Esame Ricerca Operativa 09/07/2018

Cogn., nome e matr.:

Problema 1 (versione voto max 30): Torneo di tennis

16 giocatori devono affrontarsi in un torneo di tennis. Il tabellone può essere visto come un albero binario completo con 16 foglie, numerate da 1 a 16, in cui vanno collocati i giocatori. L'albero è disegnato con le foglie a sinistra e la radice a destra. Ogni nodo interno rappresenta una partita tra i due nodi suoi figli (si veda la figura).

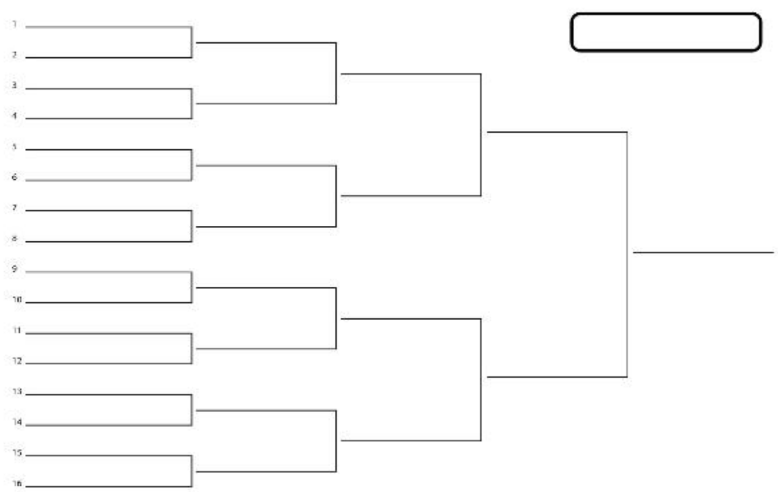

Figure 1: Il tabellone vuoto, in cui dobbiamo stabilire le partite del primo turno.

Assegnando i giocatori alle foglie $1, \dots, 16$ vengono completamente determinati i primi turni (1 vs 2, 3 vs 4, ecc.), dopodichè, per ogni partita, avanzerà al turno successivo il vincitore (ma chi egli sia non dipende da noi).

I giocatori hanno diversi gradi di abilità e, per evitare partite troppo sbilanciate e rendere il torneo spettacolare, gli organizzatori non vogliono che certe partite si svolgano troppo presto (ad esempio, che i due giocatori più forti si debbano affrontare al primo turno).

Per ogni coppia di giocatori i e j è pertanto noto un numero $t(i, j) \in \{1, \dots, 4\}$ che specifica a che turno, come minimo, può verificarsi la partita i vs j. Ad esempio

- se $t(i, j) = 1$ allora i e j possono affrontarsi al primo turno
- se $t(i, j) = 2$ allora i e j non possono affrontarsi prima del secondo turno

- se $t(i, j) = 4$, la partita i vs j, ammesso che venga mai giocata, potrebbe solo che essere la finale.

Possiamo vedere i valori $t(i, j)$ come i pesi degli archi di un grafo completo $G = (V, E, t)$ con $V = \{1, \dots, 16\}$.

È chiaro che, dato G, non è detto che sia possibile fare un tabellone che rispetti i vincoli richiesti (ad esempio, se $t(i, j) > 1$ per ogni i, j, non sarebbe possibile alcuna partita al primo turno. Oppure se $t(i, j) = 4$ per due coppie distinte (i_1, j_1) e (i_2, j_2), ci dovrebbero essere due finali anzichè una sola...). Siccome il tabellone va comunque fatto, l'obiettivo è minimizzare il numero di partite i vs j che potrebbero verificarsi prima del turno minimo dato da $t(i, j)$.

Si scriva perciò un modello di programmazione lineare intera per decidere come assegnare i giocatori alle foglie (e quindi quali saranno le partite del primo turno) in modo che il numero di coppie di giocatori che potrebbero potenzialmente incontrarsi prima di quanto specificato dal grafo G sia minimo possibile.

Soluzione

Introduciamo n^2 variabili binarie x_{ij} per $i, j = 1, \dots, 16$ il cui significato è

"$x_{ij} \leftrightarrow$ il giocatore i viene assegnato alla foglia j"

Definiamo inoltre $n(n-1)/2$ variabili binarie y_{ij} per ogni $1 \leq i < j \leq 16$ il cui senso è "i giocatori i e j sono stati messi male nel tabellone (ossia potrebbero scontrarsi prima del turno $t(i, j)$)".

La funzione obiettivo è

$$\min \sum_{i=1}^{15} \sum_{j=i+1}^{16} y_{ij}$$

Siccome ogni giocatore va assegnato a una foglia abbiamo i vincoli

$$\sum_{j=1}^{16} x_{ij} = 1 \qquad \forall i = 1, \dots, 16$$

Inoltre siccome ogni foglia deve contenere un giocatore abbiamo i vincoli

$$\sum_{i=1}^{16} x_{ij} = 1 \qquad \forall j = 1, \dots, 16$$

Detto 1 il livello delle foglie e 6 il livello della radice, per ogni coppia i, j di foglie, indichiamo con $l(i, j)$ il livello del loro primo antenato comune (ossia il nodo in cui i cammini da i alla radice e da j alla radice si incontrano). Ad esempio:

$l(1,2) = 2$
$l(9,12) = 3$
$l(1,8) = 5,$
$l(1,9) = 6$
ecc. Si noti che $l(i,j)$ *non è una variabile* ma una costante del problema. Dipende solo dall'albero binario ed è facile da calcolare.

Se nel tabellone non ci fossero errori si avrebbe

$$x_{if} + x_{jg} \leq 1 \qquad \forall i,j,f,g : t(i,j) > l(f,g)$$

Siccome noi ammettiamo gli errori (ma vogliamo minimizzarli) sostituiamo il vincolo qui sopra con

$$x_{if} + x_{jg} \leq 1 + y_{ij} \qquad \forall i,j,f,g : t(i,j) > l(f,g)$$

Chiaramente, la funzione obiettivo cercherà di tenere gli y_{ij} al minimo possibile (ad esempio, se tutti sono 0, il tabellone sarebbe ottimo e soddisferebbe tutti i vincoli di G).

Nota. Chi non avesse introdotto una notazione comoda come gli $l(x,y)$ da noi usati avrebbe comunque potuto esplicitamente scrivere i vincoli ad esempio come:

$$x_{i,f} + x_{j,g} \leq 1 + y_{i,j} \qquad \forall i,j : t(i,j) > 2, \quad \forall k = 0,\ldots,7 \quad \forall f,g : 2k+1 \leq f < g \leq 2(k+1),$$

$$x_{i,f} + x_{j,g} \leq 1 + y_{i,j} \qquad \forall i,j : t(i,j) > 3, \quad \forall k = 0,\ldots,3 \quad \forall f,g : 4k+1 \leq f < g \leq 4(k+1)$$

$$x_{i,f} + x_{j,g} \leq 1 + y_{i,j} \qquad \forall i,j : t(i,j) > 4, \quad \forall k = 0,1 \quad \forall f,g : 8k+1 \leq f < g \leq 8(k+1)$$

Problema n.1 (versione voto max 27): Esplosione spaziale

Ci troviamo a bordo di un'astronave che esploderà tra T minuti. Abbiamo un piccolo razzo da utilizzare per salvarci allontanandoci dall'astronave. La distanza di salvezza dall'astronave al momento dell'esplosione è di almeno K kilometri, o saremo annientati dall'onda d'urto.

A bordo dell'astronave ci sono n oggetti. Ogni oggetto i ha un valore c_i, un ingombro volumetrico v_i, un peso p_i e il tempo necessario per caricarlo è di t_i secondi. Il razzo di salvataggio può contenere (oltre a noi) oggetti per un volume complessivo al massimo V e per un peso complessivo al massimo M (i pesi sono tutti espressi in kg).

La velocità del razzo dipende dal suo peso[1]. Il peso del razzo con noi a bordo, ma senza oggetti, è P_0. Se P è il peso finale del razzo, la sua velocità è

$$\frac{20000}{P} \quad \text{km/sec}$$

Si scriva un modello di programmazione lineare intera per massimizzare il valore degli oggetti che porteremo con noi sul razzo tenendo presente che non abbiamo nessuna intenzione di esplodere (ossia dobbiamo garantirci di poterci salvare).

Soluzione Introduciamo n variabili binarie x_i per decidere se prendere o no l'i-mo oggetto.

La fz obiettivo è

$$\max \sum_{i=1}^{n} c_i x_i$$

il vincolo sul volume complessivo è

$$\sum_{i=1}^{n} v_i x_i \leq V$$

e quello sul peso complessivo

$$\sum_{i=1}^{n} p_i x_i \leq M$$

Infine, detto P_X il peso caricato (ossia $P = P_0 + P_X$), per percorrere K chilometri serviranno $KP/20000$ secondi. Noi vogliamo che questo valore sia $\leq T' - t_X$ (dove $T' = 60T$ è il tempo prima dell'esplosione in secondi, e t_X è il tempo impiegato per caricare gli oggetti) i.e.

$$K(P_0 + P_X) \leq 20000(60T - t_X)$$

da cui

$$KP_X - 20000 t_X \leq 1200000T - KP_0$$

che, tradotto nel modello di PLI, diventa

$$\sum_{i=1}^{n} (Kp_i - 20000 t_i) x_i \leq M'$$

dove abbiamo definito la costante $M' := 1200000T - KP_0$.

[1]NB: supponiamo la velocità costante sin dalla partenza, tralasciando i dettagli dell'accelerazione

Problema 2 : Duale di Max Matching

Il problema del massimo matching pesato su un grafo $G = (V, E)$ con costi $c : E \mapsto \mathbb{R}^+$ può essere modellato nel seguente modo:

$$\max \sum_{ij \in E} c_{ij}\, x_{ij} \qquad (1)$$

subject to

$$\sum_{ij \in \delta(i)} x_{ij} \leq 1 \qquad \forall i \in V \qquad (2)$$

$$x_{ij} \in \{0, 1\} \qquad \forall ij \in E \qquad (3)$$

Si scriva il generico duale del rilassamento di PL del modello (1),(2),(3). Si scrivano poi, specificamente, il primale e il duale relativamente al caso $V = \{1, 2, 3, 4, 5, 6\}$, $E = \{12, 13, 16, 24, 25, 26, 35\}$ e $c_{24} = c_{25} = 1$, $c_{26} = c_{35} = 2$, $c_{16} = 3$, $c_{13} = 4$ e $c_{12} = 5$.

Che interpretazione si può dare del duale di max matching quando $c_{ij} = 1$ per ogni $ij \in E$?

Sol:

Introduciamo una variabile duale y_i per ogni $i \in V$.

La funzione obiettivo del duale è

$$\min \sum_{i \in V} y_i$$

con i vincoli

$$y_i + y_j \geq c_{ij} \qquad \forall ij \in E$$

sulle variabili

$$y_i \geq 0 \quad \forall i \in V$$

Per l'esempio specifico, il primale è

$$\max 5x_{12} + 4x_{13} + 3x_{16} + x_{24} + x_{25} + 2x_{26} + 2x_{35}$$

subject to

$$\begin{array}{ccccccccc}
x_{12} & +x_{13} & +x_{16} & & & & & \leq 1 \\
x_{12} & & & +x_{24} & +x_{25} & +x_{26} & & \leq 1 \\
 & x_{13} & & & & & +x_{35} & \leq 1 \\
 & & & x_{24} & & & & \leq 1 \\
 & & & & +x_{25} & & +x_{35} & \leq 1 \\
 & & x_{16} & & & +x_{26} & & \leq 1
\end{array}$$

con $x \geq 0$.

Il duale è

$$\min y_1 + y_2 + y_3 + y_4 + y_5 + y_6$$

subject to

$$\begin{array}{lllllll}
y_1 & +y_2 & & & & & \geq 5 \\
y_1 & & +y_3 & & & & \geq 4 \\
y_1 & & & & & +y_6 & \geq 3 \\
 & y_2 & & +y_4 & & & \geq 1 \\
 & y_2 & & & +y_5 & & \geq 1 \\
 & y_2 & & & & +y_6 & \geq 2 \\
 & & y_3 & & +y_5 & & \geq 2
\end{array}$$

con $y \geq 0$.

Se $c_{ij} = 1$ per ogni ij allora il duale rappresenta il modello di PL per il problema del minimo Vertex Cover.

Problema 3

Siano a, b, c, d variabili rappresentanti decisioni binarie. Scrivere uno o più vincoli lineari che forzino la seguente relazione tra a, b, c, d:

"se $a = b$ allora anche $c = d$"

(si noti che si tratta di un "se", e non di un "se e solo se")

Sol:

Si tratta di rendere impossibili le seguenti qudruple:

0 0 1 0
0 0 0 1
1 1 1 0
1 1 0 1

Ogni quadrupla può essere resa impossibile intrepretandola come un OR della negazione dei valori della quadrupla, ad esempio

0 0 1 0

è impossibile se vale

$$a \vee b \vee \neg(c) \vee d$$

che, come disuguaglianza, diventa

$$a + b + (1 - c) + d \geq 1$$

ossia

$$a + b + d - c \geq 0$$

facendo lo stesso sulle altre quadruple otteniamo

$$a + b + c - d \geq 0$$

$$a + b + c - d \leq 2$$

$$a + b + d - c \leq 2$$