

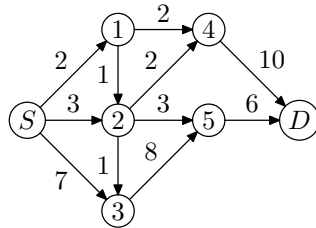
COMPITO DI RICERCA OPERATIVA II

ESERCIZIO 1. (6 punti) Sia data la rete rappresentata in figura con le capacità indicate su ciascun arco. Supponendo di partire dal flusso ammissibile iniziale

$$x_{S1} = 2 \quad x_{S2} = 2 \quad x_{S3} = 5 \quad x_{12} = 1 \quad x_{14} = 1$$

$$x_{23} = 1 \quad x_{24} = 2 \quad x_{25} = 0 \quad x_{35} = 6 \quad x_{4D} = 3 \quad x_{5D} = 6$$

si determini il flusso massimo e il taglio minimo per questa rete.



ESERCIZIO 2. (6 punti) Sia dato il problema del trasporto con le seguenti disponibilità nei 3 depositi

$$a_1 = 80 \quad a_2 = 30 \quad a_3 = 50,$$

le seguenti richieste nei 4 negozi

$$b_1 = 40 \quad b_2 = 30 \quad b_3 = 60 \quad b_4 = 30,$$

e i seguenti costi unitari di trasporto dai depositi ai negozi:

	N1	N2	N3	N4
D1	5	8	7	6
D2	1	5	8	7
D3	4	3	5	6

Si individui una base ammissibile iniziale per questo problema utilizzando la regola dell'angolo nord-ovest. Si stabilisca se la soluzione trovata è ottima oppure no. Nel caso non lo sia se ne trovi una migliore.

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	—	3	3	—	5
a_2	3	1	7	6	4
a_3	—	11	8	—	9
a_4	5	4	1	2	6
a_5	5	2	—	—	6

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

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

i	1	2	3	4
v_i	4	5	3	7
p_i	2	3	2	4

Si determini una soluzione ottima e il valore ottimo di questo problema utilizzando la programmazione dinamica.

ESERCIZIO 5. (6 punti) Sia dato il problema KNAPSACK con capacità dello zaino $b = 18$ e con 8 oggetti aventi i seguenti valori v_i e pesi p_i e pesi

i	1	2	3	4	5	6	7	8
v_i	8	7	9	3	8	9	11	7
p_i	5	6	4	2	3	7	5	4

Si calcoli l'upper bound per il sottinsieme della regione ammissibile $S(I_0, I_1)$ con $I_0 = \{2\}$ e $I_1 = \{3, 8\}$, si individui una soluzione ammissibile per il problema KNAPSACK dato e si effettui il branching di tale sottinsieme.

ESERCIZIO 6. (3 punti) Sia dato un grafo orientato completo $G = (V, A)$ con $|V| = n$. Sia v_{ij} la distanza associata ad ogni arco $(i, j) \in A$. Ricorda che il modello matematico del problema TSP è il seguente

$$\begin{aligned}
 \min \quad & \sum_{i,j \in V, i \neq j} v_{ij} x_{ij} \\
 & \sum_{i \in V, i \neq j} x_{ij} = 1 \quad \forall j \in V \\
 & \sum_{i \in V, i \neq j} x_{ji} = 1 \quad \forall j \in V \\
 & \sum_{i \in U, j \in V \setminus U} x_{ij} \geq 1 \quad \forall U \subseteq V : 2 \leq |U| \leq |V| - 2 \\
 & x_{ij} \in \{0, 1\} \quad \forall i, j \in \{1, \dots, n\}, i \neq j
 \end{aligned} \tag{1}$$

Ricordando la definizione di rilassamento di un problema dato, si dimostri che il seguente problema

$$\begin{aligned}
 \min \quad & \sum_{i,j \in V, i \neq j} v_{ij} x_{ij} \\
 & \sum_{i,j \in V, i \neq j} x_{ij} = n \\
 & x_{ij} \in \{0, 1\} \quad \forall i, j \in \{1, \dots, n\}, i \neq j
 \end{aligned} \tag{2}$$

(3)

è un rilassamento del problema TSP.

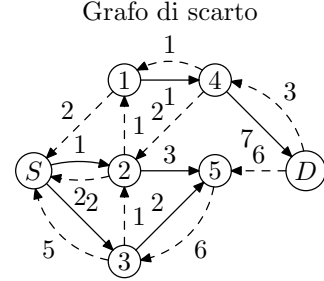
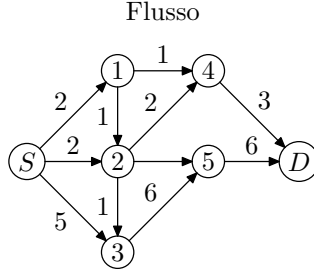
ESERCIZIO 7. (3 punti) Dato un grafo non orientato $G = (V, A)$ con peso $w_e > 0$ associato a ciascun arco $e \in A$, si consideri il problema di matching di peso massimo su tale grafo. Per ciascuna delle seguenti affermazioni dire se è sempre vera e, in caso contrario, mostrare un controesempio:

- (1) ogni matching M di peso massimo è massimale, ovvero non esiste un altro matching M' che contiene M ;
- (2) ogni matching M di peso massimo ha al più un arco in meno rispetto ad un matching di cardinalità massima sullo stesso grafo;
- (3) ogni matching M di peso massimo ha un numero di archi che è almeno la metà del numero di archi di un matching di cardinalità massima sullo stesso grafo.

SOLUZIONI

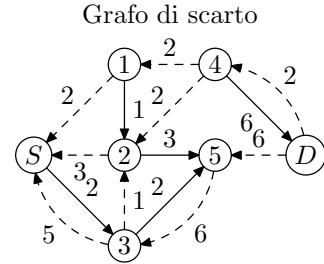
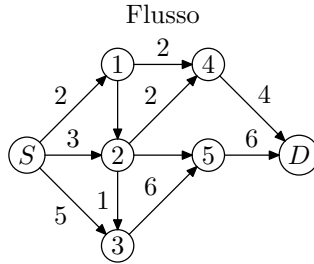
1. Impostando l'algoritmo di Ford-Fulkerson si ottiene quanto segue.

Iterazione 1.



Esiste un cammino aumentante $(S, 2, 1, 4, D)$ con $\delta = 1$.

Iterazione 2.



- Non esistono altri cammini aumentanti, quindi il flusso ottenuto — di valore $v = 10$ — è il massimo possibile. Il taglio minimo è identificato da $U = \{S, 2, 3, 5\}$, di capacità $C(U) = c_{S1} + c_{24} + c_{5D} = 10$.
2. Il problema è bilanciato: $\sum_i a_i = \sum_j b_j = 160$. Quindi si può procedere direttamente con il metodo dell'angolo Nord-Ovest, che fornisce la base iniziale B_0 .

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

	1	2	3	4	a_i
1	40 ⁻	30	10 ⁺		80
2	⁺		30 ⁻		30
3			20	30	50
	40	30	60	30	

x_{ij}

	1	2	3	4
1	0	0	0	-2
2	-5	-4	0	-2
3	1	-3	0	0

$\bar{c}_{ij}$

Il calcolo dei costi ridotti mostra che la soluzione non è ottima. Per ottenere una soluzione migliore si può effettuare il cambio di base $B_1 = B_0 + (2, 1) - (2, 3)$.

$$B_1 = \{(1, 1), (1, 2), (1, 3), (2, 1), (3, 3), (3, 4)\}$$

	1	2	3	4	a_i
1	10	30	40		80
2	30				30
3			20	30	50
	40	30	60	30	

x_{ij}

3. Utilizzando l'algoritmo ungherese si arriva alla seguente tabella.

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

$\bar{c}_{ij}$

Si può osservare che $\Delta = \{(a_1, b_3), (a_2, b_1), (a_3, b_5), (a_4, b_4), (a_5, b_2)\}$ è un accoppiamento con $|\Delta| = 5$, e $\bar{c}(\Delta) = 0$, quindi corrisponde a un assegnamento ottimo con valore ottimo pari a 19.

4. Utilizzando le relazioni ricorsive

$$f_k^*(s) = \max \{f_{k+1}^*(s), v_k + f_{k+1}^*(s - p_k)\}$$

si ottengono i seguenti valori.

		0	1	2	3	4	5	6
$f_k^*(s)$	1							11
	2	0	0	3	5	7	8	10
	3	0	0	3	3	7	7	10
	4	0	0	0	0	7	7	7

		0	1	2	3	4	5	6
$d_k^*(s)$	1							si
	2	no	no	no	si	no	si	no
	3	no	no	si	si	no	no	si
	4	no	no	no	no	si	si	si

Per backtracking (in grassetto i valori di $d_k^*(s)$ usati) si ottiene la soluzione ottima $S^* = \{1, 4\}$.

5. Per calcolare l'upper bound richiesto occorre considerare gli oggetti 1, 4, 5, 6, 7 non appartenenti a $i_0 \cup I_1$, e una capacità residua $b' = b - p(I_1) = 18 - 8$. Ordinando gli oggetti per rapporti v_i/p_i decrescenti si ottiene la sequenza (5, 7, 1, 4, 6). L'upper Bound richiesto è quindi

$$\text{UB}(I_0, I_1) = v(I_1) + v_5 + v_7 + \left\lfloor \frac{2}{5}v_1 \right\rfloor = 9 + 7 + 8 + 11 + \left\lfloor \frac{2}{5}8 \right\rfloor = 38.$$

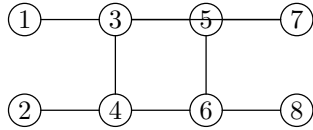
Una soluzione ammissibile è $I_1 \cup \{5, 7\}$ con valore pari a 35. Il branching genera due figli con rispettivamente $I_0 = \{1, 2\}$, $I_1 = \{3, 8\}$ e $I_0 = \{2\}$, $I_1 = \{1, 3, 8\}$.

6. Si chiami R il secondo problema, e siano S_{TSP} , S_R gli insiemi delle soluzioni ammissibili per i due problemi. Si può osservare che l'unico vincolo di R risulta dalla somma della prima serie di vincoli di TSP; quindi $x \in S_{\text{TSP}} \implies x \in S_R$, cioè $S_{\text{TSP}} \subseteq S_R$. Inoltre i due problemi hanno la stessa funzione obiettivo, quindi la relazione che lega i rispettivi valori ottimi $v(R)$, $v(\text{TSP})$ è

$$v^*(R) = \min \{v(x) : x \in S_R\} \leq \min \{v(x) : x \in S_{\text{TSP}}\} = v^*(\text{TSP}).$$

7. (1) Vero. Se un matching di peso massimo M fosse contenuto in un $M' \supset M$ si avrebbe $w(M') = w(M) + w(M' \setminus M)$ con $w(M' \setminus M) > 0$, contraddicendo l'ottimalità di M .

(2) Falso. Si consideri il seguente esempio, con $w_{34} = w_{56} = 10$, e $w_{ij} = 1$ per tutti gli altri archi. Il matching di peso massimo è $M^* = \{(3, 4), (5, 6)\}$, con $|M^*| = 2$, mentre esiste un matching di cardinalità pari almeno a 4, $M = \{(1, 3), (2, 4), (5, 7), (6, 8)\}$.



(3) Vero. Sia M un matching di peso massimo e M^* un matching di cardinalità massima. Per il punto (1) si sa che M deve essere massimale; inoltre si può vedere facilmente che anche M^* deve essere massimale. Sia $(i, j) \in M$; sono possibili due casi.

- (1) $(i, j) \in M^*$; l'arco è in comune ai due matching.
- (2) $(i, j) \notin M^*$; allora, siccome M è massimale deve esistere un arco $(u, i) \in M^* \setminus M$, oppure un $(v, j) \in M^* \setminus M$, oppure entrambi.

Quindi per questo arco (i, j) di M se ne possono contare uno o due in M^* . Ora, si rimuovano da G i nodi (i, j) e tutti gli archi incidenti: M^* , M sono ancora due matching validi (e massimali) per il grafo risultante, e si può ripetere il ragionamento. Si può continuare in questo modo fino a cancellare tutti gli archi di M ed M^* ; dal momento che alla rimozione di un arco di M corrisponde la rimozione di al massimo due archi di M^* , avremo che $2|M| \geq |M^*|$.