RICERCA OPERATIVA (a.a. 2008/09)

Nome Cognome:

Corso di Laurea: $\boxed{\text{I}}$ $\boxed{\text{SI}}$ **Matricola:** **Corso:** $\boxed{\text{A}}$ $\boxed{\text{B}}$

1) La ditta *FastShip* deve caricare n bancali su una nave avente m stive. Sono noti il peso p_i del bancale i e la capacità u_j della stiva j. Per motivi di stabilità del carico della nave, la stiva 1 e la stiva m devono essere necessariamente utilizzate. Inoltre, la differenza in valore assoluto tra il peso totale degli oggetti caricati in tali due stive non deve eccedere una soglia di tolleranza prefissata ε.

Sapendo che l'utilizzo della stiva j comporta il pagamento di un costo di manutenzione f_j, e che il caricamento del bancale i nella stiva j richiede un costo di caricamento c_{ij}, si formuli in termini di P.L.I. il problema di decidere come caricare i bancali nelle stive della nave, in modo da rispettare il vincolo di stabilità del carico ed i vincoli di capacità, minimizzando il costo totale derivante dalla manutenzione delle stive e dalle operazioni di caricamento.

SVOLGIMENTO

Per descrivere il problema, introduciamo le seguenti variabili logiche:

$$x_{ij} = \begin{cases} 1, & \text{se il bancale } i \text{ viene caricato nella stiva } j \\ 0, & \text{altrimenti,} \end{cases} \qquad i = 1, 2, \ldots, n, \ j = 1, 2, \ldots, m.$$

$$y_j = \begin{cases} 1, & \text{se la stiva } j \text{ viene utilizzata} \\ 0, & \text{altrimenti,} \end{cases} \qquad j = 2, \ldots, m-1.$$

Utilizzando tali variabili logiche, il problema può essere formulato nel modo seguente:

$$\begin{array}{lll} \min & \sum_{j=2}^{m-1} f_j y_j + \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} & \\ & \sum_{j=1}^{m} x_{ij} = 1 & i = 1, \ldots, n \\ & \sum_{i=1}^{n} p_i x_{ij} \leq u_j y_j & j = 2, \ldots, m-1 \\ & \sum_{i=1}^{n} p_i x_{ij} \leq u_j & j = 1, m \\ & \sum_{i=1}^{n} x_{ij} \geq 1 & j = 1, m \\ & \sum_{i=1}^{n} p_i x_{i1} - \sum_{i=1}^{n} p_i x_{im} \leq \varepsilon & \\ & \sum_{i=1}^{n} p_i x_{im} - \sum_{i=1}^{n} p_i x_{i1} \leq \varepsilon & \\ & x_{ij}, y_j \in \{0, 1\} & i = 1, \ldots, n, \ j = 2, \ldots, m-1 \end{array}$$

Il primo blocco di vincoli garantisce che ogni bancale sia caricato in una stiva. Il secondo blocco garantisce che ogni stiva j (con $j \neq 1, m$) sia caricata nel rispetto della propria capacità u_j: se utilizzata, il primo membro è positivo e quindi la variabile y_j deve necessariamente assumere valore 1. I successivi due gruppi di vincoli garantiscono che le stive 1 e m siano necessariamente utilizzate nel rispetto della propria capacità. Gli ultimi due gruppi di vincoli impongono che la differenza in valore assoluto tra il peso totale degli oggetti caricati nella stiva 1 ed il peso totale degli oggetti caricati nella stiva m sia minore od uguale a ε. Il costo totale è dato dalla somma dei costi di manutenzione delle stive utilizzate e dei costi di caricamento.

2) Si considerino la funzione $f : [0, M] \to \mathbb{R}$ così definita

$$f(z) := \begin{cases} 0 & \text{se } \ 0 \le z \le m \\ b + az & \text{se } \ m < z \le M, \end{cases}$$

dove $a, b \in \mathbb{R}_+$ sono valori dati, ed il problema di Programmazione Matematica

$$\min \left\{ \ f\Big(\sum_{i=1}^{n} c_i x_i\Big) \ : \ \sum_{i=1}^{n} p_i x_i \ge Q, \quad x_i \in \{0,1\} \ i = 1, ..., n \ \right\},$$

dove $Q, c_i, p_i \in \mathbb{R}_+$ sono valori dati (tali che $\sum_{i=1}^{n} c_i \le M$ e $\sum_{i=1}^{n} p_i \ge Q$). Riformulare il problema dato in termini di Programmazione Lineare Intera, giustificando la risposta.

SVOLGIMENTO

La funzione obiettivo del problema non è lineare in quanto f è una funzione lineare a tratti. È possibile riformulare il problema in termini di PLI introducendo una variabile decisionale y che distingua le situazioni in cui ci si trova nei due tratti distinti:

$$y = \begin{cases} 0 & \text{se } \ 0 \le \sum_{i=1}^{n} c_i x_i \le m \\ \\ 1 & \text{se } \ m \le \sum_{i=1}^{n} c_i x_i \le M. \end{cases}$$

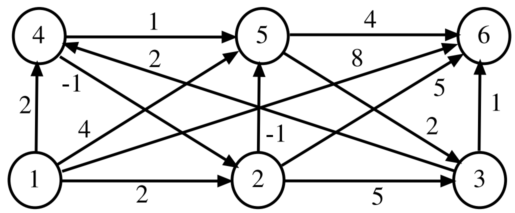
Per legare la variabile y ai valori effettivamente assunti dalla quantità $\sum_{i=1}^{n} c_i x_i$, introduciamo una variabile z_1, che individui i valori del primo intervallo, ed una variabile z_2 che individui i valori del secondo intervallo. Pertanto i vincoli $0 \le z_1 \le m(1-y)$ e $my \le z_2 \le My$ esprimono questo legame e la somma $z_1 + z_2$ individua l'effettivo valore per la quantità considerata. La funzione obiettivo è data da $by + az_2$. Infatti, se $y = 0$, allora anche $z_2 = 0$ e la funzione obiettivo vale 0, mentre se $y = 1$, allora $z_1 = 0$ e la funzione obiettivo vale $b + a\sum_{i=1}^{n} c_i x_i$.

Si ottiene quindi la seguente formulazione:

$$\begin{aligned} & \min \quad by + az_2 \\ & 0 \le z_1 \le m(1-y) \\ & my \le z_2 \le My \\ & \sum_{i=1}^{n} c_i x_i = z_1 + z_2 \\ & \sum_{i=1}^{n} p_i x_i \ge Q \\ & x_i, y \in \{0,1\} \quad i = 1, ..., n \end{aligned}$$

Si osservi che quando $\sum_{i=1}^{n} c_i x_i = m$ entrambe le scelte $(y, z_1, z_2) = (0, m, 0)$ e $(y, z_1, z_2) = (1, 0, m)$ sono possibili. La prima fornisce il giusto valore della funzione obiettivo mentre la seconda fornisce il valore $b + am$. Poiché questo valore è più grande di quello corretto, l'ambiguità di questa doppia rappresentazione viene risolta automaticamente a livello di ottimizzazione.

3) Si individui un albero dei cammini minimi di radice 1 sul grafo in figura, utilizzando l'algoritmo più appropriato dal punto di vista della complessità computazionale e giustificando la scelta effettuata. Per ogni iterazione si forniscano il nodo selezionato u, i vettori dei predecessori e delle etichette, l'insieme dei nodi candidati Q. Durante l'algoritmo si esplorino gli archi della la stella uscente del nodo selezionato u in ordine crescente del nodo testa. Al termine si disegni l'albero dei cammini minimi individuato.



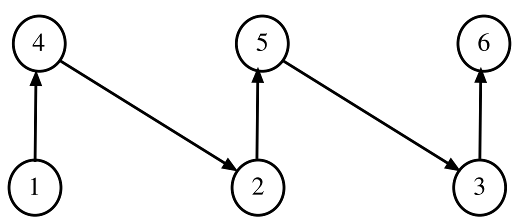
SVOLGIMENTO

Essendo presenti archi di costo negativo (ma non cicli di costo negativo), l'algoritmo più conveniente dal punto di vista della complessità computazionale, tra quelli studiati, è l'algoritmo SPT.L, in cui Q è implementata come fila, che ha complessità in tempo $O(nm)$.

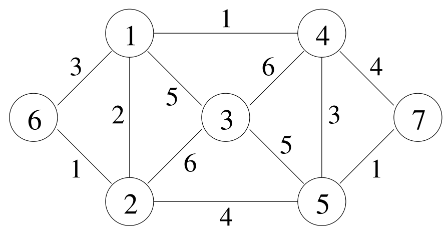
$M = (n-1)c_{max} + 1 = 5 \times 8 + 1 = 41.$

it.	u	$p[1]$	$p[2]$	$p[3]$	$p[4]$	$p[5]$	$p[6]$	$d[1]$	$d[2]$	$d[3]$	$d[4]$	$d[5]$	$d[6]$	Q
0		*nil*	1	1	1	1	1	0	41	41	41	41	41	$\{1\}$
1	1	*nil*	1	1	1	1	1	0	2	41	2	4	8	$\{2,4,5,6\}$
2	2	*nil*	1	2	1	2	2	0	2	7	2	1	7	$\{4,5,6,3\}$
3	4	*nil*	4	2	1	2	2	0	1	7	2	1	7	$\{5,6,3,2\}$
4	5	*nil*	4	5	1	2	5	0	1	3	2	1	5	$\{6,3,2\}$
5	6	*nil*	4	5	1	2	5	0	1	3	2	1	5	$\{3,2\}$
6	3	*nil*	4	5	1	2	3	0	1	3	2	1	4	$\{2,6\}$
7	2	*nil*	4	5	1	2	3	0	1	3	2	0	4	$\{6,5\}$
8	6	*nil*	4	5	1	2	3	0	1	3	2	0	4	$\{5\}$
9	5	*nil*	4	5	1	2	3	0	1	2	2	0	4	$\{3\}$
10	3	*nil*	4	5	1	2	3	0	1	2	2	0	3	$\{6\}$
11	6	*nil*	4	5	1	2	3	0	1	2	2	0	3	$\emptyset$

L'albero trovato è mostrato in figura.



4) Si applichi l'algoritmo di Kruskal per determinare un albero di copertura di costo minimo sul grafo in figura. Per ogni iterazione si indichino: l'arco in esame; quale fra le operazioni di inserzione e cancellazione viene applicata; nel primo caso mostrare un taglio, nel secondo fornire il ciclo individuato dall'algoritmo. Al termine fornire l'albero di copertura di costo minimo $T = (N, A_T)$. Tale soluzione ottima è unica? Giustificare la risposta.

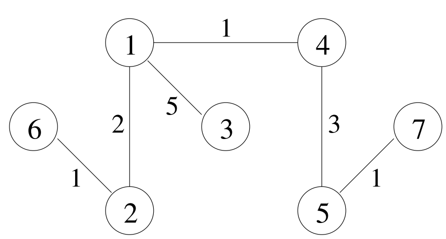


SVOLGIMENTO

Si consideri il seguente ordinamento degli archi per costo non decrescente:
$(1,4), (2,6), (5,7), (1,2), (1,6), (4,5), (2,5), (4,7), (1,3), (3,5), (2,3), (3,4)$.

	arco	operazione	taglio	ciclo
it.1)	$(1,4)$	inserzione	$(\{1\}, \{2,3,4,5,6,7\})$	
it.2)	$(2,6)$	inserzione	$(\{2\}, \{1,3,4,5,6,7\})$	
it.3)	$(5,7)$	inserzione	$(\{5\}, \{1,2,3,4,6,7\})$	
it.4)	$(1,2)$	inserzione	$(\{2,6\}, \{1,3,4,5,7\})$	
it.5)	$(1,6)$	cancellazione		$(1,6,2)$
it.6)	$(4,5)$	inserzione	$(\{5,7\}, \{1,2,3,4,6\})$	
it.7)	$(2,5)$	cancellazione		$(2,5,4,1)$
it.8)	$(4,7)$	cancellazione		$(4,7,5)$
it.9)	$(1,3)$	inserzione	$(\{3\}, \{1,2,4,5,6,7\})$	

L'algoritmo termina all'iterazione 9, con l'albero T in figura, in quanto $|A_T| = n - 1 = 6$.



La soluzione individuata non è l'unica soluzione ottima: una soluzione alternativa può essere ottenuta sostituendo l'arco $(1,3)$ con l'arco $(3,5)$ di ugual costo, il che accadrebbe all'ultima iterazione se si fosse scelto il corrispondente ordinamento degli archi.