

COMPITO DI RICERCA OPERATIVA II

APPELLO DEL 25/7/05

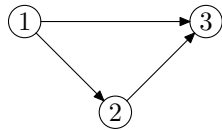
ESERCIZIO 1. (5 punti) Sia data la rete in figura con i valori associati ai tre nodi

$$b_1 = -5 \quad b_2 = 0 \quad b_3 = 5$$

e i seguenti costi unitari di trasporto sui tre archi

$$c_{12} = -2 \quad c_{13} = 4 \quad c_{23} = -3.$$

Si stabilisca, utilizzando il metodo due fasi, se esiste una soluzione ammissibile per il problema di flusso a costo minimo su tale rete.



ESERCIZIO 2. (6 punti) si consideri un problema di trasporto con tre depositi aventi disponibilità

$$a_1 = 20 \quad a_2 = 60 \quad a_3 = 20,$$

tre negozi con le seguenti richieste

$$b_1 = 30 \quad b_2 = 40 \quad b_3 = 50$$

e i seguenti costi unitari di trasporto

	<i>N1</i>	<i>N2</i>	<i>N3</i>
<i>D1</i>	1	4	5
<i>D2</i>	3	1	5
<i>D3</i>	8	7	3

Si determini una soluzione ottima e il valore ottimo di questo problema.

ESERCIZIO 3. (5 punti) Sia dato il problema di assegnamento con la seguente tabella dei costi (il simbolo $-$ si trova in corrispondenza di coppie che non si possono formare)

	b_1	b_2	b_3	b_4	b_5
a_1	$-$	6	8	5	6
a_2	10	8	6	4	4
a_3	6	9	3	7	10
a_4	8	5	$-$	6	$-$
a_5	6	11	$-$	2	2

Si determini una soluzione ottima e il valore ottimo di questo problema.

ESERCIZIO 4. (6 punti) Sia dato il problema KNAPSACK con capacità dello zaino $b = 7$ e con 5 oggetti aventi i seguenti valori v_i e pesi p_i e pesi

i	1	2	3	4	5
v_i	8	9	7	6	1
p_i	4	5	3	2	1

Si determini il valore ottimo e *tutte* le soluzioni ottime di questo problema utilizzando la programmazione dinamica.

ESERCIZIO 5. (6 punti) Dato il problema KNAPSACK dell'esercizio precedente, lo si inizi a risolvere con l'algoritmo branch-and-bound fermandosi quando si è calcolato il lower bound per i due nodi figli del nodo radice e si è scelto il nodo successivo su cui si farà branching.

ESERCIZIO 6. (3 punti) Ad una certa iterazione della risoluzione di un problema TSP mediante l'algoritmo branch-and-bound si ha che il valore di upper bound UB è pari a 15 e l'albero di branch-and-bound contiene 6 nodi foglia $S_1, S_3, S_6, S_7, S_9, S_{10}$ con i relativi lower bound:

$$L(S_1) = 18 \quad L(S_3) = 12 \quad L(S_6) = 15 \quad L(S_7) = 16 \quad L(S_9) = 14 \quad L(S_{10}) = 10.$$

Per ciascuna delle seguenti affermazioni dire se è vera o falsa (MOTIVARE LA RISPOSTA):

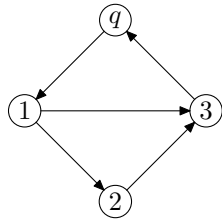
- (1) la soluzione ottima del problema TSP può trovarsi in uno qualsiasi degli attuali nodi foglia;
- (2) ogni nodo foglia contiene certamente almeno una soluzione ammissibile del problema TSP;
- (3) l'insieme delle soluzioni ottime del problema di assegnamento risolto per determinare un lower bound per il nodo S_9 può contenere un circuito hamiltoniano.

ESERCIZIO 7. (3 punti) Si consideri un problema di flusso massimo con capacità degli archi tutte intere e per il quale la soluzione ottima del corrispondente problema di taglio minimo è unica. Per ciascuna delle seguenti affermazioni dire se è vera o falsa (MOTIVARE LA RISPOSTA):

- (1) se riduco di una sola unità le capacità di tutti e soli gli archi che non fanno parte del taglio minimo, il valore ottimo del problema di flusso massimo non cambia;
- (2) se aumento di una unità la capacità di tutti e soli gli archi che fanno parte del taglio minimo, il valore ottimo del problema di flusso massimo aumenta di una quantità pari al numero di archi del taglio minimo;
- (3) se aumento di una unità la capacità di tutti gli archi, il valore ottimo del problema di flusso massimo aumenta di una quantità non superiore al numero di archi uscenti dal nodo sorgente S della rete.

SOLUZIONI

1. Si imposta il problema di prima fase aggiungendo un nodo q con $b_q = 0$ e gli archi $(q, 1)$, $(3, q)$.



Nel problema di prima fase agli archi $(1, 2)$, $(1, 3)$, $(2, 3)$ viene attribuito costo nullo, mentre agli archi $(q, 1)$, $(3, q)$ viene attribuito costo unitario. Una base iniziale valida per il problema è ora

$$B_0 = \{(q, 1), (3, q), (1, 2)\},$$

alla quale corrispondono valori di flusso

$$\begin{aligned} x_{q1} = 5, & \quad x_{3q} = 5, & \quad x_{12} = 0 & \quad (\text{variabili in base}), \\ x_{13} = 0, & \quad x_{23} = 0 & & \quad (\text{variabili fuori base}), \end{aligned}$$

e costi ridotti

$$\bar{c}_{13} = 2, \quad \bar{c}_{23} = 0.$$

La soluzione corrispondente a B_0 , di costo $z = 10 > 0$, è quindi ottima per il problema di prima fase. Il problema originale non ha quindi soluzioni ammissibili.

2. Il problema non è bilanciato, in quanto $\sum_i a_i - \sum_j b_j = 100 - 120 = -20$. Si aggiunge quindi un magazzino supplementare 4 con $a_4 = 20$.

	1	2	3	a_i
1	1	4	5	20
2	3	1	5	60
3	8	7	3	20
4	0	0	0	20
b_j	30	40	50	

Il metodo dell'angolo Nord-Ovest fornisce la base

$$B_0 = \{(1, 2), (2, 1), (2, 2), (2, 3), (3, 3), (3, 4)\}$$

	1	2	3	a_i
1	20			20
2	10	40	10	60
3			20	20
4			20	20
b_j	30	40	50	

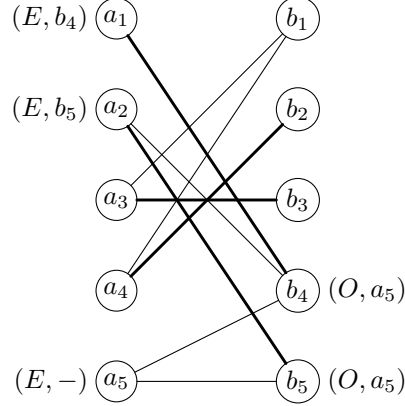
x_{ij}

	1	2	3	
1	0	5	2	
2	0	0	0	
3	7	8	0	
4	2	4	0	
	$\bar{c}_{ij}$			

Avendo $\bar{c}_{ij} \geq 0$ per ogni coppia (i, j) , la soluzione corrispondente a B_0 , di costo totale 200 risulta ottima.

3. Applicando l'algoritmo ungherese si perviene alla seguente matrice.

	b_1	b_2	b_3	b_4	b_5
a_1	∞	1	3	0	1
a_2	3	4	2	0	0
a_3	0	6	0	4	7
a_4	0	0	∞	1	∞
a_5	1	9	∞	0	0



L'accoppiamento $\Delta = \{(a_1, b_4), (a_2, b_5), (a_3, b_3), (a_4, b_2)\}$ ha cardinalità massima, ma non è un assegnamento ($|\Delta| < 5$). Si procede quindi all'aggiornamento dei costi ridotti.

	b_1	b_2	b_3	b_4	b_5
a_1	∞	1	3	0	1
a_2	3	4	2	0	0
a_3	0	6	0	4	7
a_4	0	0	∞	1	∞
a_5	1	9	∞	0	0

 $\Rightarrow$

	b_1	b_2	b_3	b_4	b_5
a_1	∞	0	2	0	1
a_2	2	3	1	0	0
a_3	0	6	0	5	8
a_4	0	0	∞	2	∞
a_5	0	8	∞	0	0

L'accoppiamento $\Delta = \{(a_1, b_4), (a_2, b_5), (a_3, b_3), (a_4, b_2), (a_5, b_1)\}$ corrisponde a un assegnamento ottimo di costo 23.

4. Calcolando gli stati della programmazione dinamica si ottengono i seguenti valori.

	0	1	2	3	4	5	6	7
1								15
2	0	1	6	7	8	13	14	15
3	0	1	6	7	8	13	14	14
4	0	1	6	7	7	7	7	7
5	0	1	1	1	1	1	1	1

	0	1	2	3	4	5	6	7
1								sn
2	no	no	no	no	no	no	no	si
3	no	no	no	sn	si	si	si	si
4	no	no	si	si	si	si	si	si
5	no	si	si	si	si	si	si	si

Per calcolare tutte le soluzioni ottime, occorre tener traccia dei casi nei quali $f_{k+1}^*(s) = v_k + f_{k+1}^*(s - p_k)$. Questi casi, etichettati con "sn" nella tabella $d_k^*(s)$, corrispondono a "biforcazioni", in corrispondenza delle quali entrambi le scelte si rivelano ottimali. Il backtracking deve esplorare tutte le biforcazioni che incontra, ottenendo le soluzioni ottime

$$S_1^* = \{1, 4, 5\},$$

$$S_2^* = \{1, 3\},$$

$$S_3^* = \{2, 4\},$$

tutte di valore 15.

5. Riordinando gli oggetti per valori decrescenti di v_i/p_i si ottiene la lista (4, 3, 1, 2, 5). Al nodo radice 0 risulta

$$UB_0 = v_4 + v_3 + \left\lfloor \frac{2}{4}v_1 \right\rfloor = 6 + 7 + \left\lfloor \frac{2}{4}8 \right\rfloor = 17,$$

$$LB_0 = v_4 + v_3 = 13 \Rightarrow LB = 13.$$

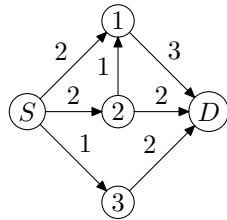
Nota: avendo tutti i v_i interi, è lecito arrotondare l'upper bound all'intero inferiore. Il branch si effettua sull'oggetto 1, ottenendo:

$$\begin{aligned} \text{Nodo 1: } I_0 = \{1\}, I_1 = \emptyset: \quad & \text{UB}_1 = 6 + 7 + \left\lfloor \frac{2}{5}9 \right\rfloor = 16, \\ & \text{LB}_1 = 6 + 7 = 13, \end{aligned}$$

$$\begin{aligned} \text{Nodo 2: } I_0 = \emptyset, I_1 = \{1\}: \quad & \text{UB}_2 = 8 + 6 + \left\lfloor \frac{1}{3}7 \right\rfloor = 16, \\ & \text{LB}_2 = 8 + 6 = 14 \implies \text{LB} = 14. \end{aligned}$$

Seguendo la strategia di esplorazione best-first, in questo caso il nodo da sottoporre a branch si può scegliere ad arbitrio in quanto $\text{UB}_1 = \text{UB}_2$. Se non si fossero arrotondati i bound, si avrebbe $\text{UB}_1 = 16.6$, $\text{UB}_2 = 16.3$, quindi si effettuerebbe il branch sul nodo 1.

6. (1) Falso. Non si trova sicuramente in S_1, S_7 , in quanto $L(S_1), L(S_7) > \text{UB}$.
- (2) Falso. Una foglia può contenere un assegnamento (e quindi un rilassamento valido di TSP) ma nessuna soluzione ammissibile.
- (3) Vero. Non ci sono dati per confutare questa possibilità.
7. (1) Falso. Nel seguente esempio il taglio indotto da $U = \{S\}$ è l'unico taglio minimo di capacità 5. Diminuendo di 1 tutte le capacità degli archi che non fanno parte del taglio minimo si genera il taglio indotto da $U' = \{S, 1, 2, 3\}$, di capacità 4.



- (2) Falso. Si può osservare sullo stesso esempio.
- (3) Falso. Si consideri la seguente rete, dove tale incremento produce un aumento del flusso massimo pari a 4, mentre solo due archi escono dalla sorgente.

