OTTIMIZZAZIONE COMBINATORIA
Soluzioni della prova scritta del 6 febbraio 2018

1. E' dato un insieme P di n programmi, e si conosce l'insieme L_k delle librerie che occorre caricare per eseguire ciascun programma $k \in P$. Una volta caricata, la generica libreria i impegna risorse per una quantità c_i. Per metterla a disposizione di tutti i programmi che la richiedono è sufficiente caricarne una copia.

GRUPPO A: Formulate come PL 01 il problema di scegliere almeno m programmi in modo da minimizzare la quantità complessiva di risorse impegnate da tutte le librerie che questi richiedono.

GRUPPO B: Formulate come PL 01 il problema di massimizzare il numero di programmi eseguibili caricando le librerie richieste per il loro funzionamento in modo che la somma delle risorse da esse impegnate non superi un dato valore m.

La decisione da prendere prevede la scelta di un sottoinsieme S dei programmi di P. Indichiamo quindi con $\mathbf{x}$ il vettore caratteristico di S, ponendo $x_k = 1$ se $k \in S$ e $x_k = 0$ altrimenti. A questo punto dobbiamo esprimere le risorse richieste dai programmi di S, il cui ammontare dipende dalle librerie che essi richiedono. Indichiamo perciò con $\mathbf{y}$ il vettore caratteristico delle librerie richieste, ponendo $y_i = 1$ se S contiene uno o più programmi che richiedono la libreria i, e $y_i = 0$ altrimenti. Evidentemente il vettore $\mathbf{y}$ è legato al vettore $\mathbf{x}$ dalle seguenti relazioni

$$x_k \le y_i \qquad \text{per ogni } k \in P \text{ e } i \in L_k$$

L'ammontare delle risorse richieste dai programmi di S è perciò dato dal prodotto scalare di y per il vettore c delle risorse richieste da ciascuna libreria. Per il GRUPPO A l'obiettivo si formula dunque

$$\min \quad \sum_{i \in L} c_i y_i$$

soggetto all'ulteriore vincolo sul numero minimo di programmi che si vogliono eseguire

$$\sum_{k \in P} x_k \ge m$$

Per il GRUPPO B l'obiettivo si formula invece

$$\max \quad \sum_{k \in P} x_k$$

soggetto all'ulteriore vincolo sul tetto all'ammontare di risorse utilizzate

$$\sum_{i \in L} c_i y_i \le m$$

2. Descrivete il Problema 1 mediante un opportuno grafo G definito sugli insiemi P dei programmi e L delle librerie. Che proprietà possiede?

GRUPPO A: Complementate le variabili di decisione 01 associate, nella soluzione del Problema 1, alla scelta dei programmi. Che proprietà ha l'insieme definito sul grafo G da queste variabili complementate e da quelle associate alle librerie selezionate?

GRUPPO B: Complementate le variabili di decisione 01 associate, nella soluzione del Problema 1, alla scelta delle librerie. Che proprietà ha l'insieme definito sul grafo G da queste variabili complementate e da quelle associate alle librerie selezionate?

Il grafo G ha $P \cup L$ come insieme di vertici, e un arco per ogni coppia (k, i) con $k \in P$ e $i \in L_k$. G ha la proprietà di essere bipartito, e quindi non contiene cicli dispari.

GRUPPO A: Complementando le variabili x_k associate ai programmi $k \in P$ e ponendo quindi $x_k = 1 - z_k$ il vincolo sul vettore $\mathbf{y}$ si scrive $1 - z_k \le y_i$, cioè

$$y_i + z_k \ge 1 \qquad \text{per ogni } k \in P \text{ e } i \in L_k$$

Questo vincolo impone di selezionare almeno un nodo fra k e i per ogni arco (k, i). Le variabili definiscono quindi un insieme trasversale di G.

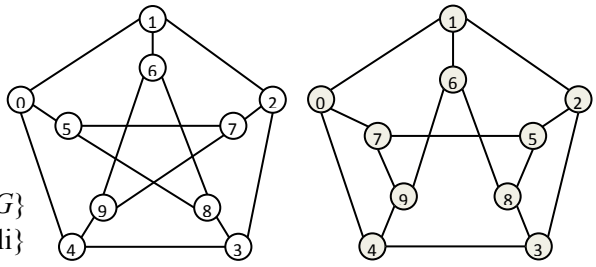
GRUPPO B: Complementando le variabili y_i associate alle librerie $i \in L$ e ponendo quindi $y_i = 1 - z_i$ il vincolo sul vettore **y** si scrive $x_k \leq 1 - z_i$, cioè

$$z_i + x_k \leq 1 \qquad \text{per ogni } k \in P \text{ e } i \in L_k$$

Questo vincolo impone di selezionare non più di un nodo fra k e i per ogni arco (k, i). Le variabili definiscono quindi un insieme stabile di G.

3. Considerate il grafo $G = (V, E)$ di figura (GRUPPO A a sinistra). Ricorrendo alle nozioni di subclusione e scambio, spiegate per quale motivo un problema di ottimizzazione combinatoria nel quale l'insieme universo U coincide con V oppure E e la famiglia delle soluzioni è definita da

$\Im$ = {insiemi stabili di G}
$\Im$ = {insiemi trasversali di G}
$\Im$ = {matching di G}
$\Im$ = {edge-cover di G}
$\Im$ = {clique di G}
$\Im$ = {insiemi di archi che formano cammini di G}
$\Im$ = {insiemi di archi di G che non formano cicli}



è oppure non è risolto all'ottimo dall'Algoritmo Greedy per qualsiasi funzione peso c.

L'Algoritmo Greedy non risolve in generale all'ottimo i problemi di massimo insieme stabile, minimo insieme trasversale, massimo matching, minimo edge cover, massimo cammino. In tutti questi casi esistono infatti due insiemi massimali (minimali) che non sono massimi (minimi).

GRUPPO A. Per l'insieme stabile, S = {2, 5, 8, 9}, S' = {2, 5, 6}. Per il trasversale, i complementi di S, S'. Per il matching, M = {05, 16, 27, 38, 49}, M' = {04, 23, 57, 68}. Per l'edge cover, C = {05, 16, 27, 38, 49}, C' = {04, 16, 23, 49, 57, 68}. Per il cammino, P = {01, 12, 23, 34, 49, 97, 75, 58, 86} e P' = {01, 16, 69, 97, 75, 58, 83, 32}.

GRUPPO B. Per l'insieme stabile, S = {0, 3, 5, 9}, S' = {1, 8, 9}. Per il trasversale, i complementi di S, S'. Per il matching, M = {05, 16, 27, 38, 49}, M' = {01, 23, 58, 79}. Per l'edge cover, C = {05, 16, 27, 38, 49}, C' = {01, 16, 23, 49, 58, 79}. Per il cammino, P = {01, 12, 23, 34, 49, 97, 75, 58, 86} e P' = {01, 16, 69, 97, 75, 58, 83, 32}.

L'Algoritmo Greedy non risolve all'ottimo neppure il problema della massima clique perché, nonostante in entrambi i grafi i massimali abbiano tutti cardinalità 2, questa proprietà non è verificata per i massimali contenuti in un generico sottoinsieme di nodi (se così fosse, il problema ammetterebbe la *Proprietà delle Basi*, che è equivalente a quella di *Scambio*): per esempio, l'insieme S = {1, 5, 6} contiene due massimali – {5} e {1, 6} – di cardinalità diversa. Sia allora c_u il peso del nodo u e immaginate che nel grafo di sinistra si abbia $c_5 = 4$, $c_1 = c_6 = 3$, $c_0 = 1$ e $c_u = 0$ per tutti gli altri nodi. L'Algoritmo Greedy inizierebbe con il prendere il nodo 5, ma poi potrebbe scegliere solo il nodo 0 ottenendo così una soluzione di peso 4 + 1 = 5. La soluzione {1, 6} ha invece peso 6, dunque con questa scelta dei pesi il Greedy fallisce.

In entrambi i grafi l'Algoritmo Greedy risolve invece all'ottimo del massimo insieme di archi che non formano cicli (corrispondente al problema dell'albero ricoprente, che com'è noto possiede la struttura di matroide).

4. Risolvete tramite branch-and-bound il seguente problema di programmazione lineare intera:

GRUPPO A: $\min \; 3x_1 + 4x_2 + 5x_3$ subject to $2x_1 + 4x_2 + 6x_3 \geq 6$, $x_1, x_2, x_3 \geq 0$, intere

GRUPPO B: $\max \; 3x_1 + 4x_2 + 5x_3$ subject to $2x_1 + 4x_2 + 6x_3 \leq 6$, $x_1, x_2, x_3 \geq 0$, intere

Iniziate dalla soluzione ammissibile $x_1 = 3$, $x_2 = x_3 = 0$ (GRUPPO A) ovvero $x_1 = x_2 = 1$, $x_3 = 0$ (GRUPPO B). Operate le dicotomie, nell'ordine, sulle variabili frazionarie ottenute dai rilassamenti lineari dei sottoproblemi incontrati, e calcolate i bound sempre per rilassamento lineare. Specificate quali nodi possono chiudersi per inammissibilità e quali per effetto del bound trovato.