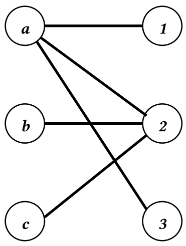
Esame di Ricerca Operativa - 19 febbraio 2025

Facoltà di Scienze MM.FF.NN. - Verona

4 esercizi per 63 punti in palio (voto $\geq$ punti -6, $40 \longrightarrow 30$ *e lode*)

- CORREZIONE -

Esercizio 1 (con 10 richieste: 1+2+1+2+1+1+1+11+3+3 = 26 punti [modellazione/riduzioni]):



In un grafo $G = (V, E)$, chiamiamo:

1. *node cover* ogni $X \subseteq V$ che contenga almeno un estremo di ogni arco $uv \in E$,
2. *matching* ogni $M \subseteq E$ non contenente due archi con un estremo in comune.

Un grafo è detto *bipartito tra* A *e* B se $V = A \cup B$ con $A \cap B = \emptyset$ e ogni arco in E ha un estremo in A e l'altro in B.

Sei importanti problemi modello espressi nel linguaggio dei grafi sono:

1,2- Min (Bipartite) Node Cover: trova un node cover di minima cardinalità in un grafo (bipartito) dato in input.

3,4- Max (Bipartite) Matching: trova un matching di massima cardinalità in un grafo (bipartito) dato in input.

5- Directed s,t-Connectivity: trova un massimo numero di cammini tra due nodi s e t di un generico grafo diretto in input, col vincolo che i cammini siano disgiunti sugli archi (ogni arco compare in al più un cammino).

6- Undirected s,t-Connectivity: trova un massimo numero di cammini disgiunti sugli archi tra due nodi dati di un generico grafo non-diretto.

Richieste dell'Esercizio 1

1.1 (1 pt, problems basic comprehension) con riferimento all'istanza in figura, fornire: un matching ottimo e dire quanti sono, un matching massimale ma non di massima cardinalità e dire quanti sono, un node cover ottimo e dire quanti sono, un node cover minimale ma non ottimo e dire quanti sono.

1.2 (2 pt, model as ILP) Formula come problemi di Programmazione Lineare Intera (PLI) i problemi Max Matching e Min Node Cover per la specifica istanza in figura (1pt) e per grafo generico (1pt).

1.3 (1 pt, model generality) Tra il Max Matching e il Max Bipartite Matching quale dei due modelli è più espressivo? Ossia, quale dei due problemi è più generale e quindi più ambizioso da risolvere?

1.4 (2 pt, classic model knowledge) Descrivere i modelli/problemi del massimo flusso e del minimo taglio in un grafo diretto con capacità intere sugli archi. Quale relazione vi è tra questi due modelli?

1.5 (1 pt, model Directed Connectivity) Modella Directed s,t-Connectivity in termini di Max Flow.

1.6 (1 pt, model Undirected Connectivity 1) Riduci Undirected s,t-Connectivity a Max Flow.

1.7 (1 pt, model Undirected Connectivity 2) Riduci Undirected- a Directed s,t-Connectivity.

1.8 (11 pt, model Bipartite Matching) Modella Max Bipartite Matching in termini di Max Flow. (1pt se spieghi in modo chiaro come produrre un'istanza I_F di Max Flow a partire dalla generica istanza I_M di Max Bipartite Matching), (1pt se è chiaro come da un matching M di I_M si possa produrre un flusso dello stesso valore per I_F), (1pt se chiaro come da un s,t-flow per I_F si ottenga un matching M di I_M dello stesso valore), (1pt se chiaro come da un node cover X di I_M si ottenga un s,t-cut per I_F dello stesso costo), (1pt se chiaro come da un s,t-cut per I_F si ottenga un node cover X

di I_M dello stesso costo), (1pt se dimostri che la cardinalità di nessun node cover può essere inferiore a quella di un matching), (1pt se enunci correttamente il MaxFlow-MinCut Theorem), (1pt se indichi come possa essere dimostrato), (3pt se lo dimostri).
1.9 (3 pt, model flow as ILP) Offri la formulazione PLI del problema Max Flow. Dimostra l'integralità del rilassamento di tale formulazione (2pt).
1.10 (3 pt, explore the dual) Scrivi il duale di tale rilassamento. Scrivi una formulazione di PLI del modello Min Cut che abbia tale duale come suo rilassamento (2pt).

Svolgimento esercizio 1.

Richiesta 1 (1 pt) (goal: problems basic comprehension**).**

Un matching ottimo è l'insieme di due archi $\{a1, b2\}$. L'unico arco che non è in nessun matching ottimo è $a2$, di fatto $\{a2\}$ è un matching massimale non ottimo. Per produrre un matching ottimo si prenda uno qualsiasi degli altri due archi incidenti in a e uno qualsiasi degli altri due archi incidenti in 2. I matching ottimi sono pertanto 4. Il node cover ottimo è uno solo: $\{a, 2\}$.

Richiesta 2 (2 pt) (goal: model as ILP**).**

Si assuma dato un generico grafo $G = (V, E)$, non necessariamente bipartito.

Min Node Cover:

introduciamo una variabile binaria x_v per ogni nodo $v \in V$, con l'idea che $x_v = 1$ se e solo se il nodo v è da includere nel node cover codificato. In pratica x intende essere il vettore caratteristico (o vettore di incidenza) del node cover incognito. Tramite esso possiamo esprimere la seguente formulazione di PLI:

$$\begin{aligned} \min \sum_{v \in V} & x_v \\ & x_u + x_v \geq 1 \quad \text{per ogni } uv \in E \\ & x \in \{0,1\} \quad \text{per ogni } v \in V \end{aligned}$$

Con riferimento al grafo in figura la formulazione che ne risulta è:

$$\begin{aligned} \min\, & x_a + x_b + x_c + x_1 + x_2 + x_3 \\ & x_a + x_1 \geq 1 && \text{(arco } a1\text{)} \\ & x_a + x_2 \geq 1 && \text{(arco } a2\text{)} \\ & x_a + x_3 \geq 1 && \text{(arco } a3\text{)} \\ & x_b + x_2 \geq 1 && \text{(arco } b2\text{)} \\ & x_c + x_2 \geq 1 && \text{(arco } c2\text{)} \\ & x_a, x_b, x_c, x_1, x_2, x_3 \in \{0,1\} \end{aligned}$$

Max Matching

introduciamo una variabile binaria x_{uv} per ogni arco $uv \in E$, con l'idea che $x_{uv} = 1$ se e solo se l'arco uv è da includere nel matching codificato. In pratica x intende essere il vettore caratteristico del matching incognito. Per impostare la famiglia di vincoli necessari, per ogni $v \in V$ denotiamo con $\delta(v)$ l'insieme degli archi incidenti nel nodo v.

$$\max \sum_{uv \in E} x_{uv}$$
$$\sum_{uv \in \delta(v)} x_{uv} \leq 1 \quad \text{per ogni } v \in V$$
$$x_{uv} \in \{0,1\} \quad \text{per ogni } uv \in E$$

Con riferimento al grafo in figura la formulazione che ne risulta è:

$$\begin{aligned}
&\max x_{a1} + x_{a2} + x_{a3} + x_{b2} + x_{c2} \\
&\quad x_{a1} + x_{a2} + x_{a3} \leq 1 && (\text{nodo } a) \\
&\quad x_{b2} \leq 1 && (\text{nodo } b) \\
&\quad x_{c2} \leq 1 && (\text{nodo } c) \\
&\quad x_{a1} \leq 1 && (\text{nodo } 1) \\
&\quad x_{a2} + x_{b2} + x_{c2} \leq 1 && (\text{nodo } 2) \\
&\quad x_{a3} \leq 1 && (\text{nodo } 3) \\
&\quad x_{a1}, x_{a2}, x_{a3}, x_{b2}, x_{c2} \in \{0,1\}
\end{aligned}$$

che, eliminando le ridondanze e i vincoli dominati da altri vincoli, si semplifica a:

$$\begin{aligned}
&\max x_a + x_b + x_c + x_1 + x_2 + x_3 \\
&\quad x_{a1} + x_{a2} + x_{a3} \leq 1 \\
&\quad x_{a2} + x_{b2} + x_{c2} \leq 1 \\
&\quad x_{a1}, x_{a2}, x_{a3}, x_{b2}, x_{c2} \in \{0,1\}
\end{aligned}$$

Richiesta 3 (1 pt) (goal: model generality**).**

Il problema Max Bipartite Matching è la *restrizione* di Max Matching ai soli grafi bipartiti in quanto rifiuta di prendere in input grafi che non sono bipartiti, ma su grafi bipartiti i due problemi chiedono esattamente la stessa cosa. Per questa ragione Max Matching risulta essere un problema più generale e un modello più espressivo.

Richiesta 4 (2 pt) (goal: classic model knowledge**).**

Dato un grafo diretto $G = (V, E)$ con capacità intere sugli archi $c : E \to \mathbb{N}$ e con due nodi speciali $s, t \in V$, il problema Max Flow chiede di trasportare quanto più possibile dal nodo s al nodo t, ossia, introdotta una variabile reale $x_{(u,v)}$ per ogni arco $(u, v) \in E$ si intende massimizzare quanto fuoriesca da s, ovvero il funzionale $\sum_{(s,v) \in E} x_{(s,v)}$, sotto i vincoli di capacità $x_{(u,v)} \leq c_{(u,v)} \forall (u, v) \in E$ e di solenoidalità $\sum_{(u,v) \in E} x_{(u,v)} = \sum_{(v,z) \in E} x_{(v,z)} \forall v \in V$. Tali vincoli garantiscono che $\sum_{(v,t) \in E} x_{(v,t)} = \sum_{(s,v) \in E} x_{(s,v)}$.

Il problema Min Cut chiede di trovare un sottoinsieme E' degli archi a costo $\sum_{(v,z) \in E'} c_{(u,v)}$ minimo tale che ogni cammino da s a t includa almeno un arco in E'.

Richiesta 5 (1 pt) (goal: model Directed Connectivity**).**

L'istanza $(G = (V, E), s, t)$ di Directed s,t-Connectivity è perfettamente rappresentata dall'istanza $(G = (V, E), s, t, c \equiv 1)$ di Max Flow.

Richiesta 6 (1 pt) (goal: model Undirected Connectivity 1**).**

Data l'istanza $(G = (V, E), s, t)$ di Undirected s,t-Connectivity si consideri il grafo diretto $D = (V, A)$, sul medesimo insieme di nodi, e con $A = \{(u, v), (v, u) \mid uv \in E\}$. L'istanza $(G = (V, E), s, t)$

di Undirected s,t-Connectivity è perfettamente rappresentata dall'istanza $(D = (V, A), s, t, c \equiv 1)$ di Max Flow.

Richiesta 7 (1 pt) (goal: model Undirected Connectivity 2).

Data l'istanza $(G = (V, E), s, t)$ di Undirected s,t-Connectivity si consideri il grafo diretto $D = (V, A)$, sul medesimo insieme di nodi, e con $A = \{(u, v), (v, u) \mid uv \in E\}$. L'istanza $(G = (V, E), s, t)$ di Undirected s,t-Connectivity è perfettamente rappresentata dall'istanza $(D = (V, A), s, t)$ di Directed s,t-Connectivity.

Richiesta 8 (11 pt) (goal: model Bipartite Matching).

Dato un grafo bipartito $G = (U \cup V, E)$ come istanza di Max Bipartite Matching, introduco due nuovi nodi s e t e considero il grafo diretto $D = (U \cup V \cup \{s, t\}, A)$ dove $A = \{(s, u) \mid u \in U\} \cup \{(v, t) \mid v \in V\} \cup \{(u, v) \mid u \in U, v \in V, uv \in E\}$ e la funzione $c : A \to \mathbb{N}$ definita da $c \equiv 1$. Affermo, e di seguito verificheremo seguendo i successivi punti della richiesta, che il massimo valore di un flusso per (D, s, t, c) è pari alla massima cerdinalità di un matching nel grafo bipartito G. In primo luogo, ove M sia un qualsiasi matching di G posso produrre $|M|$ cammini da s a t in D come segue: per ciascun $uvinM$, con $uinU$ e $vinV$, si consideri il cammino $(s, u), (u, v), (v, t)$. Si noti che nessun arco di D è utilizzato da più di uno di questi cammini, dato che M era un matching e, siccome ogni arco di D ha capacità almeno 1 questo significa che in D possiamo produrre un s, t-flusso di valore almeno $|M|$. Di converso, ogni s, t-flusso di valore φ in D può essere decomposto in cammini unitari e ciascuno di questi cammini deve necessariamente attraversa un arco con coda in U e testa in V, ossia un arco che sostanzialmente era presente anche in G. Poichè D è di fatto un DAG è facile vedere che tutti questi cammini hanno lunghezza 3. Devono inoltre essere disgiunti sui nodi visto che il loro prime e i loro ultimo arco ha capacità 1. Pertanto gli φ archi centrali di questi cammini corrispondono ad un matching di M. Questi due primi punti già dimostrano la correttezza della riduzione, gli altri punti possono essere facilmente dimostrati con analogo rigore.

Richiesta 9 (3 pt) (goal: model flow as ILP).

Per ogni arco (u, v) del grafo diretto $D = (V, A)$ in cui si richieda di trovare un flusso massimo introduciamo una variabile $f_{(u,v)}$ che stia ad indicare quanto flusso potremmo inviare lungo di esso. La formulazione di PLI (ma, vedremo subito, anche di PL) è:

$$\max \sum_{(s,v) \in \delta^+(s)} f_{(s,v)}$$

$$\sum_{(u,x) \in \delta^-(x)} f_{(u,x)} - \sum_{(x,v) \in \delta^+(x)} f_{(u,x)} = 0 \text{ per ogni } x \in V \setminus \{s, t\}$$

$$f_{(u,v)} \leq c_{(u,v)} \text{ per ogni } (u, v) \in A$$

$$f_{(u,v)} \geq 0 \text{ per ogni } (u, v) \in A \text{ (oppure anche } f_{(u,v)} \in \mathbb{N})$$

Che non vi sia sostanziale differenza tra le formulazioni di PLI e di PL combinate quì sopra disegue dal vatto che quando le $f_{(u,v)}$ vengano riguardate come variabili reali piuttosto che non intere lo spazio delle soluzioni ammissibili diviene un politopo intero, ossia un politopo di cui tutti i vertici hanno tutte le coordinate intere. Infatti, in una soluzione frazionaria estremale, dove per nessuna variabile sarà possibile sia alzare che abbassare il valore di quella sola variabile senza perdere in ammissibilità, ogni arco il cui valore è frazionario dovrà necessariamente essere adiacente ed un altro arco di valore frazionario, considerato che capacità sono intere. Si consideri quindi il sottografo ricoprente che contenga i soli archi a valore frazionario. Se esso contiene un ciclo C allora esiste un ε tale che la soluzione considerata è la media aritmetica tra le due soluzioni ottenute alzando oppure abbassando di ε i valori su ciascun arco di C. Questa operazione che per come descritta assume che gli archi di C siano tutti diretti in uno stesso verso ha in realtà una versione solo un attimo più complicata del caso in C gli intervalli di archi orientati in senso orario si alternion ad intervalli di archi orientati in senso opporto (dove cambia la direzione si deve cambiare anche il segno della variazione di entitaà ε). Quindi il grafo dei soli archi frazionari è aciclico (ossia una foresta), e possiamo concentraci su una sua qualsiasi componente connsessa, che sarà un albero. Ogni albero con almeno un arco ha almeno due fuglie a e b e contiene un unico cammino $P_{a,b}$ tra a e b. L'idea resta quella di produrre due soluzionio ammissibile di cui quella in considerazione è la media aritmetica (questo significa che la soluzione considerata non può essere un vertice) con lo stesso giochetto che in una soluzione il primo arco venga incrementato di ε (per un qualche ε) e nell'altra venga decrementato di ε; poi su ciascun altro arco si opera la stessa modifica che per l'arco precedente oppure la modifica opposta (se il verso è opposto che per l'arco precedente).

Richiesta 10 (3 pt) (goal: explore the dual**).**

Seguendo pedissequamente le regole, il duale del rilassato, ossia di:

$$\begin{aligned}
\max &\sum_{(s,v)\in\delta^+(s)} f_{(s,v)} \\
&\sum_{(u,x)\in\delta^-(x)} f_{(u,x)} - \sum_{(x,v)\in\delta^+(x)} f_{(u,x)} = 0 \quad \text{per ogni } x \in V \setminus \{s,t\} \\
&f_{(u,v)} \le c_{(u,v)} \quad \text{per ogni } (u,v) \in A \\
&f_{(u,v)} \ge 0 \quad \text{per ogni } (u,v) \in A
\end{aligned}$$

richiederebbe un attimo di piccole accortezze (nulla di speciale, ma semplifichiamo) per essere composto e rimaneggiato per favorirne l'interpretazione. Meglio esercitare in bella partenza un piccolo trucco che riduca al minimo ogni anomalia: buttiamo via ogni arco (t,s) eventualmente già presente (tanto nessun s,t-flusso ottimo ne farà uso) e inseriamo quindi un unico arco (t,s) di capacità infinita; chiamiamo D' il nuovo digrafo così ottenuto da D. Se ora imponiamo il vincolo di solenoidalità (bilanciamento perfetto tra quanto entra e quanto esce) anche per i nodi s e t che prima ne erano esenti, scopriamo che vi è una corrispndenza biunivoca tra questo nuovo tipo di flussi (chiamati circolazioni) nel grafo aumentato e i flussi nel grafo originario che di fatto ne sono la proiezione (si ottengono cioè semplicemente ignorando il valore del flusso sull'arco aggiunto). In praticolare, gli s,t-flussi di massimo (o di pari) valore nell grafo originario corrispondono circolazioni che presentano un massimo (o pari) valore del funzionale $f_{(t,s)}$. Dualizzeremo pertanto:

$$\begin{aligned}
\max\, & f_{(t,s)} \\
&\sum_{(u,x)\in(\delta^-)_{D'}(x)} f_{(u,x)} - \sum_{(x,v)\in(\delta^+)_{D'}(x)} f_{(u,x)} = 0 \quad \text{per ogni } x \in V \\
&f_{(u,v)} \le c_{(u,v)} \quad \text{per ogni } (u,v) \in A \\
&f_{(u,v)} \ge 0 \quad \text{per ogni } (u,v) \in A \cup \{(t,s)\}
\end{aligned}$$

Dualizzare questa formulazione viene più facile in quanto ogni cosa è uniforme, ci basta seguire una stessa regola. In pratica, associate (come moltiplicatori) ai vincoli di eguaglianza, abbiamo una variabile free y_v per ogni nodo $v \in V$. E, associate ai vincoli di capacità, abbiamo una variabile non-negativa $z_{(u,v)}$ per ogni arco $(u,v) \in A \cup \{(t,s)\}$. Otteniamo

$$\begin{aligned}
\min & \sum_{(u,v)\in A\cup\{(t,s)\}} c_{(u,v)} z_{(u,v)} \\
& z_{(t,s)} + y_s - y_t \geq 1 \\
& z_{(u,v)} + y_v - y_u \geq 0 \quad \text{per ogni } (u,v) \in A \\
& z_{(u,v)} \geq 0 \quad \text{per ogni } (u,v) \in A
\end{aligned}$$

Si noti che se in una soluzione l'ammissibilità di una soluzione è ivariante allo schiftare di uno stesso δ i valori di tutte le variabili $y_v, v \in V$. Ne consegue che possiamo fissare arbitrariamente il valore di una di queste, optiamo per fissare $y_t = 0$, che equivale a proiettarla via.

L'interpretazione che diamo a quanto si ottiene è: la variabile y_s è sollevata a quota 1 da un pavimento al di sopra del quale ogni altra variabile y deve stare per i vincoli di non-negatività, con la y_t che poggia sul pavimento. Dopodichè pet ogni un arco (u,v) vale che $z_{(u,v)}$, che è sempre non-negativa e ci costa $c_{(u,v)}$, deve valere almeno quanto il dislivello $y_u - y_v$. Il fatto che s è sollevato di 1 mentre t è a terra ci obbliga a pagare su almeno un arco di ogni cammino che vada da (s,t). Ci si accende la lampadina che un s,t-talio si caratterizza proprio per essere un'insieme di archi che interseca ogni tale cammino.

Il duale formula pertanto il rilassamento continuo di una qualche formulazione combinatorica del problema dell's,t-taglio di costo minimo.

Il max-flow min-cut theorem afferma che in ogni istanza il valore massimo di un s,t-flusso eguaglia il minimo costo di un s,t-taglio. Avendo già dimostrato l'interessa del politopo del rilassamento continuo della formulazione combinatorica del flusso, un modo per dimostrare tale teorema è dimostrare con ragionamenti analoghi che anche il politopo associato al duale è intero.

Un altro modo per dimostrarlo è considerando un algoritmo primale come quello di Ford e Fulkerson, osservare che deve sempre terminare, e osservare che a terminazione resta individuato un s,t -taglio dello stesso valore dell's,t-flusso corrente. Ovviamente ci si stà avvalendo anche del lemma della dualità debole, ossia della consapevlezza che il valore di nessun s,t-flusso può avere valore maggiore al costo di nessun s,t-taglio.

Esercizio 2 (con 8 richieste: 1+1+1+1+1+2+1+2 = 10 punti [programmazione dinamica]):

Un robot, inizialmente situato nella cella **A−1**, deve portarsi nella sua home, nella cella **I−10**.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
A	0	1	1	0	1	1	0	0	•	6
B	2	•	1	•	0	0	•	0	0	5
C	0	•	0	•	•	0	0	1	1	4
D	0	0	1	0	0	0	1	•	0	3
E	0	0	•	1	0	1	2	0	0	1
F	0	1	3	1	•	3	1	•	0	1
G	3	•	2	1	2	•	•	3	1	•
H	2	1	2	•	•	1	1	1	•	0
I	4	4	3	3	2	1	1	•	0	0

I movimenti base consentiti da ogni cella sono il passo verso destra (ad esempio dalla cella A−3 alla cella A−4) o il passo verso in basso (ad esempio dalla cella A−3 alla cella B−3) e il passo diagonale (ad esempio dalla cella A–3 alla cella B–4). Se il robot deve evitare le celle proibite (●), quanti sono i percorsi ammissibili? Inoltre, se in ogni cella permessa si incontra un pedaggio del valore riportato nella cella stessa, sapresti minimizzare la somma dei numeri che appaiono lungo il suo percorso?

Richieste dell'Esercizio 2

2.1 (1 pt, numero percorsi) Numero di percorsi ammissibili da **A–1** a **I–10**
2.2 (1 pt, num percorsi da B–3) Numero di percorsi ammissibili da **B–3** a **I–10**
2.3 (1 pt, num percorsi a F–6) Numero di percorsi ammissibili da **A–1** a **F–6**
2.4 (1 pt, num percorsi per D–5) Numero di percorsi da **A–1** a **I–10** passanti per **D–5**
2.5 (1 pt, opt val) Minimo totale di pedaggi su un cammino da **A–1** a **I–10**. (E soluzione di tale valore).
2.6 (2 pt, numero cammini ottimi) Numero cammini ottimi da **A–1** a **I–10**
2.7 (1 pt, opt val per D–5) Minimo totale di pedaggi su un cammino da **A–1** a **I–10** passante per **D–5**
2.8 (2 pt, num paths of opt val via D–5) Numero cammini ottimi da **A–1** a **I–10** passanti per **D–5**

Svolgimento esercizio 2.

La risposta alle prime due domande può essere reperita nella rispettiva cella della tabella **num cammini da**, dove in ogni cella C, partendo da quelle in basso a destra, si é computato il numero di percorsi che vanno dalla cella C alla cella I–10.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
A	675	489	361	233	161	49	9	5	0	0
B	186	0	128	0	72	40	0	4	1	0
C	186	0	128	0	0	32	8	2	1	0
D	107	79	70	58	36	18	6	0	1	0
E	19	9	0	12	10	8	4	2	1	0
F	5	5	4	2	0	2	2	0	1	0
G	0	0	1	1	1	0	0	2	1	0
H	0	0	0	0	0	1	1	1	0	1
I	0	0	0	0	0	0	0	0	1	1

Tabella 2: **num cammini da**

Per rispondere alle domande successive serve anche la tabella **num cammini a**, dove in ogni cella C, partendo da quelle in alto a sinistra, è riportato il numero di percorsi che vanno dalla cella A–1 alla cella C.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
A	1	1	1	1	1	1	1	1	0	0
B	1	0	2	0	2	4	0	2	3	3
C	1	0	2	0	0	6	10	12	17	23
D	1	2	4	6	6	12	28	0	29	69
E	1	4	0	10	22	40	80	108	137	235
F	1	6	10	20	0	62	182	0	245	617
G	1	0	16	46	66	0	0	182	427	0
H	1	2	18	0	0	66	66	248	0	427
I	1	4	24	42	42	108	240	0	248	675

Tabella 3: **num cammini a**

Ritrovare il valore 675 ci conforta, forse non abbiamo introdotto errori di calcolo nel computo delle due tabelle. La risposta alla terza domanda è contenuta nella rispettiva cella di questa seconda tabella.

La quarta domanda richiede di combinare le informazioni provenienti dalle due tabelle: la risposta è ottenuta come prodotto dei due valori riportati nelle due tabelle entro la cella di passaggio obbligato per il robot.

Per rispondere alle prossime due domande compilo un'ulteriore tabella, dove in ogni cella C, partendo da quelle in basso a destra, si computa il minimo costo di un percorso che va dalla cella C alla cella I–10. Computiamo inoltre e riportiamo in piccolo, per ogni cella C, il numero di percorsi di tale valore ottimo.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
A	14^{10}	14^4	13^4	11^2	11^2	10^2	3^2	3^1	-1^0	-1^0
B	14^6	-1^0	12^4	-1^0	9^4	9^2	-1^0	3^1	2^1	-1^0
C	12^6	-1^0	11^4	-1^0	-1^0	9^2	8^2	3^1	2^1	-1^0
D	12^5	12^1	11^4	10^2	9^6	9^2	8^2	-1^0	1^1	-1^0
E	12^3	12^1	-1^0	10^2	9^2	9^2	7^2	1^2	1^1	-1^0
F	12^1	12^1	11^1	7^1	-1^0	8^2	5^2	-1^0	1^1	-1^0
G	-1^0	-1^0	8^1	6^1	5^1	-1^0	-1^0	4^2	1^1	-1^0
H	-1^0	-1^0	-1^0	-1^0	-1^0	3^1	2^1	1^1	-1^0	0^1
I	-1^0	-1^0	-1^0	-1^0	-1^0	-1^0	-1^0	-1^0	0^1	0^1

Tabella 4: **num cammini da**

Leggendo i valori riportati nella cella A–1 scopriamo che il minimo costo di una traversata é di 14, e che esistono 10 diversi percorsi ammissibili che totalizzano questo valore.

Per rispondere alle ulteriori domande compilo un'ulteriore tabella, dove in ogni cella C, partendo da quelle in alto a sinistra, si computa il minimo costo di un percorso che va dalla cella A–1 alla cella C. Computiamo inoltre e riportiamo in piccolo, per ogni cella C, il numero di percorsi di tale valore ottimo.

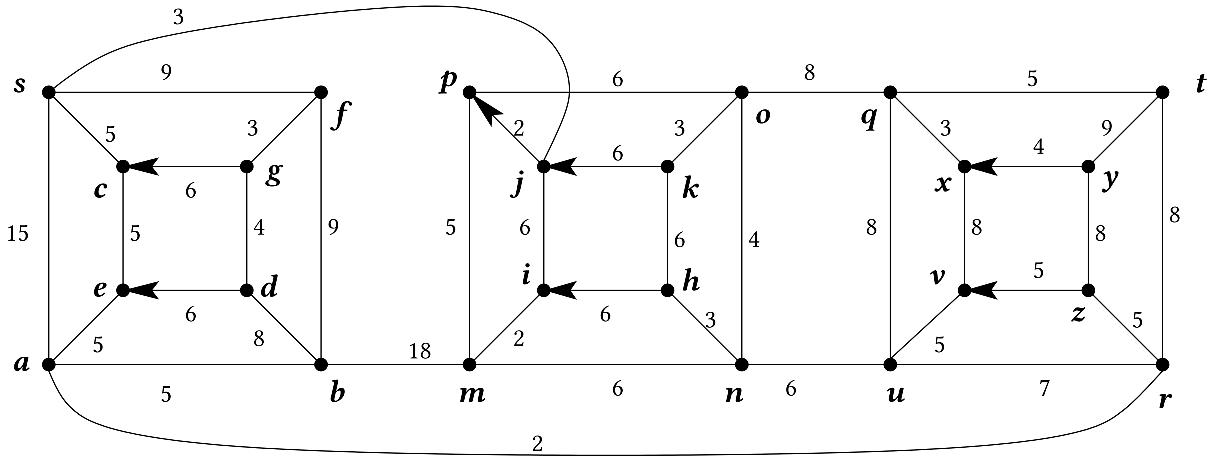
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
A	0^1	1^1	2^1	2^1	3^1	4^1	4^1	4^1	-1^0	-1^0
B	2^1	-1^0	3^1	-1^0	3^1	4^1	-1^0	4^2	4^3	9^3
C	2^1	-1^0	3^1	-1^0	-1^0	4^1	4^2	5^4	6^4	13^3
D	2^1	2^2	4^1	4^1	4^1	4^2	5^5	-1^0	6^4	16^3
E	2^1	2^4	-1^0	5^2	5^2	6^2	8^2	8^2	8^2	17^3
F	2^1	3^6	6^6	7^6	-1^0	9^2	10^2	-1^0	8^4	18^3
G	5^1	-1^0	8^6	9^6	11^6	-1^0	-1^0	13^2	14^2	-1^0
H	7^1	8^1	10^7	-1^0	-1^0	12^6	13^6	14^8	-1^0	14^2
I	11^1	15^1	18^1	21^1	23^1	24^1	25^1	-1^0	14^8	14^{10}

Tabella 5: **num cammini a**

Avendo riempito l'intera tabella (non serviva per solo rispondere alle ultime due domande), nella cella I–10 troviamo conferma che il minimo costo raccoglibile lungo una traversata é di 14, e che esistono 10 diversi percorsi ammissibili che totalizzano questo valore. Le risposte alle ulteriori domande sono ottenute consultando queste due tabelle, eventualmente entrambe: nel caso di celle di passaggio obbligato il valore ottimo andrà ottenuto tramite somma (avendo cura di non conteggiare due volte il valore della cella di passaggio) mentre il numero di soluzioni ottime sarà il prodotto dei due numeri riportati in piccolo nelle due tabelle entro la cella di passaggio obbligato per il robot.
Riportiamo quindi i risultati finali.

consegna	num. percorsi	opt	una sol opt
A–1 $\rightarrow$ I–10	675		
B–3 $\rightarrow$ I–10	128		
A–1 $\rightarrow$ F–6	62		
passaggio per D–5	$6 * 36 = 216$		
minimo costo		14	$A1\text{-}A2\text{-}A3\text{-}B3\text{-}C3\text{-}D3\text{-}D4\text{-}E4\text{-}E5\text{-}E6\text{-}F6\text{-}F7\text{-}G8\text{-}G9\text{-}H10\text{-}I10$
n. min-cost paths	10		
min-cost D–5-path		$13 = 4 + 9$	A1-B2-A3-B3-C3-D3-D4-D5-E6-F6-F7-G8-H8-I9-I10
n. min-cost D–5-paths	$1 * 6 = 6$		

Esercizio 3 (con 11 richieste: 3+2+2+2+2+2+1+4+3+4+2 = 27 punti [grafi]):



Richieste dell'Esercizio 3

3.1 (3 pt, recognize planarity) Dire, certificandolo, se siano planari o meno il grafo G e il grafo G' ottenuto da G sostituendo l'arco ar con un arco ap.

3.2 (2 pt, recognize 2-colorability) Dire, certificandolo, quale sia il minimo numero di archi da rimuovere per rendere bipartiti i grafi G e G' (1 punto per ogni soluzione certificata da bicolorazione e 1 per ogni certificato di ottimalità, ove i grafi non siano bipartiti di loro).

3.3 (2 pt, max flow) In G, trovare un massimo flusso dal nodo s al nodo t.

3.4 (2 pt, min cut) Certificare l'ottimalità di tale flusso massimo.

3.5 (2 pt, flow sensitivity) Per quali archi un incremento della capacità dell'arco modifica il massimo valore di flusso? Specificare il massimo incremento ottenibile agendo su ciascun singolo arco.

3.6 (2 pt, certify flow sensitivity) Scegli uno qualsiasi degli archi per cui il valore di incremento che hai fornito al punto precedente è massimo ed esibisci prova che rilassandone la capacità si possa ottenere quel valore di flusso (1pt). Certifica anche che l'aumento non è superiore a quanto dichiarato (1pt).

3.7 (1 pt, one MST) In G, fornire un albero ricoprente di peso minimo.

3.8 (4 pt, count MSTs) Quanti sono gli MST in G?

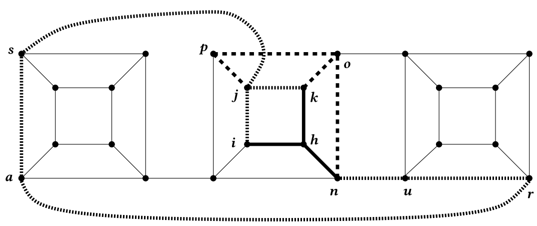
3.9 (3 pt, MST categorize edges) Etichetta ciascun arco con la lettera A se appartiene a ogni MST, B se a nessuno, C altrimenti. (Se li hai ti conviene usare 3 colori.)

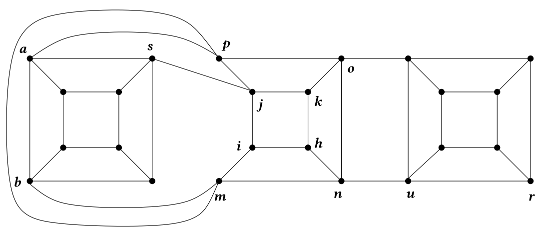
3.10 (4 pt, MST certificates) Per ciascuno dei quattro archi incidenti in u certificare l'etichetta assegnatagli al punto precedente.

3.11 (2 pt, max match) Fornire un matching di massima cardinalità in G (1pt). Sapresti dire perche non possa esserci un matching con un numero maggiore di archi? (1pt)

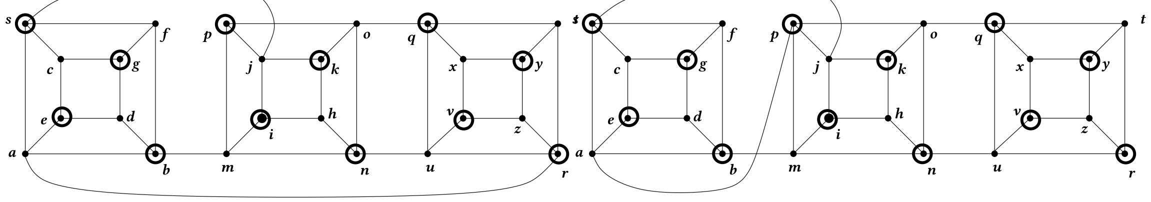
Svolgimento esercizio 3.

La non-planarità di G' è certificata dalla $K_{3,3}$ subdivision in figura, sulla sinistra. Sulla destra, si offre un planar embedding di G', che ne certifica invece la planarità.





I due grafi sono entrambi bipartiti come di fatto certificato dalla medesima 2-colorazione dei nodi.

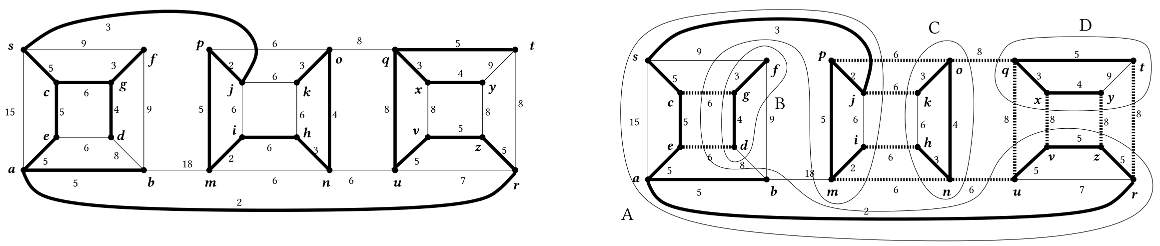


La figura quì sotto a sinistra visualizza un MST in linea spessa. Alla sua immediata destra classifichiamo gli archi di G in tre categorie:

linea spessa continua quelli che appartengono ad ogni MST

linea spessa tratteggiata quelli che appartengono a qualche MST ma non a tutti

linea sottile quelli che non appartengono ad alcun MST



Gli archi in linea spessa raccolgono i nodi in 4 isole A, B, C e D.

Ogni MST dovrà prendere precisamente un arco del taglio che separa la componente D (ogni taglio deve essere superato ma non si vuole prendere più di un arco di peso 8). Questa scelta potrà essere fatta in 5 modi diversi, dopo aver collegato tra loro le 3 componenti A, B e C con $3 - 1 = 2$ archi di peso 6. Ci sono 3 diversi modi di massima nel fare questo:

evitare archi tra A e B siccome tra B e C ci sono 4 archi di peso 6 mentre tra A e C c'è un sono arco di peso 6 ho $4 \times 1 = 4$ possibilità.

evitare archi tra B e C siccome tra A e B ci sono 2 archi di peso 6 (quelli di peso maggiore non ci interessano) mentre tra A e C c'è un sono arco di peso 6 ho $2 \times 1 = 2$ possibilità.

evitare archi tra A e C siccome tra A e B ci sono 2 archi di peso 6 (quelli di peso maggiore non ci interessano) mentre tra B e C ci sono 4 archi di peso 6 ho $2 \times 4 = 8$ possibilità.

Pertanto, per collegare tra loro le 3 componenti ho $4 + 2 + 8 = 14$ modi diversi e il numero degli MST è $14 \times 5 = 70$.

Forniamo ora dei certificati specifici per la classificazione dei 4 archi per cui richiesto:

arco nu in qualche soluzione ottima in quanto arco di peso minimo del taglio che separa la componente A dal resto del grafo. Non in tutte in quanto arco di peso massimo nel ciclo $nurasfbm$.

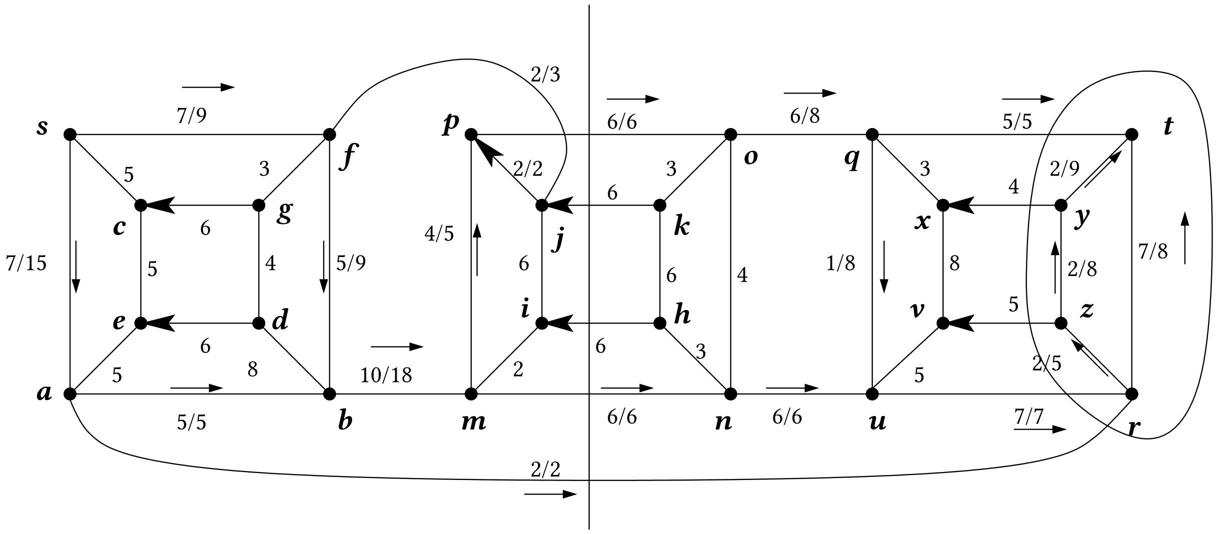
arco qu in qualche MST in quanto arco di peso minimo del taglio che separa la componente D dal resto del grafo. Non in tutte in quanto arco di peso massimo nel ciclo $quno$.

arco qu in qualche MST in quanto arco di peso minimo del taglio che separa la componente D dal resto del grafo. Non in tutte in quanto arco di peso massimo nel ciclo $quno$.

arco vu in tutti gli MST in quanto arco di peso strettamente minimo del taglio che separa il nodo u dal resto del grafo.

arco ru in nessun MST in quanto arco di peso massimo nel ciclo $ruvz$.

Un flusso ottimo (valore 14) è visualizzato in figura coi due tagli minimi: il taglio di spiaggia $\{t, r, z, y\}$ e quello che separa nel mezzo il lato desto da quello sinistro della figura. Ciascuno dei due tagli certifica l'ottimalità del flusso, come di ogni altro possibile flusso di valore 14.



,

Si noti che i due tagli hanno in comune il solo arco ar.

Siccome l'arco ar è l'unico ad appartenere a tutti i certificati di ottimalità allora ar è anche l'unico arco il cui aumento di capacità comporta un aumento nel valore del massimo flusso.

Potendo aumentare la capacità dell'arco ar diventa possibile instradare 4 ulteriori unità di flusso da s a t: una lungo il cammino $sart$ e le altre tre lungo il cammino $sarzyt$. L'ottimalità del flusso di valore 17 che ne risulta è certificata dal taglio di spiaggia $\{t, y, z\}$.

Esercizio 4 (con 1 richieste: 0 = 0 punti [simplesso]):

Nelle intenzioni era presente anche un esercizio da 12 punti sul simplesso ed analisi di sensitività. Ma all'esame abbiamo omesso di distribuire i fogli relativi a tale esercizio.

Richieste dell'Esercizio 4

4.1 (0 pt,)

Svolgimento esercizio 4.

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. Per gli appelli estivi in aula delta, non abbiamo controllo dell'aria condizionata e l'ambiente potrà risultarvi troppo freddo. Data la durata dell'appello consiglio di portarsi golfini, snack, acqua e matite o pennarelli colorati. Potete portarvi materiali cartacei ma non è consentita alcuna strumentazione elettronica. Dovete portare il tesserino col vostro numero di matricola.

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 la 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: cercate e concretizzate quelli che più vi convengono, non impegolatevi a dimostrare quello che non sapete o dove incontrate incertezze. Contano le risposte corrette, fornite in chiarezza, ed i certificati (in questo la struttura dell'esame ribadisce il ruolo metodologico ubiquito dei concetti di complessità computazionale propagandati nel corso). Tutto il resto (incluse le castronerie colossali ma anche le doppie risposte discordanti) non verrà conteggiato. Ricordate che in buona sostanza il voto corrisponderà al punteggio positivamente raccolto. 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 ci interessano le vostre mancanze o lacune quanto piuttosto quello che dimostrate di saper fare.

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 che trovi 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, sfruttalo al meglio senza sprecarlo. Una prima utilità è quella di testare la tua preparazione all'esame. Dopo aver letto il testo, consigliamo pertanto di svolgere l'esercizio quantomeno nella propria mente. Ma, in sufficiente numero di esemplari, poi anche materialmente, prestando attenzione ai tempi impiegati ed ai punti conseguiti. Solo a valle di un'esperienza almeno parziale con l'esercizio, passa alla lettura del documento di correzione. Se non sai come affrontare l'esercizio, sbircia sì la correzione, ma cercando di utilizzarla solo come suggerimento, cercando di riacquisire quanto prima autonomia nella conduzione dell'esercizio.

E se invece ti sembra di saper risolvere del tutto l'esercizio? Beh, a questo punto vale il converso: controlla che quanto hai in mente come soluzione corrisponda a quanto considerato e proposto come svolgimento opportuno. 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 mancato conseguimento dei punti ad esse associati. Aggiungo che per le tipologie di esercizio classiche, descrizioni curate dei più noti algoritmi risolutori possono essere facilmente reperite altrove (perchè non collaborare a raccogliere una ricca collezione di link a tali sorgenti?).