26-01-2017

1. La zia Bice, ricamatrice, coordina la preparazione dei bavaglini da vendere al prossimo mercatino. I bavaglini sono di tre tipi: maschile, femminile e unisex. Ogni bavaglino richiede dei filati nelle quantità, in cm, indicate nella seguente tabella, che riporta anche il tempo in minuti richiesto e il ricavo di vendita.

Bavaglino	Azzurro	Rosa	Giallo	Verde
Maschile	100	10	30	20
Femminile	10	100	40	20
Unisex	30	10	50	70

I fornitori di filati mettono a disposizione delle confezioni con le seguenti caratteristiche (metri di filati dei vari colori e prezzo in euro):

Confezione	Azzurro	Rosa	Giallo	Verde	Prezzo
1	40	30	50	20	20
2	20	50	40	50	25
3	30	40	40	10	15

Ciascun bavaglino richiede manodopera per 15 minuti e viene venduto a 5 euro. La zia Bice e le sue numerose amiche potranno dedicare ai bavaglini 200 ore del loro tempo e devolveranno il ricavato delle vendite, al netto dei costi per i soli filati, in beneficienza. Tenendo conto che tutti i bavaglini ricamati saranno sicuramente venduti, scrivere il modello di programmazione lineare che determini quanti bavaglini ricamare al fine di massimizzare le somme devolute in beneficienza, considerando anche che:

- sono richiesti almeno 10 bavaglini per tipo;
- si vogliono acquistare al massimo due tipi di confezione;
- ciascun fornitore pratica uno sconto del 5% sul prezzo unitario di vendita se si acquistano almeno 10 delle loro confezioni (suggerimento: modellare la decisione sul numero di confezioni da acquistare a prezzo scontato).

Cominciamo con il creare le variabili decisionali per modellare i bavaglini in base al colore e le confezioni in base al colore come segue:

x_i: quantità di bavaglini del tipo $i \in \{M, F, U\}$

y_i: metri di filato di bavaglini della confezione $i \in \{1,2,3\}$

Quindi, volendo massimizzare i ricavi di beneficenza, avremo che:

- Ogni bavaglino viene venduto a 5 euro
- Ci sono dei costi di produzione, che vanno *sottratti* dai ricavi

$$max\ 5(x_U + x_F + x_M) - (20y_1 + 25y_2 + 15y_3)$$
$$s.t.$$
$$15x_M + 15x_F + 15x_U \leq 12000$$

(vincolo tempo complessivo, sapendo che per fare un bavaglino ci si impiegano 15 minuti e abbiamo convertito le 200 ore in minuti, quindi diventerebbe $200 * 60 = 12000$)

Sappiamo inoltre che:

- "sono richiesti almeno 10 bavaglini per tipo"
$$x_M \geq 10, x_F \geq 10, x_U \geq 10$$

- "si vogliono acquistare al massimo due tipi di confezione"
Ciò richiede la creazione di un'apposita variabile binaria.
z_i: variabile logica che vale 1 se acquistiamo la confezione del tipo $i \in \{M, F, U\}$
$$z_1 + z_2 + z_3 \leq 2$$
Con un vincolo di attivazione $x_i \leq Mz_i, \forall i \in \{M, F, U\}$
$$x_1 \leq Mz_1, x_2 \leq Mz_2, x_3 \leq Mz_3$$

- "ciascun fornitore pratica uno sconto del 5% sul prezzo unitario di vendita se si acquistano almeno 10 delle loro confezioni (suggerisce di modellare la decisione sul n. di conf. da acquistare scontate)
Usiamo una variabile logica per j:

w_j: variabile logica che vale 1 se acquistiamo a prezzo scontato confezioni del tipo $i \in \{1,2,3\}$, 0 altrimenti

$$y_1 \geq 10w_1, y_2 \geq 10w_2, y_3 \geq 10w_3$$

Implicitamente, si considerano i seguenti vincoli spuri, considerando lo sconto del 5% in funzione delle variabili a prezzo pieno presenti:

$$20 * w_1 + 25 * (1 - w_1) - (1 - w_1) * 20 * 0.05 <= 20 * y_1$$
$$25 * w_2 + 20 * (1 - w_2) - (1 - w_2) * 25 * 0.05 <= 25 * y_2$$
$$15 * w_3 + 15 * (1 - w_3) - (1 - w_3) * 15 * 0.05 <= 15 * y_3$$

Domini: $x_i \in Z_+, y_j \in Z_+, z_i \in \{0,1\}, w_j \in \{0,1\}, i \in \{M, F, U\}, j \in \{1,2,3\}$

2. Si risolva con il metodo del simplesso, applicando la regola anticiclo di Bland, il seguente problema di programmazione lineare:

$$\begin{array}{llrcrcrcr}
\max & & & & x_2 & - & 2\,x_3 & & \\
\text{s.t.} & x_1 & - & 2\,x_2 & + & x_3 & \leq & 1 \\
& x_1 & - & x_2 & + & 2\,x_3 & \geq & -1 \\
& & & 2\,x_2 & - & x_3 & \leq & 2
\end{array}$$

$$x_1 \geq 0 \qquad x_2 \geq 0 \qquad \mathbf{x_3 \leq 0}$$

$\max x_2 - 2x_3$

s.A $x_1 - 2x_2 + x_3 \leq 1$

$x_1 - x_2 + 2x_3 \geq -1$

$2x_2 - x_3 \leq 2$

$x_1 \geq 0, x_2 \geq 0, x_3 \leq 0$

$\to \min -x_2 + 2x_3$

s.A $x_1 - 2x_2 + x_3 + x_4 = 1$

$x_1 - x_2 + 2x_3 - x_5 = -1$

$2x_2 - x_3 + x_6 = 2$

$\to \hat{x}_3 = -x_3 \; , \; \hat{x}_3 \geq 0$

$\min -x_2 - 2\hat{x}_3$

s.A $x_1 - 2x_2 - \hat{x}_3 + x_4 = 1$

$x_1 - x_2 - 2\hat{x}_3 - x_5 = -1$

$2x_2 + \hat{x}_3 + x_6 = 2$

$\to$ termini noti positivi

$\min -x_2 - 2\hat{x}_3$

s.A $x_1 - 2x_2 - \hat{x}_3 + x_4 = 1$

$-x_1 + x_2 + 2\hat{x}_3 + x_5 = 1$

$2x_2 + \hat{x}_3 + x_6 = 2$

$B = (x_4\ x_5\ x_6)$

x_1	x_2	$\bar{x}_3$	x_4	x_5	x_6	$-z$	b
0	-1	-2	0	0	0	-1	0
x_4 1	-2	-1	1	0	0	0	1
x_5 -1	$\boxed{1}$	2	0	1	0	0	1
x_6 0	2	1	0	0	1	0	2

F. C → SI
AMM → SI
ILLIMIT. → NON SO
OTTIMA → NON SO
ENTRA → TRA x_2 e $\bar{x}_3$ → x_2 (BLAND)
ESCE → $\operatorname{argmin}\left\{\frac{1}{-2}, \frac{1}{1}, \frac{2}{2}\right\} \to x_5$
 x_5 x_6

$B = (x_4\ x_2\ x_6)$

x_1	x_2	$\bar{x}_3$	x_4	x_5	x_6	$-z$	b
-1	0	0	0	1	0	-1	1
x_4 -1	0	3	1	2	0	0	3
x_2 -1	1	2	0	1	0	0	1
x_6 $\boxed{2}$	0	-3	0	-2	1	0	0

$R_2' \leftarrow R_2$, $R_1' \leftarrow R_1 + 2R_2$, $R_3' \leftarrow R_3 - 2R_2$
$R_0' \leftarrow R_0 + R_2$ F. C → SI, AMM → SI,
ILLIMIT. → NON SO, OTTIMA → NON
ENTRA → x_1 x_4 x_2 x_6
ESCE → $\operatorname{argmin}\left\{\frac{3}{-1}, \frac{1}{-1}, \frac{0}{2}\right\} \to x_6$

$2-\frac{3}{2} = \frac{4-3}{2} = \frac{1}{2}$ $\quad 3-\frac{3}{2} = \frac{6-3}{2} = \frac{3}{2}$

$3\cdot\frac{2}{3} = \frac{6}{3} = 2$

$B = [x_4\ x_2\ x_1]$ $\quad 1 \cdot \cancel{2} \to 2$

	x_1	x_2	$\hat{x}_3$	x_4	x_5	x_6	$-z$	b
	0	0	-3/2	0	0	1/2	-1	1
x_4	0	0	$\boxed{3/2}$	1	1	1/2	0	3
x_2	0	1	1/2	0	0	1/2	0	1
x_1	1	0	-3/2	0	-1	1/2	0	0

$R_3' \leftarrow R_3/2$, $R_2' \leftarrow R_2 + R_3$, $R_1' \leftarrow R_1 + R_3$

$R_0' \leftarrow R_0 + R_1$

F.C → SI
AMM. → SI
ENTRA → $\hat{x}_3$

$z = -4$
$x = (3, 0, 2, 0, 0, 0)$

F.C → SI
AMM → SI
OTTIMA → SI

OSC → argmin $\left\{ \frac{3}{3/2}, \frac{1}{1/2}, \frac{0}{-3/2} \right\}$
$\quad x_4 \quad x_2 \quad x_1$

$B = [\hat{x}_3\ x_2\ x_1]$

x_1	x_2	$\hat{x}_3$	x_4	x_5	x_6	$-z$	b
0	0	0	$\cancel{3/}$1	1	1	-1	4
0	0	1	2/3	2/3	1/3	0	2
0	1	0	-1/3	-1/3	1/3	0	0
1	0	0	$\cancel{3/}$1	0	1	0	3

$R_0' \leftarrow R_0 + \frac{3}{2} R_1$

$R_1' \leftarrow \frac{2}{3} R_1$, $R_2' \leftarrow R_2 - \frac{1}{2} R_1$, $R_3' \leftarrow R_3 + \frac{3}{2} R_1$

3. Risolvere con il metodo del branch-and-bound il seguente problema di zaino 0/1

$$
\begin{array}{llllllllllllll}
\max & 14\,x_1 & + & 6\,x_2 & + & 12\,x_3 & + & 12\,x_4 & + & 7\,x_5 & + & 15\,x_6 & & \\
\text{s.t.} & 10\,x_1 & + & 6\,x_2 & + & 8\,x_3 & + & 6\,x_4 & + & 8\,x_5 & + & 12\,x_6 & \leq & 30 \\
& x_i \in \{0,1\}\,,\ i = 1\ldots6 & & & & & & & & & & & &
\end{array}
$$

Per risolvere questo problema knapsack 0/1 utilizzando il metodo branch and bound, possiamo procedere come segue:

- Si inizia creando un elenco di elementi, dove ogni elemento è rappresentato come una tupla (valore, peso). Per il problema dato, l'elenco di elementi sarebbe:
- [(14, 10), (6, 6), (12, 8), (12, 6), (7, 8), (15, 12)]

- Ordinare l'elenco degli elementi in ordine non crescente rispetto al rapporto tra valore e peso. Questo ci permetterà di considerare prima gli elementi di maggior valore.

- Creare una funzione upper bound che stimi il valore massimo ottenibile dagli elementi rimanenti. Un modo semplice per farlo è includere tutti gli oggetti rimanenti nello zaino, indipendentemente dal loro peso. In questo modo si otterrà un upper bound sul valore totale che si può ottenere.

- Creare una funzione che esegua la ricerca branch and bound. Questa funzione deve avere come input l'elenco degli elementi, il peso corrente, il valore corrente e il limite superiore.

- Inizializzare il miglior valore trovato finora a zero.

- Se il peso corrente è maggiore del peso massimo, restituisce il miglior valore trovato finora.

- Se il valore corrente più il limite superiore è inferiore al valore migliore trovato finora, restituisce il valore migliore trovato finora.

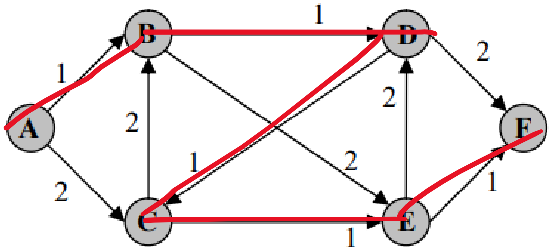
- Se non ci sono altri elementi da considerare, restituisce il valore attuale.

- Per ogni elemento, calcolare il valore e il peso se l'elemento è incluso nello zaino e il valore e il peso se l'elemento non è incluso nello zaino.

- Richiamare ricorsivamente la funzione branch and bound con il valore e il peso aggiornati per entrambi i casi e aggiornare il miglior valore trovato finora, se necessario.

- Restituire il miglior valore trovato finora.

4. Dato il seguente grafo, calcolare i cammini minimi a partire dal nodo A verso tutti gli altri nodi.



a. si scelga l'algoritmo da utilizzare e si motivi la scelta;
b. si applichi l'algoritmo scelto (riportare e **giustificare** i passi dell'algoritmo in una tabella);
c. si riportino l'albero e il grafo dei cammini minimi, e due cammini minimi da A a F (**descrivere** il procedimento per ottenere albero, grafo e cammini).

a) Il miglior algoritmo in questo contesto è Dijkstra, in quanto più efficiente e usabile in quanto tutti i costi ridotti sono positivi.
b) Ora, il calcolo dei cammini minimi, riportati in tabella. Si ricorda che:
- $\hat{v}$ rappresenta l'etichetta minima di ogni iterazione
- $\bar{S}$ rappresentano le etichette ancora da fissare
- Il segno * rappresenta l'etichetta fissata
- Il segno – rappresenta l'etichetta controllata ma non aggiornata
- Il segno x rappresenta l'etichetta non controllata perché il nodo è già fissato
- Gli spazi vuoti servono per indicare che non considero più l'etichetta nelle varie iterazioni in quanto fissata
- L'algoritmo termina quando non ci sono più nodi in $\bar{S}$

Iterazione	Nodo A	Nodo B	Nodo C	Nodo D	Nodo E	Nodo F	$\bar{S}$	$\hat{v}$
$Inizio$	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	A, B, C, D, E, F	
$h = 1$	*	1_A	2_A				B, C, D, E, F	A
$h = 2$		*		2_B	3_B		C, D, E, F	B
$h = 3$			3_D	*		4_D	C, E, F	D
$h = 4$			*		4_E		E, F	C
$h = 5$					*	5_E	F	E
$h = 6$						*	$\emptyset$	F

Ad ogni iterazione, percorriamo il grafo e scegliamo il percorso con costo minore. Ad ogni iterazione, scegliamo e fissiamo un'etichetta che ha costo minore, scremando ad ogni iterazione quelle da controllare e avere sempre in mano il costo minimo. Anche in questo caso, come per gli algoritmi label correcting in caso di convergenza alla soluzione ottima, le etichette calcolate e i relativi puntatori rappresentano, rispettivamente, una soluzione duale ammissibile e i predecessori su dei cammini dall'origine ai diversi nodi. Questo funziona non avendo costi negativi.

c) Come abbiamo visto per gli algoritmi label correcting in caso di convergenza, è possibile derivare l'albero (risp. il grafo) dei cammini minimi attraverso i puntatori ai predecessori (risp. la verifica della saturazione dei vincoli duali sugli archi).
Contrariamente a quanto chiede la domanda, non esistono altri cammini minimi, seguendo tutti i possibili percorsi. Albero e grafo coincidono.

5. a. Enunciare le condizioni di complementarietà primale-duale in generale.

b. Applicare tali condizioni per dimostrare che $(x_1, x_2, x_3) = (0, 2, 1)$ è soluzione ottima del seguente problema:

$$
\begin{array}{lrcrcrcr}
\min & 2x_1 & - & 4x_2 & + & x_3 & & \\
\text{s.t.} & 2x_1 & + & x_2 & + & x_3 & \geq & 2 \\
 & x_1 & - & x_2 & & & = & -2 \\
 & & & 2x_2 & - & 3x_3 & \leq & 1 \\
 & 2x_1 & + & x_2 & + & x_3 & \geq & 1 \\
\end{array}
$$

$$x_1 \leq 0 \qquad x_2 \text{ libera} \qquad x_3 \geq 0$$

$\min \; 2x_1 - 4x_2 + x_3$

$u_1 \quad 3 = 2x_1 + x_2 + x_3 \geq 2 \qquad x_1 \leq 0$

$u_2 \quad -2 = x_1 - x_2 = -2 \qquad x_2$ l.l

$u_3 \quad 1 = 2x_2 - 3x_3 \leq 1 \qquad x_3 \geq 0$

$u_4 \quad 3 = 2x_1 + x_2 + x_3 \geq 1$

$\max \; 2u_1 - 2u_2 + u_3 + u_4$

$x_1 \quad 2u_1 + u_2 + 2u_4 \geq 2$

$x_2 \quad u_1 - u_2 + 2u_3 + u_4 = -4$

$x_3 \quad u_1 - 3u_3 + u_4 \leq 1$

$u_1 \geq 0 \qquad u_2$ libera $\qquad u_3 \leq 0 \qquad u_4 \geq 0$

$u_1 \, (3-2) = 0 \rightarrow u_1 = 0$

$u_2 \, (-2+2) = 0 \rightarrow //$

$u_3 \, (1-1) = 0 \rightarrow //$

$u_4 \, (3-1) = 0 \rightarrow u_4 = 0$

$1 > x_3 (u_1 - 3u_3 + u_4 - 1) = 0$

$$
\begin{cases}
u_1 = 0 \\
u_4 = 0 \\
\cancel{u_1} - u_2 + 2u_3 + \cancel{u_4} = -4 & \Rightarrow u_2 = 4 - \frac{2}{3} = \frac{10}{3} \\
\cancel{u_1} - 3u_3 + \cancel{u_4} = 1 & \Rightarrow u_3 = -\frac{1}{3}
\end{cases}
$$

$\rightarrow u_1 = 0$

$u_4 = 0$

$u_2 = \frac{10}{3}$

$u_3 = -\frac{1}{3}$

ok dominio

ok vincoli

ok valore funzione

6. Si consideri il seguente tableau del simplesso:

x_1	x_2	x_3	x_4	x_5	x_6	z	b
0	0	– 4	1	– 7	0	– 1	5
0	1	2	0	– 6	0	0	8
0	0	2	– 1	1	1	0	2
1	0	3	4	2	0	0	3

Si dica, **senza svolgere calcoli** e fornendo una **giustificazione teorica delle risposte**:

a. riusciamo a individuare una soluzione di base corrispondente? Quale? Possiamo subito dire se è ottima?

b. perché non è consentita l'operazione di pivot sull'elemento evidenziato nel cerchio (1)?

c. su quali elementi è possibile effettuare il pivot secondo le regole del simplesso (indipendentemente dalle regole anticiclo)?

d. quale sarà il cambio base secondo le regole del simplesso e applicando la regola di Bland? Perché la soluzione di base ottenuta in seguito a questo cambio base è sicuramente degenere?

a) La soluzione di base corrispondente è data dall'individuazione dei pivot della matrice identità, quindi avremo $[x_2, x_5, x_1]$ oppure $[x_2, x_6, x_1]$. Mentre la prima non è ottima, la seconda non si sa se lo sia, in quanto i coefficienti sono tutti ≥ 0, ma non sono tutti positivi.

b) Non è consentita l'operazione su quell'elemento in quanto non rispetta la regola di individuazione dell'elemento di pivot rispetto a variabile che entra/variabile che esce. Si dovrebbe scegliere per Bland x_3 ed effettuare lì il rapporto minimo. Qui tale cosa non accade, dunque, non viene rispettata la regola del rapporto minimo.

c) Per le regole del simplesso, possiamo effettuare pivot sulle coppie di elementi $[x_3, x_6]$ e $[x_3, x_1]$

d) Il cambio base applicando Bland viene dato su $[x_3, x_1]$. In questo caso, avremmo due righe corrispondenti al rapporto minimo, pertanto avremo che almeno una esce dalla base con valore 0, mentre le altre due assumeranno valore 0 rimanendo in base.