2+2+2=6 punti):**

Ad ogni casella $(i, j)$, $i, j = 1, \dots, n$ di una scacchiera $n \times n$ é associato un valore $v_{i,j}$. Quando sulla casella $(i, j)$ viene collocata una torre, si acquisisce il valore $v_{i,j}$ ad essa associato. Vogliamo collocare un set di al piú $k$ torri in modo da massimizzare la somma dei valori acquisiti. Non é peró consentito di collocare due o piú torri su una stessa riga o colonna.

(**(2pt)**) Si formuli questo problema di ottimizzazione come un problema di programmazione lineare intera (PLI).

(**(2pt)**) Si mostri come il vincolo sul numero delle torri sia inessenziale, riducendo ogni istanza del problema ad una nuova istanza in cui $k = n$.

(**(2pt)**) Questo problema é in P oppure NP-hard? Perché?

**svolgimento.**

(**(2pt)**) Le variabili di decisione sono le $n^2$ variabili booleane $x_{i,j}$, $i, j = 1, \dots, n$, una per ogni casella, e definite da

$$x_{i,j} = \begin{cases} 1 & \text{se nella casella } (i, j) \text{ viene collocata una torre.} \\ 0 & \text{altrimenti.} \end{cases}$$

Il problema è quindi quello di massimizzare i valori acquisiti.

$$\max \sum_{i,j} v_{i,j} x_{i,j} \ ,$$

nel rispetto dei seguenti vincoli:

**vincoli di interezza e dominio**

$$x_{i,j} \in \{0, 1\} \text{ per ogni } i, j = 1, \dots, n.$$

**vincoli di massimo una torre per riga**

$$\sum_{j=1}^{n} x_{i,j} \leq 1 \text{ per ogni } i = 1, \dots, n.$$

**vincoli di massimo una torre per colonna**

$$\sum_{i=1}^{n} x_{i,j} \leq 1 \text{ per ogni } j = 1, \dots, n.$$

**vincolo sul numero massimo di torri**

$$\sum_{i=1}^{n} \sum_{j=1}^{n} x_{i,j} \leq k.$$

(**(2pt)**) Dove $M$ sia un valore sufficientemente grande, ad esempio $M := \max_{i,j} v_{i,j}$, possiamo considerare una scacchiera di $N = n + (n - k)$ righe e colonne, con

$$v'_{i,j} = \begin{cases} v_{i,j} & \text{per ogni } i,j = 1,\dots,n, \\ 0 & \text{per ogni } i,j = n+1,\dots,N, \\ M & \text{altrimenti.} \end{cases}$$

(**(2pt)**) Il problema é in P in quanto, una volta sbarazzatisi del vincolo su $k$, puó essere visto come un problema di massimo matching pesato su grafo bipartito. Basti costruire il grafo che ha un nodo "maschio" per ogni riga ed un nodo "femmina" per ogni colonna, e dove $v_{i,j}$ sia il valore del matrimonio tra $i$ e $j$.

---

**Problema 3 (6 punti):**
Si consideri la soluzione $x_3 = x_6 = x_7 = 0$, $x_1 = 6$, $x_2 = 5$, $x_4 = 10$, $x_5 = 14$ del seguente problema.

$$\max\ x_1 + 6x_2 + C_3x_3 + 19x_4 + 10x_5 + C_6x_6 + C_7x_7$$
$$\begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \leq 36 \\ x_3 + x_4 + x_7 \leq 10 \\ x_5 + x_6 + x_7 \leq 14 \\ x_1 + x_3 + x_5 \leq 20 \\ x_2 + x_4 + x_6 \leq 15 \\ x_1, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0 \end{cases}$$

3.1.**(1pt)** Verificare esplicitamente che la soluzione proposta è ammissibile.

3.2.**(1pt)** Scrivere il problema duale.

3.3.**(1pt)** Impostare il sistema che esprima le condizioni agli scarti complementari.

3.4.**(1pt)** Risolvere il sistema per trovare una soluzione duale complementare alla soluzione primale fornita.

3.5.**(1pt)** Per quali valori dei parametri $C_3$, $C_6$ e $C_7$ la soluzione assegnata è ottima? Indica con chiarezza tutte le verifiche che sei stato chiamato a compiere.

3.6.**(1pt)** Per $C_3 = C_6 = C_7 = 10$, quanto sarei disposto a pagare per incrementare di un'unità il termine noto di ciascuno dei 5 vincoli?

**svolgimento.** Tutti i valori assegnati alle variabili sono non-negativi. Sostituendo tali valori negli ulteriori vincoli, conduciamo i seguenti controlli a verificare l'ammissibilità della soluzione proposta.

$$\begin{cases} (6) + (5) + (0) + (10) + (14) + (0) + (0) = 35 \leq 36 \\ (0) + (10) + (0) = \mathbf{10} \leq 10 \\ (14) + (0) + (0) = \mathbf{14} \leq 14 \\ (6) + (0) + (14) = \mathbf{20} \leq 20 \\ (5) + (10) + (0) = \mathbf{15} \leq 15 \end{cases}$$

Il problema duale è il seguente.

$$
\min\ 12\,y_1 + 10\,y_2 + 14\,y_3 + 20\,y_4 + 15\,y_5
$$
$$
\begin{cases}
y_1 & & & + & y_4 & & & \geq & 1 \\
y_1 & & & & & + & y_5 & \geq & 6 \\
y_1 + & y_2 & & + & y_4 & & & \geq & C_3 \\
y_1 + & y_2 & & & & + & y_5 & \geq & 19 \\
y_1 & & + & y_3 + & y_4 & & & \geq & 10 \\
y_1 & & + & y_3 & & + & y_5 & \geq & C_6 \\
y_1 + & y_2 + & & y_3 & & & & \geq & C_7 \\
\multicolumn{9}{l}{y_1, y_2, y_3, y_4, y_5 \geq 0}
\end{cases}
$$

Dalle condizioni degli scarti complementari segue $y_1 = 0$ poichè il vincolo 1 del primale non è soddisfatto ad eguaglianza. Inoltre, poichè $x_1, x_2, x_4, x_5 > 0$, i vincoli 1,2,4 e 5 del duale dovranno essere soddisfatti ad eguaglianza e quindi otteniamo le segunti equazioni.

$$
\begin{cases}
 & & + & y_4 & & & = & 1 \\
 & & & & + & y_5 & = & 6 \\
y_2 & & & & + & y_5 & = & 19 \\
 & y_3 + & & y_4 & & & = & 10
\end{cases}
$$

Il duale ammette pertanto un'unica soluzione che soddisfa gli scarti complementari rispetto alla soluzione primale assegnata: $(0, 13, 9, 1, 6)$. Dobbiamo ora verificare se questa soluzione duale di base sia ammissibile. È evidente che tutte le variabili assumono valore non negativo, ma dobbiamo anche andare a verificare i rimanenti vincoli del duale (i vincoli 3, 6 e 7).

La soluzione primale assegnata sarà ottima se e solo se la soluzione duale ad essa complementare soddisfa tutti i vincoli, ed in particolare anche i vincoli 3, 6 e 7, ossia se vale che $y_2 + y_4 = 14 \geq C_3$ (terzo vincolo), $y_3 + y_5 = 15 \geq C_6$ (sesto vincolo) e $y_1 + y_2 + y_3 = 22 \geq C_7$ (sesto vincolo). Possiamo concludere che la soluzione primale assegnata **è ottima se e solo se** $C_3 \leq 14$ e $C_6 \leq 15$ e $C_7 \leq 22$. In particolare, quando $C_3 = C_6 = C_7 = 10$ la soluzione fornita è ottima e i valori della soluzione duale ad essa complementare forniscono i prezzi ombra. Pertanto, i prezzi che sarei disposto a pagare per un incremento unitario sull'availability di ciascuno dei 5 vincoli sono: $0, 13, 9, 1, 6$.

---

**Problema 4 (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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 34 | 42 | 44 | 49 | 41 | 52 | 63 | 69 | 40 | 60 | 86 | 45 | 66 | 54 | 79 | 81 | 43 | 46 | 38 | 61 | 80 | 48 | 64 | 73 | 47 |

**4.1(1pt)** trovare una sottosequenza crescente che sia la più lunga possibile. Specificare quanto è lunga e fornirla.

**4.2(1pt)** una sequenza è detta una N-sequenza, o sequenza crescente con un possibile ripensamento, se esiste un indice $i$ tale cha ciascuno degli elementi della sequenza esclusi al più il primo e l'$i$-esimo sono strettamente maggiori dell'elemento che immediatamente li precede nella sequenza. Trovare la più lunga N-sequenza che sia una sottosequenza della sequenza data. Specificare quanto è lunga e fornirla.

**4.3(1pt)** trovare la più lunga sottosequenza crescente che includa l'elemento di valore 40. Specificare quanto è lunga e fornirla.

**4.4(1pt)** trovare una sottosequenza crescente che sia la più lunga possibile ma eviti di utilizzare i primi 4 elementi. Specificare quanto è lunga e fornirla.

**4.5(1pt)** trovare una sottosequenza crescente che sia la più lunga possibile ma eviti di utilizzare gli elementi dal 13-esimo a 16-esimo. Specificare quanto è lunga e fornirla.

**4.6(2pt)** fornire un minimo numero di sottosequenze decrescenti tali che ogni elemento della sequenza originale in input ricada in almeno una di esse. Specificare quante sono e fornirle.

**svolgimento.** Dapprima compilo la seguente tabella di programmazione dinamica.

CRESCENTE

| | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ | ⇒ |
| 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 |
| 34 | 42 | 44 | 49 | 41 | 52 | 63 | 69 | 40 | 60 | 86 | 45 | 66 | 54 | 79 | 81 | 43 | 46 | 38 | 61 | 80 | 48 | 64 | 73 | 47 |
| 1 | 2 | 3 | 4 | 2 | 5 | 6 | 7 | 2 | 6 | 8 | 4 | 7 | 6 | 8 | 9 | 3 | 5 | 2 | 7 | 9 | 6 | 8 | 9 | 8 |
| ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ | ⇐ |

CRESCENTE

Possiamo ora fornire le seguenti risposte.

| tipo sottosequenza | opt val | soluzione ottima |
|---|---|---|
| crescente | 9 | 34, 42, 44, 49, 52, 63, 69, 79, 81 |
| N-sequenza | 14 | 34, 42, 44, 49, 52, 63, 69, 79, 81, 43, 46, 61, 64, 73 |
| crescente con 40 | 7 | 34, 40, 45, 54, 61, 64, 73 |
| evita i primi 4 | 6 | 41, 52, 63, 69, 79, 81 |
| evita da 13-mo a 16-mo | 9 | 34, 42, 44, 49, 52, 60, 61, 64, 73 |
| minima copertura | 9 | $\underbrace{34}_{1};\ \underbrace{42, 41, 40, 38}_{2};\ \underbrace{44, 43}_{3};\ \underbrace{49, 45}_{4};\ \underbrace{52, 46}_{5};\ \underbrace{63, 60, 54, 48}_{6};\ \underbrace{69, 66, 61}_{7};\ \underbrace{86, 79, 64, 47}_{8};\ \underbrace{81, 80, 73}_{9}$ |

Dove per il penultimo punto (4.5) si é osservato dalla tabella di DP (ultima riga) che:

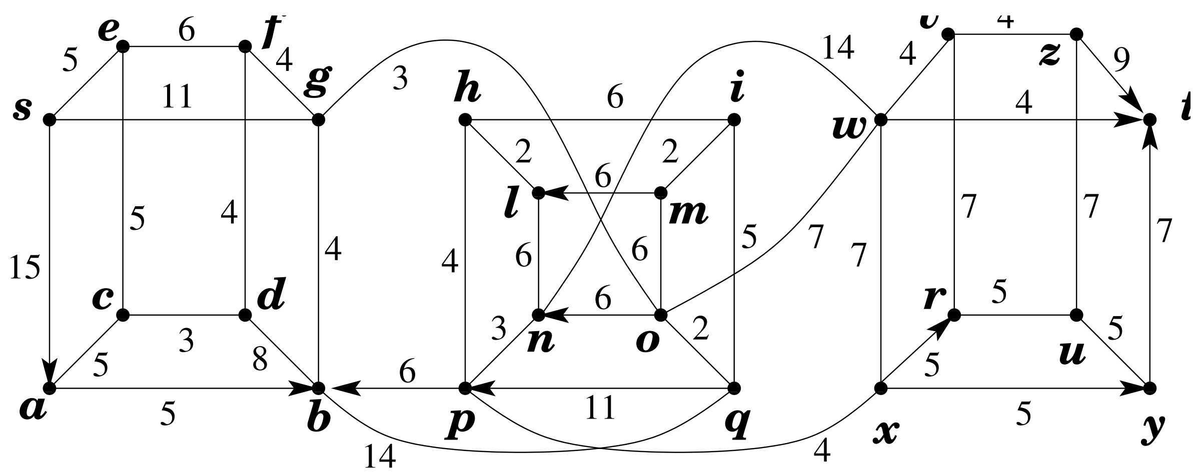
per raccogliere 8 elementi sul solo lato sinistro, l'ultimo deve valere almeno 86,
per raccogliere 7 elementi sul solo lato sinistro, l'ultimo deve valere almeno 69,
per raccogliere 6 elementi sul solo lato sinistro, l'ultimo deve valere almeno 60,
per raccogliere 5 elementi sul solo lato sinistro, l'ultimo deve valere almeno 52,
per raccogliere 4 elementi sul solo lato sinistro, l'ultimo deve valere almeno 45,
per raccogliere 3 elementi sul solo lato sinistro, l'ultimo deve valere almeno 44,
per raccogliere 2 elementi sul solo lato sinistro, l'ultimo deve valere almeno 40,
per raccogliere 1 elementi sul solo lato sinistro, esso deve valere almeno 34,

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 (4.6) ho costruito la sequenza decrescente $i$-esima collocando in essa tutti quei numeri della sequenza in input tali che la massima lunghezza di una sequenza crescente terminante in essi, come calcolata nell'ultima riga della tabella di PD, era precisamente $i$.

---

**Problema 5 (15 punti):**
Si consideri il grafo $G$, con pesi sugli archi, riportato in figura.




5.1.**(2pt)** Dire, certificandolo, (1) se il grafo $G$ è planare oppure no; (2) se il grafo $G'$ ottenuto da $G$ rimpiazzando l'arco $go$ con l'arco $gh$ è planare oppure no.

5.2.**(2pt)** Fornendo i certificati del caso, dire quale sia il minimo numero di archi la cui rimozione renda bipartito: (1) il grafo $G$; (1) il grafo $G'$.

5.3.**(1pt)** Trovare un albero ricoprente di $G$ di peso minimo.

5.4.**(3pt)** Per ciascuno dei seguenti archi dire, certificandolo, se esso appartenga a (tutte / a nessuna / a qualcuna ma non a tutte) le soluzioni ottime: $fg$, $wx$, $ln$.

5.5.**(1pt)** Trovare tutti gli alberi ricoprenti di peso minimo. (Dire quanti sono e specificare con precisione come generarli).

5.6.**(1pt)** Trovare un albero dei cammini minimi da $s$ e determinare le distanze di tutti i nodi da $s$.

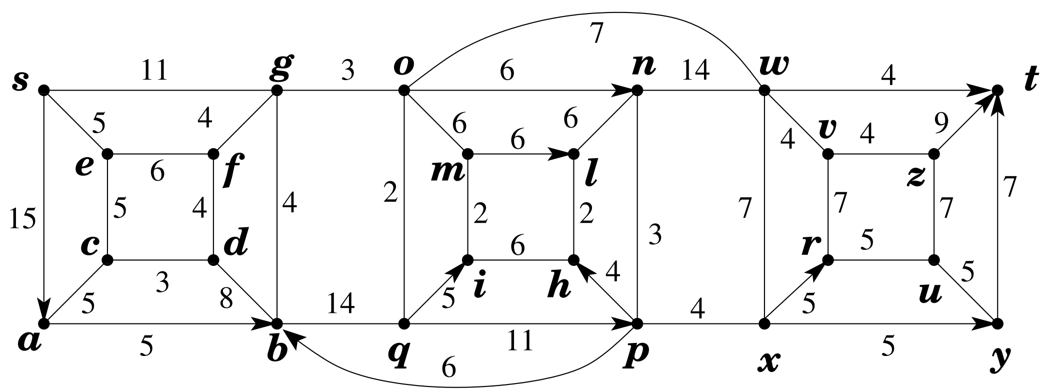
5.7.**(1pt)** Trovare tutti gli alberi dei cammini minimi da $s$. (Dire quanti sono e specificare con precisione come generarli).

5.8.**(2pt)** Trovare un massimo flusso dal nodo $s$ al nodo $t$.

5.9.**(2pt)** Certificare l'ottimalità del flusso massimo dal nodo $s$ al nodo $t$.

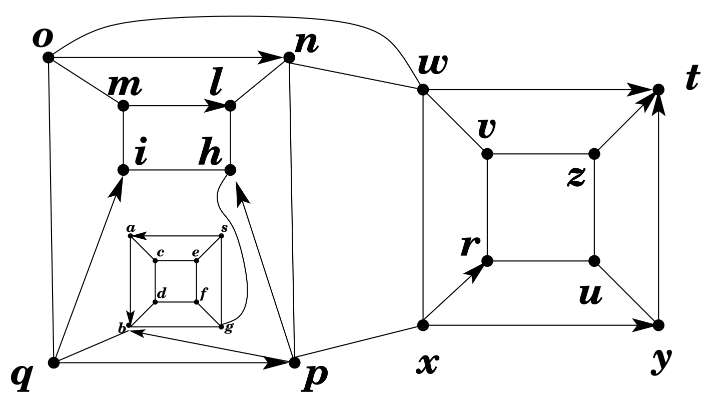
**risposte.**

Il fatto che $G$ sia planare può essere messo in evidenza esibendo il planar embedding in figura.

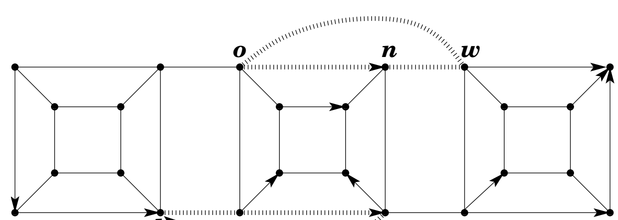



Nello svolgimento dei successivi punti converrà riferirsi al planar drawing fornito sopra.

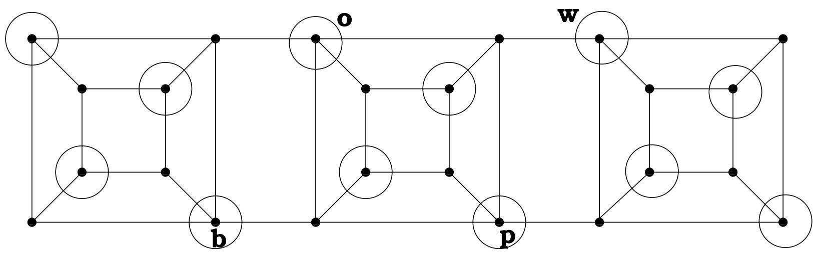
Per altro, anche $G'$ è planare come messo in evidenza (=certificato) dalla seguente figura.




Il fatto che $G$ non sia bipartito, e che sia richiesta la rimozione di almeno due archi per renderlo tale, è certificato dai due cicli dispari disgiunti sugli archi rappresentati in figura.

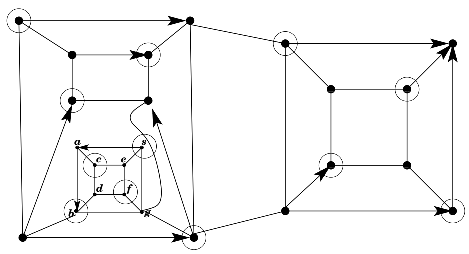



In effetti la rimozione di 2 soli archi ($ow$ e $pb$) basta a rendere $G$ bipartito come esibito in figura.




Il numero di archi la cui rimozione rende il grafo bipartito è pertanto 2.

Il grafo $G'$ ottenuto da $G$ rimpiazzando l'arco $go$ con l'arco $gh$ non é bipartito, ed almeno 2 archi devono essere rimossi per renderlo tale come messo in evidenza sempre dai 2 circuiti dispari e disgiunti sugli archi $onw$ e $bqp$. In effetti la rimozione di 2 soli archi ($ow$ e $pb$) basta a rendere $G$ bipartito come esibito in figura.

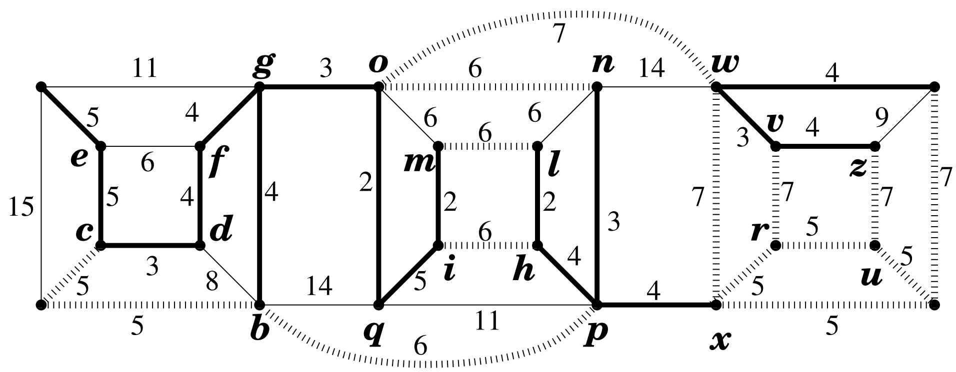


La seguente figura esprime la famiglia degli alberi ricoprenti di peso minimo. Ci sono $2 \cdot 4 \cdot 4 \cdot 5 = 160$ alberi ricoprenti di perso minimo e ciascuno di essi include i 14 archi in linea spessa, più uno qualsiasi dei 2 archi di peso 5 incidenti al nodo $a$ (i 2 archi in linea sfumata spessa presenti nella zona a sinistra), più uno qualsiasi dei 4 archi di peso 6 in linea sfumata spessa presenti nella zona centrale (gli archi $on$, $ml$, $ih$, $pb$), più uno qualsiasi dei 5 archi di peso 7 in linea sfumata spessa presenti nella zona a destra (infatti, se nel grafo $G$ contraiamo tutti gli archi di peso inferiore a 7 e rimuoviamo tutti gli archi di peso superiore a 7 ci ritroviamo con 2 soli nodi connessi da questi 4 archi disposti in parallelo), più **3** qualsiasi dei 4 archi di peso 5 in linea sfumata spessa presenti nella zona a destra (infatti, se nel grafo $G$ contraiamo tutti gli archi di peso inferiore a 5 e rimuoviamo tutti gli archi di peso superiore a 5 ci ritroviamo con una componente connessa che è un quadrato di questi 4 archi. (La componente connessa di 2 nodi connessi da 2 archi paralleli evidenzia l'intercambiabilità dei 2 archi di peso 5 incidenti al nodo $a$ di cui si era detto più sopra).

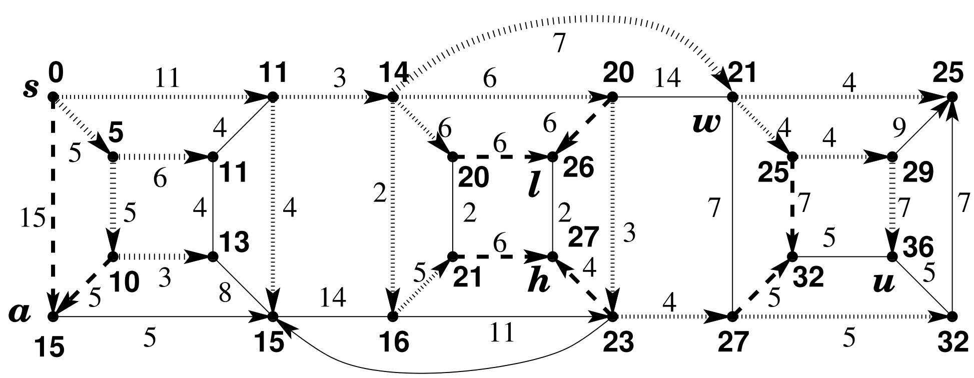
$fg$ in tutte le soluzioni ottime in quanto unico arco di peso minimo nel taglio che separa i nodi $s$, $e$, $a$, $c$, $f$, $d$ da tutti gli altri nodi;

$wx$ in qualche soluzione ottima in quanto arco di peso minimo nel taglio che separa i nodi $w$, $v$, $z$, $t$ da tutti gli altri nodi (primo certificato) ma non in tutte le soluzioni ottime in quanto arco di peso massimo nel ciclo $lnph$;

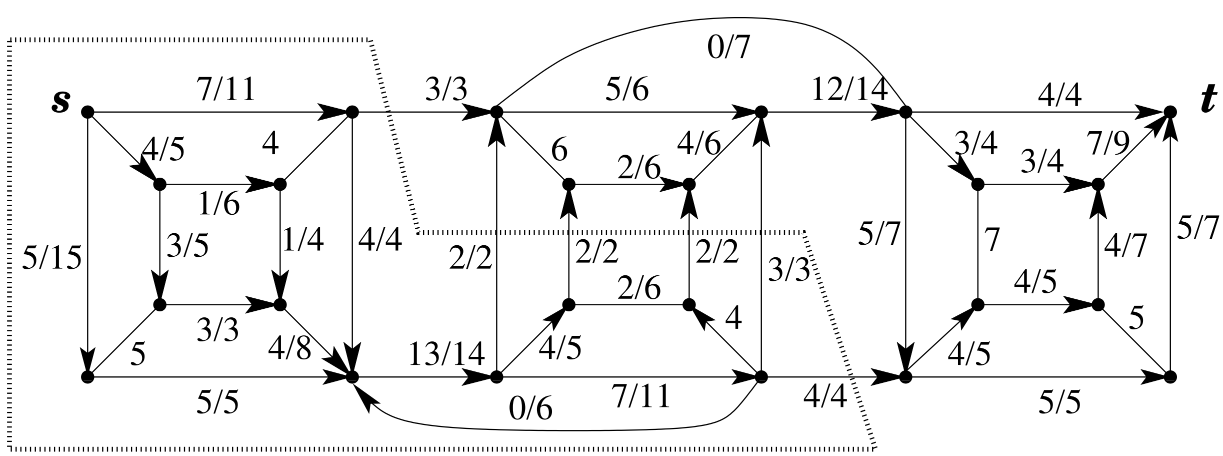
$ln$ in nessuna soluzione ottima in quanto unico arco di peso massimo nel ciclo $wxrv$.




La seguente figura esprime la famiglia degli alberi dei cammini minimi dal nodo $s$. Ci sono $2^4 = 16$ alberi dei cammini minimi dal nodo $s$ e ciascuno di essi include i 17 archi in linea spessa, più uno qualsiasi dei 2 archi tratteggiati entranti nel nodo $a$, uno qualsiasi dei 2 archi tratteggiati entranti nel nodo $h$, uno qualsiasi dei 2 archi tratteggiati entranti nel nodo $l$, e uno qualsiasi dei 2 archi tratteggiati entranti nel nodo $r$.




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 16 e satura l'insieme degli archi che attraversano la curva tratteggiata portandosi dal lato di $s$ al lato di $t$. Questi 6 archi costituiscono pertanto un minimo $s,t$-taglio, anch'esso di valore 16 e che certifica pertanto l'ottimalità del flusso proposto.

**Problema 6 (5 punti):**
Si ricerchino soluzioni algoritmiche per il seguente modello della Ricerca Operativa.

KNAPSACK* variante del KNAPSACK classico con vincolo sul numero di oggetti presi.

INPUT: Tre numeri naturali $n, k, B$ 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 di precisamente $k$ degli oggetti assegnati in input, a somma dei pesi non eccedente il budget assegnato $B$, e massimizzando il valore totale raccolto.

(**(1pt)**) Argomentare che anche questa variante del modello dello zaino, come già il KNAPSACK classico, è NP-hard.

(**(1+2+1=4pt)**) Progetto di un algoritmo pseudo-polinomiale (tramite la tecnica della programmazione dinamica):

(**(1pt)**) Definire una famiglia di (al più un numero pseudo-polinomiale di) sottoproblemi chiusa rispetto ad induzione.

(**(2pt)**) Fornire una ricorrenza risolutiva per i sottoproblemi della famiglia proposta.

(**(1pt)**) Trattare i casi base.

**svolgimento.**

(**(1pt)**) Si osservi come sia possibile ridurre il KNAPSACK classico alla versione KNAPSACK* di attuale interesse: un semplice *guess* sul possibile valore di $k$ di una soluzione ottima (ciclo for che li prova tutti uno alla volta) mostra come una procedura per la risoluzione di KNAPSACK* possa trovare impiego per risolvere istanze del KNAPSACK classico.

(**(1/3pt)**) per $n' = 0, 1, \ldots, n$, $B' = 0, 1, \ldots, B$, e $k' = 0, 1, \ldots, k$, si definisca $opt[n', B', k']$ il valore della soluzione ottima per l'istanza modificata considerando solo i primi $n'$ oggetti e ponendo $B := B'$ e $k := k'$.

(**(3/4pt)**) Riesci ora ad aggiudicarti gli altri tre punti?