Esame di Ricerca Operativa - 18 febbraio 2020
Facoltà di Scienze MM.FF.NN. - Verona
- CORREZIONE - punti in palio: 74, con voto $\geq$ punti

Problema 1 (5+10=15 punti [modellazione/riduzioni]):
Per fare un auto servono 4 ruote, un motore, ed un volante. Alla BasicCars abbiamo quattro operai (Dante, Carlo, Bruno ed Angela). Sebbene siano stati assunti con contratti diversi, con un numero diverso di ore previsto, ciascuno di loro sa produrre qualsiasi componente base, seppur con diverse produttività (espresse in unità di componente per ogni ora). I parametri in questione sono catturati nella seguente tabella:

Operaio	Ore a contratto	Produttività		
		ruote (R)	motori (M)	sterzi (S)
Angela (A)	200	10	15	20
Bruno (B)	80	20	5	10
Carlo (C)	150	15	10	5
Dante (D)	100	10	15	5

(5pt) [modellazione] Modellare con la PL il problema di determinare il numero di ore che ciascun operaio debba essere assegnato su ciascuna linea di produzione (R, M o S) in modo da massimizzare la quantità di auto complessivamente prodotte.

(10pt) [riduzione, hard] Si evidenzi come, volendo insistere sull'interezza della soluzione, il problema generale risulti sufficientemente espressivo da poter mappare in esso istanze generiche di Knapsack.

svolgimento. (5pt) Lo formuleremo come un problema di PL. La prima cosa da fare è individuare lo spazio delle scelte, ossia quanto in sostanza competa alla nostra responsabilità manageriale, mentre ogni altra cosa finisce con l'esserne in conclusione determinata. Er boss stabilisce quante ore $h_{i,j}$ l'operaio $i \in \{A, B, C, D\}$ debba dedicare a ciascuna linea $j \in \{R, M, S\}$, e questo determina ogni cosa a seguire. In linea di principio non serve introdurre ulteriori variabili, le 12 variabili introdotte sopra catturano tutto il non-determinismo insito in questo problema 'NPortante nelle applicazioni. Sappiamo tuttavia che a volte l'introduzione di qualche variabile ausiliaria può facilitarci nel meglio spezzettare alcune determinazioni derivate. Si noti che la quantità di automobili completate è condizionata dal componente resa disponibile in minor quantità. Se indichiamo con y_j la quantità di componente j ($j \in \{R, M, S\}$) prodotta complessivamente dai 4 operai, la quantità di auto ottenute, nostra funzione obiettivo che intendiamo massimizzare, risulta esprimibile nella forma $\min\{y_R, y_M, y_S\}$ che non è lineare, ma è facilmente linearizzabile inserendo una variabile y cui si richieda di soddisfare i vincoli $y \leq y_j$, $j \in \{R, M, S\}$. Se l'introduzione delle y_j è in linea di principio evitabile, non altrettanto può dirsi della y che risulta determinata non solo dalle scelte su come impiegare gli operai ma anche dal fatto che si spinge per massimizzare le auto prodotte.

L'obiettivo é quello di massimizzare il numero di automobili prodotte

$$\max y\,,$$

nel rispetto dei seguenti vincoli:

vincoli sulle componenti necessarie alla realizzazione delle auto

$$
\begin{aligned}
4y &= 10x_{A,R} + 20x_{B,R} + 15x_{C,R} + 10x_{D,R} && \text{(ruote prodotte)}\\
y &= 15x_{A,M} + 5x_{B,M} + 10x_{C,M} + 15x_{D,M} && \text{(motori prodotti)}\\
y &= 20x_{A,S} + 10x_{B,S} + 5x_{C,S} + 5x_{D,S} && \text{(sterzi prodotti)}
\end{aligned}
$$

vincoli da contratto sulle ore

$$
\begin{aligned}
x_{A,R} + x_{A,M} + x_{A,S} &\leq 200 && \text{(ore a contratto per Angela)}\\
x_{B,R} + x_{B,M} + x_{B,S} &\leq 80 && \text{(ore a contratto per Bruno)}\\
x_{C,R} + x_{C,M} + x_{C,S} &\leq 150 && \text{(ore a contratto per Carlo)}\\
x_{D,R} + x_{D,M} + x_{D,S} &\leq 100 && \text{(ore a contratto per Dante)}
\end{aligned}
$$

vincoli di non negativitá

$$x_{A,R}, x_{A,M}, x_{A,S}, x_{B,R}, x_{B,M}, x_{B,S}, x_{C,R}, x_{C,M}, x_{C,S}, x_{D,R}, x_{D,M}, x_{D,S} \geq 0.$$

Stiamo supponendo per semplicità che le variabili $x_{i,j}$ non siano vincolate ad essere intere in modo da poter proporre una modellazione di PL (per la quale sono a disposizione algoritmi polinomiali). Introducendo il vincolo di interezza su queste variabili 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). Facendo questo passiamo tuttavia da una ad una modellazione PLI (problema NP-completo).

(10pt) Vogliamo mostrare che vi sono ottime ragioni per accontentarsi dell'approssimazione introdotta rilassando i voncoli di integralità: senza tale semplificazione il problema risulta NP-completo in quanto possiamo mappare in esso istanze generiche di KNAPSACK. Siano $(val_i, peso_i)$, $i = 1, 2, \ldots, n$ delle coppie peso valore atte a descrivere n oggetti tra cui scegliere quali infilare in uno zaino di capacità W. La domanda è se esista un sottoinsieme di indici di oggetti $I \subseteq \{1, 2, \ldots, n\}$ che possano essere accomodati nello zaino in quanto $\sum_{i \in I} peso_i \leq W$ e che totalizzino un valore complessivo $\sum_{i \in I} val_i \geq Q$ tale da raggiungere un certo quorum Q dato anch'esso in input. Mostriamo di seguito come sia sempre possibile mappare tale istanza nel problema che ci è stato chiesto di analizzare anche nei suoi aspetti computazionali. Formuliamo la generica domanda di KNAPSACK appena descritta come segue. Ci chiediamo se sia possibile costruire un'automobile, dove ogni automobile abbia bisogno di Q marmitte, $\sum_{i=1}^{n} peso_i - W$ tubi di scarico, e di un fusibile di tipo i per $i = 1, \ldots, n$. Disponiamo di $2n$ operai, con l'operaio $(2i - 1)$ assunto con un contratto di val_i ore che potrebbe impiegare per produrre una marmitta all'ora oppure, impiegando tutte le sue val_i ore, sarebbe in grado di costruire un fusibile di tipo i, per $i = 1, \ldots, n$. Sempre per $i = 1, \ldots, n$, anche l'operaio $2i$, disponendo di tutte e $peso_i$ le sue ore a contratto, sarebbe in grado di costruire un fusibile di tipo i, ma potrebbe altresì impiegare quel tempo per costruire tubi di scappamento alla velocità di un tubo di scappamento all'ora. La descrizione della riduzione è ora completa in tutti i suoi parametri e struttura. In pratica vogliamo che tutte le ore degli operai di indice pari siano impiegate nella produzione di tubi di scappamento, tranne al più W di esse. Si noti però che per ciascuno di questi operai di indice pari che non realizzi tutti i suoi $peso_i$ tubi di scappamento, meglio lo faccia per una buona causa: realizzerà lui il fusibile di tipo i (e non produrrà quindi alcun tubo di scappamento); ciò consentirà all'operaio $(2i-1)$ di concentrasi esclusivamente sulle marmitte producendone val_i.

Lasciamo per esercizio la dimoastazione del "lemma facile":

> Se con riferimento all'istanza originaria di KNAPSACK esiste un sottoinsieme di indici di oggetti $I \subseteq \{1, 2, \ldots, n\}$ tale che $\sum_{i \in I} peso_i \leq W$ e $\sum_{i \in I} val_i \geq Q$, allora la nostra azienda sarà in grado di produrre un'automobile.

e del "lemma difficile":

> Se l'azienda è in grado di produrre un'automobile allora esiste un sottoinsieme di indici di oggetti $I \subseteq \{1, 2, \ldots, n\}$ tale che $\sum_{i \in I} peso_i \leq W$ e $\sum_{i \in I} val_i \geq Q$.

Problema 2 (3+3+3+3=12 punti):

Un grafo diretto (ossia con gli archi che hanno tutti un verso, ossia una testa ed una coda) si dice aciclico se non contiene cicli diretti. Il nostro scopo è riconoscere se un grafo diretto $G = (V, A)$ è aciclico. Invece che progettare un algoritmo apposito e codificarlo, vorremo ridurre questa domanda ad un problema di programmazione lineare.

(3pt) Mostra come, a partire da G, sia possibile comporre un problema $P_{NO}(G)$, di programmazione lineare, tale che $P_{NO}(G)$ sia soddisfacibile se e solo se G non è aciclico.

(3pt) Mostra come, a partire da G, sia possibile comporre un problema $P_{SI}(G)$, di programmazione lineare, tale che $P_{SI}(G)$ sia soddisfacibile se e solo se G è aciclico.

(3pt) Mostra come ottenere un certificato di non aciclicità di G a partire da una soluzione ammissibile di $P_{NO}(G)$.

(3pt) Mostra come ottenere un certificato di aciclicità di G a partire da una soluzione ammissibile di $P_{SI}(G)$.

svolgimento.

(3pt) componiamo un problema $P_{NO}(G)$, di programmazione lineare, tale che $P_{NO}(G)$ sia soddisfacibile se e solo se G non è aciclico.

Rimuoviamo le due negazioni contrapposte: G non aciclico significa che G contiene un ciclo, ossia una questione chiaramente in NP (come certificato di sì basta fornire un ciclo) e pertanto sicuramente riducibile alla PLI. Per modellarla come un problema di PL, pensiamo ad un ciclo come ad una circolazione di flusso. Introduciamo quindi una variabile di flusso x_a per ogni arco $a = (u, v) \in A$ e scriviamo l'equazione di conservazione del flusso ad ogni nodo $v \in V$.

$$\sum_{a=(u,v) \in A} x_a + \sum_{a=(v,z) \in A} x_a = 0 \qquad \text{(per ogni nodo } v \in V. \tag{1}$$

$$\tag{2}$$

A queste n equazioni di conservazione del flusso aggiungiamo una diseguaglianza atta ad escludere il flusso identicamente nullo. Ad esempio:

$$\sum_{a \in A} x_a \geq 1. \tag{3}$$

$$\tag{4}$$

Se C è un ciclo diretto di G allora il vettore caratteristico $\chi_C : A \in\mapsto \{0,1\}$ con $\chi_C(a) = 1$ se e solo se $a \in C$ soddisfa a tutti i vincoli introdotti, come anche ai vincoli $0 \leq x_a \leq 1$ per ogni $a \in A$.

Viceversa, se $x \in [0,1]^m$ soddisfa il Vincolo 3, allora c'è un arco $a_0 = (\overline{u}_0, \overline{v}_0)$ tale che $x_{a_0} > 0$. Se noi mettiamo una pedina in v, e combiniamo il vincoli di conservazione del flusso per il nodo v con il fatto che $x_{a_0} > 0$, allora c'è un arco $a_1 = (\overline{v}_0, \overline{v}_1)$ tale che $x_{a_1} > 0$. Questo ragionamento può essere iterato producendo una sequenza di archi $a_0, a_1, a_2, \ldots, a_k$ che dovrà infine chiudere un ciclo.

Con questo ragionamento abbiamo in realtà mostrato **(3pt)** come ottenere un certificato di non aciclicità di G a partire da una soluzione ammissibile di $P_{NO}(G)$.

(3pt) componiamo un problema $P_{SI}(G)$, di programmazione lineare, tale che $P_{SI}(G)$ sia soddisfacibile se e solo se G è aciclico.

E' nota la seguente buona caratterizzazione dei DAG:

> un grafo $G = (V, A)$ è aciclico se e solo se esiste un ordinamento $<$ dei suoi nodi tale che per ogni arco $(u, v) \in A$ vale $u < v$.

In altre parole, come certificato di aciclicità di un grafo G possiamo fornire una messa in linea dei suoi nodi. Introduciamo quindi una variabile x_v per ogni nodo $v \in V$ e tesa a rappresentare l'ascissa del punto della linea dove il nodo v và a collocarsi.

Vogliamo che, con riferimento a questo embedding dei nodi, tutti gli archi siano rivolti in avanti, ossia:

$$x_v \geq x_u + 1 \qquad \text{(per ogni arco } (u,v) \in A. \tag{5}$$

$$\tag{6}$$

Non ci sono problemi ad assumere $0 \leq x_a \leq n$ per ogni $v \in V$, c'è abbastanza spazio dato che i nodi sono n in tutto. Quindi, dato un ordinamento dei nodi, possiamo prendere come x_v la posizione del nodo v dentro l'ordine, e questo vettore di n valori soddisfa tutti gli m Vincoli 5.

Viceversa, se $x \in [0,n]^n$ soddisfa i Vincoli 5, allora possiamo vederlo come un embedding dei nodi del grafo sulla linea, e tutti gli archi sono effettivamente rivolti in avanti.

Con questo ragionamento abbiamo in realtà mostrato come ottenere un topological sort ei nodi, ossia un certificato di aciclicità di G a partire da una soluzione ammissibile di $P_{SI}(G)$.

Chiaramente non è possibile che G contenga un ciclo diretto ed al tempo stesso ammetta un topological sort. Un grafo non può cioè avere sia un certificato di sì che di no.

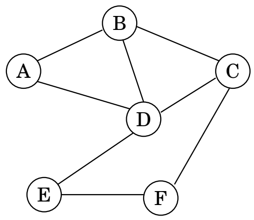
Si potrebbe pensare di utilizzare la teoria della dualità per andare a dimostrare che, per ogni grafo G, si ha almeno uno di questi due certificati (buona caratterizzazione)?

Problema 3 (2+2+5=9 punti):

Un MATCHING in un grafo $G = (V, E)$ è un sottoinsieme di archi $M \subseteq E$ tale che ogni nodo in V è estremo di al più un arco in M. Un matching di G è detto massimale se non esiste un altro matching di G che lo contenga propriamente.

Ad esempio, $\{AB, DE\}$ e $\{DC, EF\}$ sono due matchings non-massimali mentre $\{BC, DE\}$ e $\{AB, DE, CF\}$ sono due matchings massimali per il grafo G in figura.

Quando ad ogni arco e è associato un costo w_e, allora il costo di $X \subseteq E$ è espresso da $val(X) := \sum_{e \in X} w_e$.



	AB	AD	BC	BD	CD	CF	DE	EF
Costo	12	13	15	14	11	16	17	18

Siamo interessati a trovare matching massimali di costo minimo.

(2pt) Formulare come un problema di Programmazione Lineare Intera (PLI) la ricerca di un matching massimale di costo minimo per il grafo G in figura.

(2pt) Mostrare come sia più in generale possibile formulare come un problema di Programmazione Lineare Intera (PLI) la ricerca di un matching massimale di costo minimo su un grafo $G = (V, E)$ generico.

(5pt) Dimostrare che la ricerca di un matching massimale di costo minimo su un grafo $G = (V, E)$ generico è un problema NP-hard riducendo ad esso il problema del minimo node cover.

svolgimento.

(2pt) Abbiamo una variabile $x_i \in \{0, 1\}$ per $i = AB, AD, BC, BD, CD, CF, DE, EF$, con l'idea che 1 significa "arco incluso nel matching massimale" e 0 significa "arco non incluso nel matching massimale".

Volendo minimizzare il costo del matching massimale, la funzione obbiettivo sarà:

$$\min 12\,x_{AB} + 13\,x_{AD} + 15\,x_{BC} + 14\,x_{BD} + 11\,x_{CD} + 16\,x_{CF} + 17\,x_{DE} + 18\,x_{EF}$$

Abbiamo due insiemi di vincoli.

matching. Dobbiamo imporre che la soluzione sia un matching. Predisponiamo dei vincoli che corrispondono ai nodi.

nodo A: $x_{AB} + x_{AD} \leq 1$;

nodo B: $x_{AB} + x_{BC} + x_{BD} \leq 1$;

nodo C: $x_{BC} + x_{CD} + x_{CF} \leq 1$;

nodo D: $x_{AD} + x_{BD} + x_{CD} + x_{DE} \leq 1$;

nodo E: $x_{DE} + x_{EF} \leq 1$;

nodo F**:** $x_{EF} + x_{CF} \leq 1$.

massimalità. Dobbiamo imporre che il matching sia massimale. Predisponiamo dei vincoli che corrispondono agli archi.

arco AB**:** $x_{AB} + x_{AD} + x_{BC} + x_{BD} \geq 1$;

arco AD**:** $x_{AD} + x_{AB} + x_{BD} + x_{CD} + x_{DE} \geq 1$;

arco BD**:** $x_{BD} + x_{AB} + x_{BC} + x_{AD} + x_{CD} + x_{DE} \geq 1$;

arco BC**:** $x_{BC} + x_{AB} + x_{BD} + x_{CD} + x_{CF} \geq 1$;

arco CD**:** $x_{CD} + x_{BC} + x_{CF} + x_{AD} + x_{BD} + x_{DE} \geq 1$;

arco CF**:** $x_{CF} + x_{BC} + x_{CD} + x_{EF} \geq 1$;

arco DE**:** $x_{DE} + x_{AD} + x_{BD} + x_{CD} + x_{EF} \geq 1$;

arco EF**:** $x_{EF} + x_{DE} + x_{CF} \geq 1$.

(2pt) Nel caso di un grafo $G = (V, E)$ generico introduciamo una variabile $x_{uv} \in \{0, 1\}$ per ogni arco $uv \in E$, con l'idea che 1 significa "arco incluso nel matching massimale" e 0 significa "arco non incluso nel matching massimale".

Otteniamo quindi la seguente formulazione PLI per il problema del MAXIMAL-MATCHING di costo minimo.

$$\begin{aligned} \min \sum_{uv \in E} C_{uv} x_{uv} \,, \\ \sum_{e \in \delta(v)} x_e \leq 1 \text{ per ogni nodo } v \in V, \\ x_{uv} + \sum_{e \in \delta(u) \setminus \{uv\}} + \sum_{e \in \delta(v) \setminus \{uv\}} \geq 1 \text{ per ogni arco } uv \in E, \\ x_{uv} \in \{0, 1\} \text{ per ogni arco } uv \in E. \end{aligned}$$

(5pt) Partendo da un generico grafo non-pesato $H = (V, E)$ assegnatoci come istanza di NODECOVER, costruiamo un grafo pesato G assegnando costo C sufficientemente grande (si pensi $C = n + 1$, ma in realtà già $C = 2$ basterebbe) ad ogni arco di H, e aggiungendo poi, per ogni nodo v di H, i due nodi v' e v'' ed i due archi vv' e $v'v''$ di peso 1 e 0 rispettivamente.

Questa richiesta era da considerarsi pienamente esaudita se si sanciva con chiarezza quanto enunciato nei seguenti due lemmi (che in questo documento conviene io anche dimostri per completezza). Ovviamente lo studente che riuscisse anche a fornire dimostrazioni chiare (in fondo, certificati) verrebbe premiato.

Lemma easy: Se H ammette un node cover X con $|X| \leq k$, allora G ammette un matching massimale M di costo al più k.

dimostrazione: Si consideri $M := \{vv' : v \in X\} \cup \{v'v'' : v \in V \setminus X\}$. Che M sia un matching massimale di G si può lasciare al lettore King Arthur, così come la verifica che il costo totale di M ammonti a $|X|$. (O altrimenti si architetta un modo elegante ed efficace per metterlo in evidenza senza ingombrare.)

Lemma hard: Se G ammette un matching massimale M di costo al più k allora H ammette un node cover X con $|X| \leq k$.

dimostrazione: Perchè questo lemma valga basterebbe prendere $C = 2$, ma per offrirne una dimostrazione più semplice ed immediata assumiamo $C = n$. Dato un matching massimale M in G di costo al più k, se $k \geq n$ allora un node cover X di H con $|X| \leq k$ è dato semplicemente da $X := V$. Altrimenti sappiamo che M non include alcun arco originale di H, ossia tutti gli archi di M sono del tipo vv' (di costo 1) oppure del tipo $v'v''$ (di costo 0). Inoltre, essendo M un matching massimale di G, allora per ogni nodo $v \in V$ avremo che esattamente una delle seguenti due proprietà vale:

$P_1(v)$ il matching M contiene l'arco vv';

$P_2(v)$ il matching M contiene l'arco $v'v''$.

Sia $X := \{v \in V :$ vale la $P_1(v)\}$. Chiaramente $|X|$ è pari al costo di M e X è un node cover di H perchè per ogni arco uv di H almeno una tra $P_1(u)$ e $P_1(v)$ deve valere essendo M un matching massimale in G.

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 (strettamente) 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 che 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.

tipo sottosequenza	opt val	soluzione ottima
crescente		
N-sequenza		
crescente con 40		
evita i primi 4		
evita da 13-mo a 16-mo		
minima copertura		

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	$\underbrace{34}_{1};\ \underbrace{42, 41, 40, 38}_{2};\ \underbrace{44, 43}_{3};\ \underbrace{49, 45}_{4};\ \underbrace{52, 46}_{5};\ \underbrace{63, 60, 54, 48}_{6};\ \underbrace{69, 66, 61}_{7};\ \underbrace{86, 79, 64, 47}_{8};\ \underbrace{81, 80, 73}_{9}$

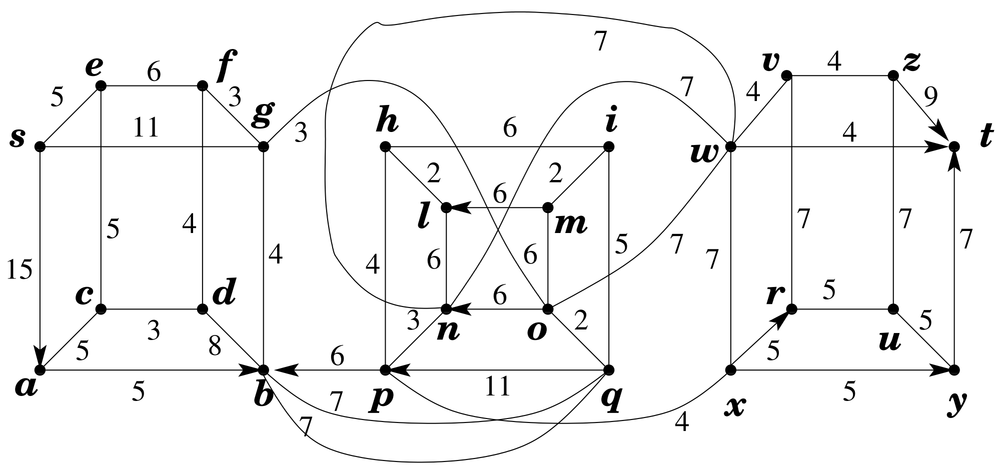
Dove per il penultimo punto (4.5) si é osservato dalla tabella di DP (ultima riga) che:

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).

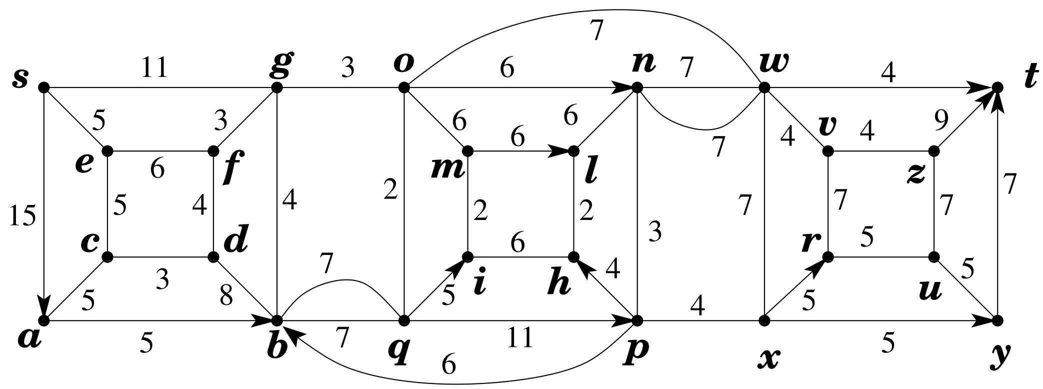
5.6.**(1pt)** Trovare un albero dei cammini minimi da s e determinare le distanze di tutti i nodi da s.

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.

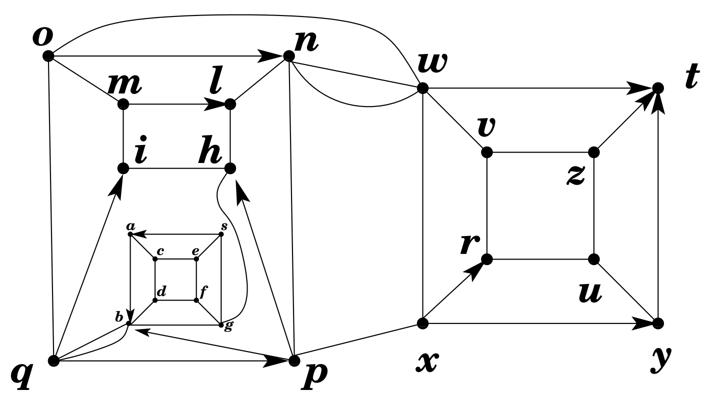
5.9.**(2pt)** Certificare l'ottimalità del flusso massimo dal nodo s al nodo t.

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

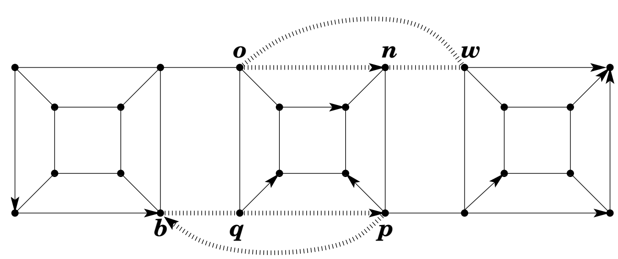


Nello svolgimento dei successivi punti converrà riferirsi al planar drawing fornito sopra.

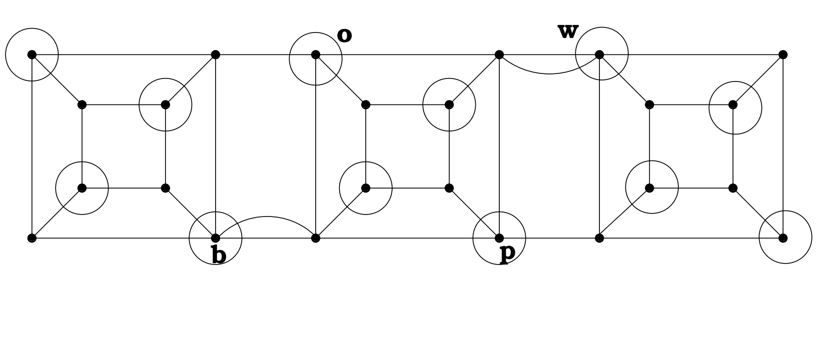
Per altro, anche G' è planare come messo in evidenza (=certificato) dalla seguente figura.



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.

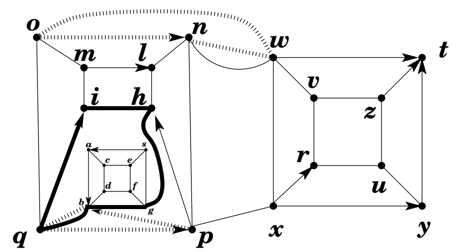


In effetti la rimozione di 2 soli archi (ow e pb) basta a rendere G bipartito come esibito in figura.

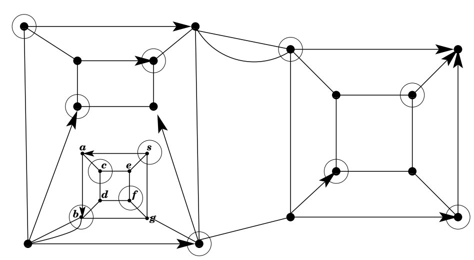


Il numero di archi la cui rimozione rende G bipartito è pertanto 2.

Il grafo G' ottenuto da G rimpiazzando l'arco go con l'arco gh non é bipartito, ed almeno 3 archi devono essere rimossi per renderlo tale come messo in evidenza dai 3 cicli dispari disgiunti sugli archi esibiti nella seguente figura (sempre i 2 triangoli onw e bqp ma anche il ciclo $ihgbq$).



In effetti la rimozione di 3 soli archi dovrà bastare a rendere G' bipartito, in quanto, una volta rimosso l'arco gh, il grafo G' risulta essere un sottografo del grafo G, e quindi possiamo a quel punto riutilizzare la soluzione a 2 archi di G (ow e pb). Tale soluzione ottima (per altro non unica, riesci a scorgerne altre?) viene esibita nella seguente figura (aderendo ora al planar embedding adottato per G').

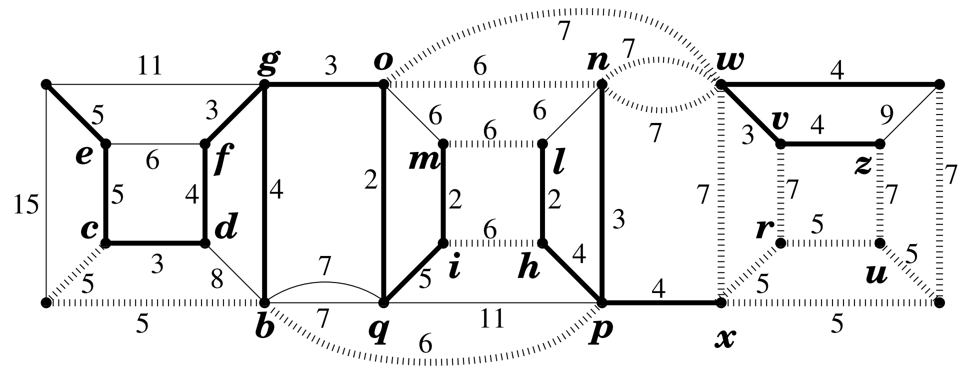


La seguente figura esprime la famiglia degli alberi ricoprenti di peso minimo. Ci sono $2 \cdot 4 \cdot 4 \cdot 7 = 224$ alberi ricoprenti di perso minimo e ciascuno di essi include i 14 archi in linea spessa, più uno qualsiasi dei 2 archi di peso 5 incidenti al nodo a (i 2 archi in linea sfumata spessa 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 7 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 6 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).

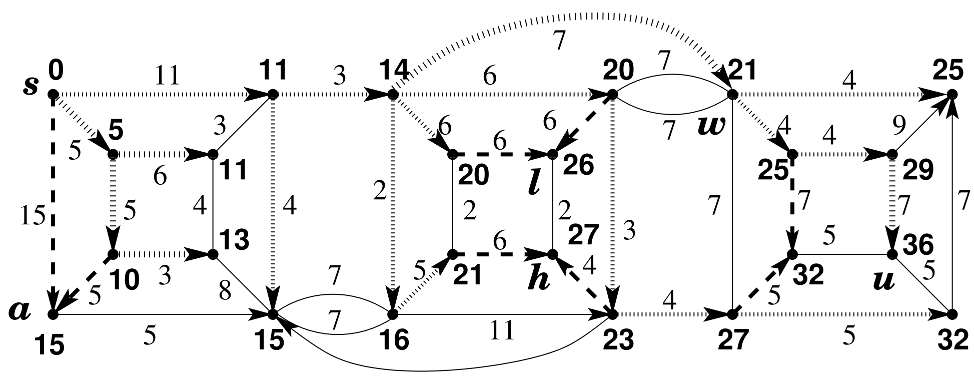
fg in tutte le soluzioni ottime in quanto unico arco di peso minimo nel taglio che separa i nodi s, e, a, c, f, d da tutti gli altri nodi;

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 $wxrv$;

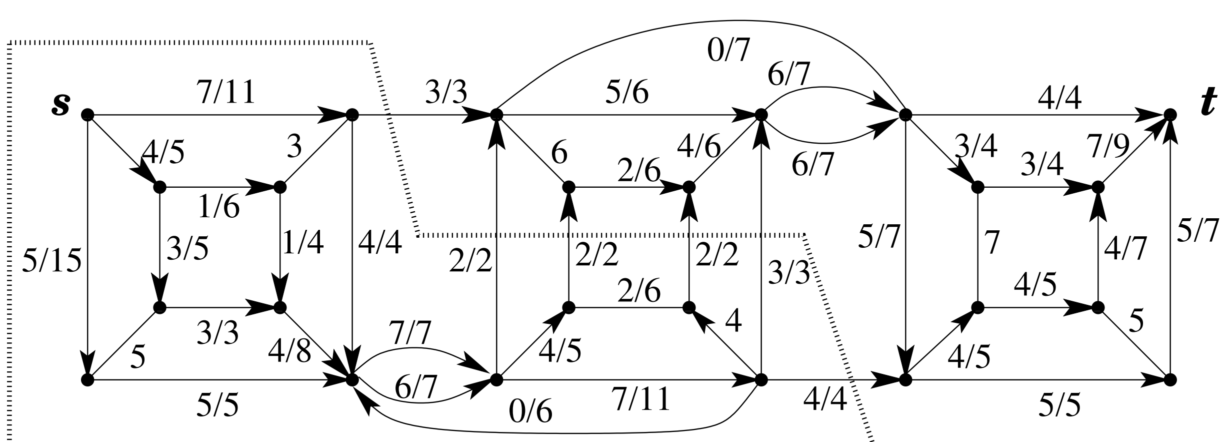
ln in nessuna soluzione ottima in quanto unico arco di peso massimo nel ciclo $lnph$.



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 20 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, 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 (16 punti):

$$\max\ 8x_1 - 6x_2 + 4x_3$$
$$\begin{cases} 3x_1 & -2x_2 & +x_3 & \leq & 0 \\ x_1 & -x_2 & +x_3 & \leq & 3 \\ x_1 & +x_2 & +x_3 & \geq & 9 \\ & x_2 & & \leq & 21 \\ \multicolumn{5}{l}{x_1 \geq 0, x_2 \leq 0, x_3 \leq 0} \end{cases}$$

6.1(1pt) Portare in forma standard.

6.2(1pt) Impostare il problema ausiliario.

6.3(1pt) Scrivere il primo tableau per il problema ausiliario ed osservare che la soluzione di base che esso esprime non è ammissbile.

6.4(1pt) Compiere un primo pivot per conquistare l'ammissibilità della soluzione di base associata al secondo tableau.

6.5(1pt) Risolvere il problema ausiliario all'ottimo.

6.6(1pt) Stabilire se il problema originale era ammissibile: in caso negativo si concluda il presente esercizio offrendo dimostrazione della sua non-ammissibilità, in caso posivo si produca un primo tableau con soluzione di base associata ammissibile per il problema originario in forma standard e si prosegua con le domande a seguire.

6.7(1pt) Si verifichi la correttezza di tale tableau con la prova del 9 della PL (benchè non ammissibile, si impieghi l'origine come soluzione di base ovvia e di immediata computazione).

6.8(1pt) Anche ove fosse non ammissibile, si renda esplicita la soluzione duale di base associata a questo primo dizionario per la seconda fase.

6.7(2+1pt+1pt+1pt) Risolvere il problema originario all'ottimo. I punti aggiuntivi vengono attribuiti se ad ognuno dei diversi pivot che dovrai compiere effettuerai esplicitamente una prova del 9: un punto se almeno uno dei dizionari lo verifichi con la prima soluzione di base primale ammissibile, un punto se almeno uno dei dizionari lo verifichi con la prima soluzione di base duale ammissibile, un punto se verifichi con almeno una soluzione tutti i dizionari visitati. Come ogni altra evidenza che date per ottenere punti, queste prove devono essere offerte in modo chiaro ed esplicito, e cosiglio di incorniciare ogni vostra risposta che miri a diventare punti.

6.7(1+1pt) rendere esplicita la soluzione duale ottima. Utilizzarla per dimostrare l'ottimalità della soluzione primale.

6.5(1pt) Quanto si sarebbe disposti a pagare per ogni unità di variazione in ciascuno dei termini noti dei tre vincoli? (Per piccole variazioni.)

6.6(2pt) Fino a dove si sarebbe disposti a pagare tali prezzi ombra?

svolgimento.

(1pt) Portiamo dapprima il problema in forma standard riesprimendolo in termini delle variabili x_1, $x'_3 = x_3$ e $x'_2 = -x_2$ e moltiplicando per -1 il vincolo di $\geq$:

$$\max\ 8x_1 + 6x'_2 - 4x'_3$$
$$\begin{cases} \begin{array}{rrrrr} 3x_1 & +2x'_2 & -x'_3 & \leq & 0 \\ x_1 & +x'_2 & -x'_3 & \leq & 3 \\ -x_1 & +x'_2 & +x'_3 & \leq & -9 \\ & -x'_2 & & \leq & 21 \end{array} \\ x_1, x'_2, x'_3 \geq 0 \end{cases}$$

(1pt) A questo punto il problema è in forma standard $\max\{cx : Ax \leq b, x \geq 0\}$. Pertanto ha almeno una soluzione di base, l'origine. Ma questo problema non è ad origine ammissibile dato che uno dei 4 termini noti è negativo (-3). Rimaniamo quindi col problema di reperire una prima soluzione di base ammissibile dalla quale avviare il metodo del simplesso. Non è un problema scontato, di fatto tale soluzione potrebbe non esistere (se il problema non ammette soluzioni ammissibili).

Il secondo punto dell'esercizio chiede di superare la difficoltà del reperire una prima soluzione di base ammissibile con il metodo del problema ausiliario. Il problema ausiliario è sempre ammissibile ed è ottenuto introducendo una variabile "di colla" x_0. In prima battuta del problema originario essenzialmente ci interessa investigare solo l'ammissibilità, e quindi viene gettata a mare la funzione obiettivo originaria e ci si prefigge invece di minimizzare la quantità di colla necessaria all'ottenimento dell'ammissibilità.

$$\max\ -x_0$$
$$\begin{cases} \begin{array}{rrrrrr} 3x_1 & +2x'_2 & -x'_3 & -x_0 & \leq & 0 \\ x_1 & +x'_2 & -x'_3 & -x_0 & \leq & 3 \\ -x_1 & +x'_2 & +x'_3 & -x_0 & \leq & -9 \\ & -x'_2 & & -x_0 & \leq & 21 \end{array} \\ x_0, x_1, x'_2, x'_3 \geq 0 \end{cases}$$

Il problema ausiliario è sempre ammissibile (basta prendere un valore sufficientemente grande per x_0) e, ovviamente, il problema originario è ammissibile se e solo se il problema ausiliario ammette una soluzione ammissibile con $x_0 = 0$. Questa è la prima domanda che siamo chiamati ad affrontare, e lo faremo nella prima fase del metodo del simplesso, quella che identifica una soluzione di base ottima per il problema ausiliario.

Introduciamo le variabili di slack come segue.

$$\max\ -x_0$$
$$\begin{cases} \begin{array}{rrrrrrr} w_1 & = & 0 & -3x_1 & -2x'_2 & +x'_3 & +x_0 \\ w_2 & = & 3 & -x_1 & -x'_2 & +x'_3 & +x_0 \\ w_3 & = & -9 & +x_1 & -x'_2 & -x'_3 & +x_0 \\ w_4 & = & 21 & & +x'_2 & & +x_0 \end{array} \\ x_0, x_1, x'_2, x'_3, w_1, w_2, w_3, w_4 \geq 0 \end{cases} \iff$$

Tableau Iniziale

		x_1	x'_2	x'_3	x_0
z	0	0	0	0	-1
w_1	0	-3	-2	1	1
w_2	3	-1	-1	1	1
w_3	-9	1	-1	-1	$\boxed{1}$
w_4	21	0	1	0	1

Le equazioni con cui abbiamo definito le variabili di slack definiscono il nostro primo dizionario dal quale prende avvio il metodo del simplesso. La soluzione di base associata al primissimo dizionario è l'origine $(x_0, x_1, x'_2, x'_3) = (0, 0, 0, 0)$ che tuttavia non è ammissibile benchè ovvimente il problema ausiliario sia ammissibile per costruzione. Fortunatamente, un primo pivot risulta sempre sufficiente a raggiungere una prima soluzione di base ammissibile nel caso del problema ausiliario: facciamo entrare x_0 in base settandone il valore a -9 (si guarda al vincolo con termine noto più negativo) e facendo uscire di base la variabile di slack per quel vincolo (w_3). L'elemento di pivot è pertanto quello incorniciato e, con un passo di pivot, si perviene alla situazione seguente.

$$
\max \; -9 + x_1 - x'_2 - x'_3 - w_3
$$
$$
\begin{cases}
w_1 = 9 - 4x_1 - x'_2 + 2x'_3 + w_3 \\
w_2 = 12 - 2x_1 + 2x'_3 + w_3 \\
x_0 = 9 - x_1 + x'_2 + x'_3 + w_3 \\
w_4 = 30 - x_1 + 2x'_2 + x'_3 + w_3 \\
x_0, x_1, x'_2, x'_3, x_4, w_2, w_3 \geq 0
\end{cases}
\iff
$$

Secondo Tableau

		x_1	x'_2	x'_3	w_3
z	-9	1	-1	-1	-1
w_1	9	$\boxed{-4}$	-1	2	1
w_2	12	-2	0	2	1
x_0	9	-1	1	1	1
w_4	30	-1	2	1	1

La soluzione di base attuale non è ancora ottima in quanto nella funzione obiettivo sono presenti dei coefficienti positivi (quello della x_1 e nessun altro). La colonna di pivot sarà pertanto quella della x_1. La prima variabile in base ad annullarsi al crescere della x_1 sarebbe la w_1. Pertanto scegliamo la w_1 come variabile uscente e riga del pivot.

$$
\max \; -\tfrac{27}{4} - \tfrac{1}{4}w_1 - \tfrac{5}{4}x'_2 - \tfrac{1}{2}x'_3 - \tfrac{3}{4}w_3
$$
$$
\begin{cases}
x_1 = \frac{9}{4} - \frac{1}{4}w_1 - \frac{1}{4}x'_2 + \frac{1}{2}x'_3 + \frac{1}{4}w_3 \\
w_2 = \frac{15}{2} + \frac{1}{2}w_1 + \frac{1}{2}x'_2 + \frac{3}{2}x'_3 + \frac{1}{2}w_3 \\
x_0 = \frac{27}{4} + \frac{1}{4}w_1 + \frac{5}{4}x'_2 + \frac{1}{2}x'_3 + \frac{3}{4}w_3 \\
w_4 = \frac{111}{4} + \frac{1}{4}w_1 + \frac{9}{4}x'_2 + \frac{1}{2}x'_3 + \frac{3}{4}w_3 \\
x_0, x_1, x'_2, x'_3, x_4, w_2, w_3 \geq 0
\end{cases}
\iff
$$

Terzo Tableau

		w_1	x'_2	x'_3	w_3
z	$-\frac{27}{4}$	$-\frac{1}{4}$	$-\frac{5}{4}$	$-\frac{1}{2}$	$-\frac{3}{4}$
x_1	$\frac{9}{4}$	$-\frac{1}{4}$	$-\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$
w_2	$\frac{15}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{3}{2}$	$\frac{1}{2}$
x_0	$\frac{27}{4}$	$\frac{1}{4}$	$\frac{5}{4}$	$\frac{1}{2}$	$\frac{3}{4}$
w_4	$\frac{111}{4}$	$\frac{1}{4}$	$\frac{9}{4}$	$\frac{1}{2}$	$\frac{3}{4}$

La soluzione associata a questo terzo dizionario è ottima, ed in essa la variabile di colla riparatutto, per quanto minimizzata, vale $x_0 = \frac{27}{4}$. Questo significa che il problema originale non era ammissibile. Nel caso del problema già in forma standard, la sua inammissibilità dovrebbe apparire evidente sommando $\frac{1}{4}$ del primo vincolo e $\frac{3}{4}$ del terzo, che è come dire, sommando 3 volte il terzo vincolo al primo. Nel caso del problema originario dell'esercizio, la sua inammissibilità dovrebbe apparire evidente togliendo il triplo del terzo vincolo al primo. Si ottiene infatti un funzionale con coefficienti tutti negativi e si pretende che assuma un valore strettamente negativo quando le sue variabili (x_2 ed x_3) sono entrambe non-positive.

CONSIGLI SU COME PREPARARSI ALL'ESAME

Per conseguire un voto per l'insegnamento di Ricerca Operativa devi partecipare ad un appello di esame.
Il primo appello d'esame di ogni anno accademico ha luogo a giugno, dopo la conclusione del corso.
L'esame è scritto, dura circa 4 ore ed ha luogo in aula delta, dove, specie in estate, l'ambiente può risultare

freddo. Consiglio di portarsi golfini, snack, acqua e matite o pennarelli colorati. (E dovete portare il tesserino col vostro numero di matricola.) Chi avesse problemi con l'aria condizionata è pregato di segnalarlo.
L'esame presenta diverse tipologie di esercizi e domande su vari aspetti di quanto esposto a lezione. Nel prepararti all'esame, prendi a riferimento i testi e le correzioni dei temi precedenti come scaricabili al sito del corso:

`http://profs.sci.univr.it/ rrizzi/classes/RO/index.html`

Ogni esercizio è anche un'opportunità di apprendimento e di allenamento, usa pertanto il tuo senso critico per farne miglior uso senza sprecarlo. Una volta letto il testo di un esercizio, ti conviene sfruttarlo innanzitutto per testare la tua preparazione all'esame. Consigliamo pertanto di svolgere l'esercizio quantomeno nella propria mente, e comunque, su una buona percentuale di casi, anche materialmente (e prestando attenzione ai tempi impiegati ed ai punti conseguiti). Solo a valle di un'esperienza almeno parziale con l'esercizio, passa alla lettura della correzione. Se non sai come affrontare l'esercizio, sbircia sí la correzione, ma cercando di utilizzarla solo come suggerimento, cercando di riacquisire quanto prima autonomia nella conduzione dell'esercizio.

E una volta completato l'esercizio? Beh, a questo punto vale il converso: anche se ti sembra di avere svolto pienamente l'esercizio, omettere la successiva lettura della correzione, se fatto sistematicamente, rischia di rivelarsi una grave ingenuità. Il workflow standard cui riferirsi *cum granu salis* dovrebbe essere il seguente: esegui autonomamente l'esercizio e confronta poi le tue risposte con quelle nel rispettivo documento di correzione. Nel confronto con la correzione proposta, presta attenzione non solo alle risposte in sè, ma anche a come esse vadano efficacemente offerte all'esaminatore/verificatore, ossia alla qualità dei tuoi certificati, alla precisione della tua dialettica, a come ottemperi il contratto implicito nella soluzione di un problema ben caratterizzato. In un certo senso, questo ti consentirà di raggiungere pragmaticamente quella qualità che in molti chiamano impropriamente "ordine", che ha valore e giustamente finisce, volenti o nolenti, per essere riconosciuta in ogni esame della vita. Ordine, ma noi preferiamo chiamarlo "saper rispondere in chiarezza alla consegna" non significa bella calligrafia o descrizioni prolisse, ma cogliere tempestivamente gli elementi salienti, quelli richiesti da contratto più o meno implicito. In questo le competenze che abbiamo messo al centro di questo insegnamento di ricerca operativa potranno renderti più consapevolmente ordinato. Lo scopo del documento di correzione non è tanto quello di spiegare come l'esercizio vada risolto ma piuttosto come le risposte vadano adeguatamente esibite pena il non conseguimento dei punti ad esse associati. È secondo quest'ottica che i documenti con le correzioni sono stati scritti. Preso cura di questo delicato aspetto (chiarire cosa si voglia dallo studente), altri obiettivi che, subordinatamente, cerco di assecondare nella stesura dei documenti di correzione sono semmai: aggiungere domande che arricchiscano l'esperienza di apprendimento offerta dall'esercizio, compendiare con altre considerazioni a latere che non potevano essere richieste allo studente, avanzare proposte di percorso ulteriore, e offrire spiegazioni contestualizzate che non possano essere reperite in altro documento. Infatti, per le tipologie di esercizio classiche, descrizioni curate dei più noti algoritmi risolutori possono essere facilmente reperite altrove (e vi incoraggio ad aiutarmi ad arricchire una tabella di link a tali sorgenti, o anche possiamo curare dispense di compendio a titolo di progetti che possono concorrere al voto).

I punti messi in palio ad ogni tema eccedono significativamente quanto necessario al raggiungimento dei pieni voti, gestitevi quindi per dimostrare le competenze che avete, senza impelagarvi dove avete invece delle lacune. Non mi interessano le vostre mancanze o lacune quanto piuttosto quello che dimostrate di saper fare. Se analizzate i temi di appelli precedenti, osserverete che avete a disposizione un'ampia varietà di modi per raccogliere punti e dimostrare la vostra preparazione. Lo scopo dell'esame sono il riconoscimento e la conferma. Essi sono a loro volta funzionali all'apprendimento. L'utilizzo corretto e pieno dei testi e correzioni rese disponibili ti consentirà di:

1. verificare la tua comprensione degli argomenti trattati e degli algoritmi e metodologie illustrati durante il corso;
2. affinare la tua preparazione ai fini dell'esame, non solo mettendo a punto le tue procedure ed approcci (privati e personali), ma chiarendo inoltre cosa l'esercizio richieda di produrre senza sbavature (ad esempio, a meno che non sia esplicitamente richiesto diversamente, la maggior parte degli esercizi non chiede che lo studente spieghi od illustri come ha risolto un problema, ma solo che fornisca risposte certificate);
3. toccare con mano la portata metodologica del concetto di certificato offertaci dalla complessità computazionale.

Durante l'esame, dovrete lavore per almeno 4 ore a quella che definisco "una prova di cromatografia su carta". Serve per riconoscervi con ragionevole confidenza quanto avete lavorato, appreso, sedimentato. E trasformare questo in una proposta di voto il più congrua possibile. La logica dello svolgimento dell'esame deve essere quella di dimostrare al meglio le competenze acquisite andando con efficienza a raccogliere, dei molti punti messi in palio a vario titolo, quelli che vi risultano più funzionali al concretizzare un buon punteggio. Il punteggio in buona sostanza corrisponde al voto. Contano le risposte corrette, fornite in chiarezza, ed i certificati. Tutto il resto non verrà conteggiato. In questo la struttura dell'esame ribadisce il ruolo metodologico ed ubiquito dei concetti di complessità computazionale propagandati nel corso.

gestione dei voti conseguiti.

I voti dei singoli appelli verrano comunicati e resi disponibili tramite ESSE3. Dal 18 in sù i voti verranno registrati automaticamente a valle di un intervallo di tempo concessovi per eventualmente rifiutare il voto. L'eventuale rifiuto del voto, oppure la sua sospensione (per condurre un progetto atto ad incrementare il voto, oppure perchè lo studente richiede del tempo per pensarci, oppure chiede di poter partecipare ad appello successivo decidendo solo alla fine se consegnare o meno riscrivendo voto precedente) vanno richiesti con una mail. Ovviamente, specie per un progetto, se ne deve parlare anche a voce, ma la mail serve comunque come promemoria e contabilità.

Se hai idee su come migliorare il corso od i suoi materiali proponi un tuo progetto, con esso potrai aggiungere al voto conseguito all'esame.