Esercizi di PLI

a cura di A. Agnetis

1

Risolvere il seguente problema di PLI con l'algoritmo dei piani di Gomory:

$$\begin{aligned} \max z = 40x_1 + 24x_2 + 15x_3 + 8x_4 & & \\ 8x_1 + 6x_2 + 5x_3 + 4x_4 & \leq & 22 \\ x_i & \geq & 0 \\ x_i & \textit{intero} & \end{aligned}$$

Soluzione.
Si tratta di un problema di knapsack intero, che potrebbe dunque anche risolversi ad esempio con la programmazione dinamica, o con un branch and bound specializzato. Tuttavia utilizziamo i piani di Gomory.
Portando il problema in forma canonica:

$$\begin{array}{c|cccccc} 0 & -40 & -24 & -15 & -8 & 0 \\ \hline 22 & 8 & 6 & 5 & 4 & 1 \end{array}$$

l'elemento di pivot è chiaramente 8, da cui il tableau ottimo

$$\begin{array}{c|ccccc} 110 & 0 & 6 & 10 & 12 & 5 \\ \hline \frac{22}{8} & 1 & \frac{3}{4} & \frac{5}{8} & \frac{1}{2} & \frac{1}{8} \end{array}$$

La soluzione ottima è frazionaria. Dunque, aggiungendo il taglio si ha:

$$\begin{array}{c|cccccc} 110 & 0 & 6 & 10 & 12 & 5 & 0 \\ \hline \frac{22}{8} & 1 & \frac{3}{4} & \frac{5}{8} & \frac{1}{2} & \frac{1}{8} & 0 \\ -\frac{3}{4} & 0 & -\frac{3}{4} & -\frac{5}{8} & -\frac{1}{2} & -\frac{1}{8} & 1 \end{array}$$

applicando una iterazione del metodo duale del simplesso, si ha che l'elemento pivot è -3/4, si ottiene il nuovo tableau ottimo:

104	0	0	5	8	4	8
2	1	0	0	0	0	1
1	0	1	$\frac{5}{6}$	$\frac{2}{3}$	$\frac{1}{6}$	$\frac{4}{3}$

e quindi la soluzione ottima $(2, 1, 0, 0)$.

2

Risolvere il seguente problema di PLI con l'algoritmo dei piani di Gomory, indicando le equazioni del taglio in funzione delle variabili originarie:

$$\begin{array}{lrcl} \max z = x_1 + 2x_2 & & & \\ & x_1 - 2x_2 & \geq & -2 \\ & x_1 + x_2 & \leq & 3 \\ & x_1, x_2 & \geq & 0 \\ & x_1, x_2 & \multicolumn{2}{l}{\textit{interi}} \end{array}$$

Soluzione.
Portando il problema in forma canonica si ha

0	-1	-2	0	0
2	-1	2	1	0
3	1	1	0	1

ove

$$\begin{array}{rcl} x_3 & = & 2 + x_1 - 2x_2 \\ x_4 & = & 3 - x_1 - x_2 \end{array}$$

Dopo due iterazioni del metodo del simplesso si arriva al tableau ottimo:

$\frac{14}{3}$	0	0	$\frac{1}{3}$	$\frac{4}{3}$
$\frac{5}{3}$	0	1	$\frac{1}{3}$	$\frac{1}{3}$
$\frac{4}{3}$	1	0	-$\frac{1}{3}$	$\frac{2}{3}$

La soluzione è frazionaria, occorre quindi generare un taglio. Scegliendo la prima riga, si ha:

$$-\frac{1}{3}x_3 - \frac{1}{3}x_4 \leq -\frac{2}{3}$$

e dunque il nuovo tableau:

$$
\begin{array}{c|ccccc}
\frac{14}{3} & 0 & 0 & \frac{1}{3} & \frac{4}{3} & 0 \\
\hline
\frac{5}{3} & 0 & 1 & \frac{1}{3} & \frac{1}{3} & 0 \\
\frac{4}{3} & 1 & 0 & -\frac{1}{3} & \frac{2}{3} & 0 \\
-\frac{2}{3} & 0 & 0 & -\frac{1}{3} & -\frac{1}{3} & 1
\end{array}
$$

effettuando l'operazione di pivot attorno all'elemento sulla riga 3 e in colonna 3, si ottiene il nuovo tableau ottimo

$$
\begin{array}{c|ccccc}
4 & 0 & 0 & 0 & 1 & 1 \\
\hline
1 & 0 & 1 & 0 & 0 & 1 \\
2 & 1 & 0 & 0 & 1 & -1 \\
2 & 0 & 0 & 1 & 1 & -3
\end{array}
$$

e dunque la soluzione ottima del problema originario è $(2, 1)$. Per ottenere l'espressione del taglio, osserviamo che questo può scriversi

$$x_3 + x_4 \geq 2$$

da cui, sfruttando le espressioni di x_3 e x_4 sopra riportate, si ha in definitiva che il taglio è

$$x_2 \leq 1$$

3

Un insieme di 7 lavorazioni deve essere eseguito da una macchina a controllo numerico. Ciascuna lavorazione richiede alcuni utensili, e, una volta effettuata, porta un certo profitto. La seguente tabella mostra, per ogni lavorazione, l'insieme di utensili richiesto e il corrispondente profitto:

lavorazione	Utensili	profitto
1	A, C, D, E	2000
2	A, B, F, G	1500
3	A, B, C, E	1800
4	B, D, E, G	1700
5	E, F, G	800
6	A, D, G	900
7	A, E, F	750

Il magazzino utensili della macchina può ospitare solo al più 5 utensili. Formulare come PLI il problema di determinare la configurazione del magazzino utensili con l'obiettivo di massimizzare il profitto.

Soluzione.

Si tratta di decidere quali utensili dovranno essere montati sulla macchina, e di conseguenza quali lavorazioni potranno essere effettuate. Risulta allora conveniente introdurre le seguenti variabili di decisione:

$$x_i = \begin{cases} 1 & \text{se la lavorazione i è effettuata} \\ 0 & \text{altrimenti} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{se l'utensile j è montato} \\ 0 & \text{altrimenti} \end{cases}$$

La funzione obiettivo risulta dunque essere

$$\max 2000x_1 + 1500x_2 + 1800x_3 + 1700x_4 + 800x_5 + 900x_6 + 750x_7$$

Vediamo ora i vincoli. Un primo tipo di vincoli consiste nell'esprimere il fatto che una lavorazione può essere eseguita solo se tutti i corrispondenti utensili sono montati sulla macchina. Questo può essere espresso dicendo che una lavorazione i non può effettuarsi se l'utensile j richiesto non è montato. Nel nostro esempio, otteniamo 25 vincoli:

$$\begin{aligned}
x_1 &\leq y_A \\
x_1 &\leq y_C \\
x_1 &\leq y_D \\
x_1 &\leq y_E \\
x_2 &\leq y_A \\
x_2 &\leq y_B \\
x_2 &\leq y_F \\
x_2 &\leq y_G \\
\ldots \\
\ldots
\end{aligned}$$

In alternativa, si poteva più compattamente scrivere

$$\begin{aligned}
x_1 + x_2 + x_3 + x_6 + x_7 &\leq 5y_A \\
x_2 + x_3 + x_4 &\leq 3y_B \\
x_1 + x_3 &\leq 2y_C \\
\ldots \\
\ldots
\end{aligned}$$

dove, si noti, se $y_j = 0$ tutte le variabili corrispondenti a lvaorazioni che richiedono l'utensile j sono contemporaneamente poste a 0. (Questa seconda formulazione è però molto più debole, dal punto di vista risolutivo, della prima).

Un secondo tipo di vincoli è relativo alla capacità limitata di immagazzinamento della macchina, che può esprimersi semplicemente come

$$y_A + y_B + y_C + y_D + y_E + y_F + y_G \leq 5$$

4

In una rete di calcolatori, vi sono n terminali ciascuno dei quali deve essere collegato ad un concentratore. Ci sono m concentratori, a ognuno dei quali possono essere collegati al più k terminali. Perché un concentratore possa essere collegato a un terminale, quest'ultimo deve essere "attivo". Il costo di attivazione di un concentratore j è f_j, mentre il costo di collegamento del terminale i con il concentratore j è c_{ij}. Formulare come PLI il problema di minimizzare i costi complessivi, nel rispetto dei vincoli.

Soluzione.

Si tratta di un problema di plant location. Le variabili di decisione saranno

$$x_{ij} = \begin{cases} 1 & \text{se il terminale i è collegato al concentratore j} \\ 0 & \text{altrimenti} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{se il concentratore j è attivato} \\ 0 & \text{altrimenti} \end{cases}$$

La funzione obiettivo risulta dunque dalla somma dei due tipi di costi:

$$\min \sum_{i=1}^{n} c_{ij}x_{ij} + \sum_{j=1}^{m} f_j y_j$$

Per quanto riguarda i vincoli, oltre ai consueti vincoli di assegnamento (ciascun terminale deve essere connesso a uno e un solo concentratore)

$$\sum_{j=1}^{m} x_{ij} = 1 \quad i = 1, \ldots, n$$

avremo i vincoli per cui un terminale i non può essere collegato al concentratore j se quest'ultimo non è attivato, ossia

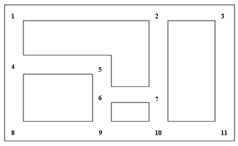
$$x_{ij} \leq y_j \quad i = 1, \ldots, n, \quad j = 1, \ldots, m$$

Infine, i vincoli sulla capacità di ciascun concentratore:

$$\sum_{i=1}^{n} x_{ij} \leq k \quad j = 1, \ldots, m$$

5

In figura è rappresentata la piantina di un museo, formato da 4 stanze e una rete di corridoi. Agli incroci dei vari corridoi (punti $1, \ldots, 11$), è possibile installare delle telecamere che riescono a controllare ambedue le direzioni ortogonali. Ad esempio, se si installa una telecamera nel punto 5, questa è in grado di sorvegliare i corridoi 4–5, 5–6 e 6–9. Supponendo che il costo di installazione di una telecamera nel punto i è pari a w_i, $i = 1, \ldots, 11$, formulare come problema di programmazione lineare a numeri interi il problema di sorvegliare tutti i corridoi del museo al costo minimo.



Soluzione.

Si tratta di un problema di set covering. Associando una variabile a ciascuno dei punti in cui potrebbe essere installata una telecamera:

$$x_i = \begin{cases} 1 & \text{se è installata una telecamera in i} \\ 0 & \text{altrimenti} \end{cases}$$

occorre imporre il vincolo che ciascun corridoio sia coperto da almeno una telecamera. La formulazione risulta quindi

$$\begin{array}{rcl}
\min \sum_{i=1}^{11} w_i x_i & & \\
x_1 + x_2 + x_3 & \geq & 1 \\
x_1 + x_4 + x_8 & \geq & 1 \\
x_2 + x_7 + x_{10} & \geq & 1 \\
x_3 + x_{11} & \geq & 1 \\
x_8 + x_9 + x_{10} + x_{11} & \geq & 1 \\
x_4 + x_5 & \geq & 1 \\
x_5 + x_6 + x_9 & \geq & 1 \\
x_6 + x_7 & \geq & 1
\end{array}$$

6

Ci sono n lavori *indivisibili* da effettuare, ognuno caratterizzato da una sua durata p_i, $i = 1, \ldots, n$. Avendo due persone a disposizione, volete ripartire i lavori tra queste due persone facendo in modo che i loro carichi di lavoro siano il più possibile bilanciati.

Formulare il problema in termini di problema di knapsack, e risolverlo con un opportuno algoritmo nel caso $n = 4$, $p_1 = 2$, $p_2 = 3$, $p_3 = 6$, $p_4 = 7$.

Soluzione.

Poiché la somma dei p_i è pari a 18, si tratta di risolvere il problema

$$\begin{aligned}
\max z = 2x_1 + 3x_2 + 6x_3 + 7x_4 & \\
2x_1 + 3x_2 + 6x_3 + 7x_4 &\leq 9 \\
x_i &\geq 0
\end{aligned}$$

dove le variabili pari a 1 e a 0 indicheranno la partizione dei lavori tra le due persone. Applicando l'algoritmo di programmazione dinamica si ottiene

$\downarrow k \rightarrow y$	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	2	2	2	2	2	2	2	2
2	0	0	2	3	3	5	5	5	5	5
3	0	0	2	3	3	5	6	6	8	9
4	0	0	2	3	3	5	6	7	8	9

Dalla tabella si ricostruisce una soluzione ottima, $x_1 = 1, x_2 = 0, x_3 = 0, x_4 = 1$.

7

In figura è rappresentata una rete stradale che unisce varie località. A ogni tratto di strada sono associati due numeri. Il primo rappresenta il tempo necessario a percorrerlo, il secondo il pedaggio che è necessario pagare. Si vuole andare da A a B nel minor tempo possibile, ma si hanno in tasca solo 44 euro per pagare i pedaggi. Formulare in termini di programmazione lineare a numeri interi il problema di trovare il percorso più conveniente.

Soluzione.

La formulazione si ottiene aggiungendo alla classica formulazione di cammino minimo un vincolo di knapsack. Dunque:

$$
\begin{aligned}
\min z &= 3x_{A1} + 4x_{A3} + 2x_{13} + 2.5x_{12} + x_{23} + 1.5x_{34} + 2x_{41} + 2x_{42} + 4x_{2B} + 6x_{4B} \\
x_{A1} + x_{A2} &= 1 \\
x_{A1} + x_{41} &= x_{13} + x_{12} \\
x_{12} + x_{42} &= x_{23} + x_{2B} \\
x_{A3} + x_{13} + x_{23} &= x_{34} \\
x_{34} &= x_{41} + x_{42} + x_{4B} \\
16x_{A1} + 20x_{A3} & \quad +30x_{13} + 13x_{12} + 6x_{23} + 25x_{34} + 11x_{41} + 8x_{42} + 14x_{2B} + 23x_{4B} \leq 44 \\
x_{ij} \quad \text{intero}
\end{aligned}
$$

8

Dato un grafo $G = (N, A)$, non orientato, si consideri il seguente problema. I nodi del gafo rappresentano città che devono essere difese da pattuglie. Le pattuglie possono essere dislocate in corrispondenza dei nodi del grafo, e possono essere di due tipi, *fisse* e *mobili*. Una pattuglia fissa dislocata nella città i difende solo la città i. Una pattuglia mobile dislocata nella città i difende, oltre a i, anche tutte le città adiacenti, ossia difende $\{i\} \cup \delta(i)$. Dislocare una pattuglia fissa ha un costo pari a 2, una mobile ha costo 3. Il problema è quello di difendere tutti i nodi del grafo al costo minimo. Formulare il problema come programmazione lineare a numeri interi.

Soluzione.

Consideriamo due insiemi di variabili, ossia:

$$
x_i = \begin{cases} 1 & \text{se è installata una pattuglia fissa in i} \\ 0 & \text{altrimenti} \end{cases}
$$

$$
y_i = \begin{cases} 1 & \text{se è installata una pattuglia mobile in i} \\ 0 & \text{altrimenti} \end{cases}
$$

La funzione obiettivo può essere immediatamente espressa come

$$\min \sum_{i \in N} (2x_i + 3y_i)$$

Si tratta ora di esprimere il vincolo sulla difesa di ciascuna città. Questo può essere scritto tenendo conto che una città i è difesa se in i è dislocata una pattuglia, oppure se in $i \cup \delta(i)$ ve ne è almeno una mobile:

$$x_i + y_i + \sum_{j \in \delta(i)} y_j \geq 1$$

9

Siete i curatori dell'opera omnia del celebre gruppo Hashirim, i quali hanno pubblicato, durante la loro carriera, complessivamente n brani, di varie durate (sia p_i la durata del brano i). Vi è stato chiesto di far uscire una collana di m compact disc, comprendente tutti i brani, con l'unico requisito che la differenza tra la durata complessiva del cd più lungo e di quello più corto sia minima.

Dovete decidere di quali brani deve essere composto ogni cd. Formulare il problema come programmazione lineare a numeri interi.

Soluzione.

Utilizziamo delle variabili di assegnamento, del tipo

$$x_{ij} = \begin{cases} 1 & \text{se il brano i è assegnato al cd j} \\ 0 & \text{altrimenti} \end{cases}$$

E dunque si hanno i vincoli in base ai quali ciascun brano deve essere assegnato a un cd:

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad i = 1, \ldots, n$$

Per scrivere la funzione obiettivo, dobbiamo introdurre le variabili scalari y e z che staranno a indicare la lunghezza del più corto e più lungo cd rispettivamente. Per fare questo basta imporre

$$y \leq \sum_{i=1}^{n} p_i x_{ij} \qquad j = 1, \ldots, m$$

$$z \geq \sum_{i=1}^{n} p_i x_{ij} \qquad j = 1, \ldots, m$$

dopo di che, la funzione obiettivo è semplicemente

$$\min z - y$$

10

Risolvere il seguente problema di knapsack con il metodo del branch and bound o con la programmazione dinamica:

$$
\begin{aligned}
\max z = 2x_1 + 3x_2 + 4x_3 + 6x_4 & \\
x_1 + 4x_2 + 5x_3 + 6x_4 &\leq 9 \\
x_j &\in \{0,1\}
\end{aligned}
$$

Risposta. La soluzione ottima è $(1,0,0,1)$, di valore 8.

11

Un'azienda produce scarpe. Nella tabella sono indicate le risorse richieste dalla produzione di ciascun paio di scarpe, e il prezzo di vendita. Il problema consiste nel determinare la produzione di ciascun tipo di scarpa, considerando le risorse disponibili, e tenendo presente che per ciascun tipo di scarpa è stabilita una soglia minima, sotto la quale non vi è convenienza a produrre (ad esempio, la produzione di scarponi sarà pari a 0 oppure dev'essere almeno pari a 100 paia, e cosi' per gli altri tipi). Formulare il problema di massimizzare il profitto in termini di programmazione lineare intera.

tipo	cuoio (g)	ore-macchina	chiodi	soglia	prezzo (euro/paio)
scarponi da montagna	850	3	20	100	150
mocassini	600	2	15	200	120
scarpe da passeggio	700	2.5	20	150	130
disponibilità	120000	7000	40000		

Soluzione.

Come in tutti i problemi di allocazione di risorse, avremo alcune variabili continue (x_1, x_2, x_3) corrispondenti al numero di unità di ciascun prodotto da produrre. La funzione obiettivo e i vincoli sulla disponibilità di risorse sono dunque:

$$
\begin{aligned}
\max z = 150x_1 + 120x_2 + 130x_3 & \\
850x_1 + 600x_2 + 700x_3 &\leq 120000 \\
3x_1 + 2x_2 + 2.5x_3 &\leq 7000 \\
20x_1 + 15x_2 + 20x_3 &\leq 40000
\end{aligned}
$$

oltre a questi vincoli, vanno però aggiunti quelli relativi alla soglia di produzione. In particolare, dobbiamo esprimere il fatto che se un tipo di articolo viene prodotto, la corrispondente variabile deve assumere almeno un certo valore. Questo richiede allora l'introduzione di variabili binarie:

$$y_i = \begin{cases} 1 & \text{se l'articolo i è prodotto} \\ 0 & \text{altrimenti} \end{cases}$$

vanno quindi aggiunti altri due tipi di vincoli. Il primo tipo impone che una variabile x_i non può essere maggiore di zero se quell'articolo non viene prodotto. Tale vincolo deve scomparire nel caso in cui invece l'articolo venga prodotto, per cui dobbiamo usare un numero M sufficientemente grande a secondo membro (ad es. $M > 100000$):

$$\begin{aligned} x_1 &\leq My_1 \\ x_2 &\leq My_2 \\ x_3 &\leq My_3 \end{aligned}$$

Un secondo tipo di vincoli deve invece imporre che, se un articolo viene prodotto, ciò deve avvenire in una certa quantità minima:

$$\begin{aligned} x_1 &\geq 100y_1 \\ x_2 &\geq 200y_2 \\ x_3 &\geq 150y_3 \end{aligned}$$

12

Una macchina deve effettuare n lavorazioni non interrompibili $\{1, 2, \ldots, n\}$. La lavorazione i richiede un tempo di processamento p_i. Per ogni lavorazione è specificata una due date d_i. Formulare in termini di programmazione lineare a numeri interi il problema di determinare l'ordine in cui eseguire le lavorazioni in modo tale da minimizzare la somma delle tardiness (ricordiamo che la tardiness T_i è definita come $\max\{0, C_i - d_i\}$, ove C_i è il tempo di completamento del job i).

Soluzione.

Si tratta di un problema di scheduling (NP completo in senso ordinario). Possiamo utilizzare le variabili continue S_i a indicare l'istante di inizio del job i, e dunque:

$$C_i = S_i + p_i \quad i = 1, \ldots, n$$

$$T_i \geq C_i - d_i \quad i = 1, \ldots, n$$

$$T_i \geq 0 \quad i = 1, \ldots, n$$

dove la variabile T_i, dal momento che stiamo minimizzando, sarà pari, in una soluzione ottima, esattamente alla tardiness del job i. La funzione obiettivo sarà dunque

$$\min \sum_{i=1}^{n} T_i$$

Rimane da esprimere il vincolo che la macchina non può eseguire più di una lavorazione alla volta. A questo scopo possiamo introdurre le variabili

$$s_{ij} = \begin{cases} 1 & \text{se la lavorazione i è effettuata prima di j} \\ 0 & \text{se la lavorazione j è effettuata prima di i} \end{cases}$$

Vogliamo dunque imporre che se $s_{ij} = 1$, allora deve essere

$$S_i + p_i \leq S_j$$

mentre ciò non è vero se $s_{ij} = 0$. Introducendo una costante M sufficientemente grande (ad esempio $M > \sum p_i$), per ogni coppia di job i, j avremo i due vincoli:

$$S_i + p_i \leq S_j + M(1 - s_{ij})$$

e

$$S_j + p_j \leq S_i + M(1 - s_{ji})$$

oltre a, ovviamente:

$$s_{ij} + s_{ji} = 1$$