

SOLUZIONI

Esercizio 1

Partendo dalla base $B = [A_2 A_4 A_5]$ entra A_1 esce A_2 ;
 alla successiva iterazione entra A_3 esce A_5 . La base
 corrente è ottima.

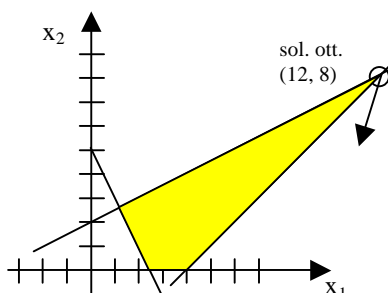
La soluzione iniziale è: $x^T = (0 \ 3 \ 0 \ 2 \ 11)$

La soluzione ottima è: $x^T = (5 \ 0 \ 2 \ 10 \ 0)$

$$\min \quad -x_1 + 3x_2$$

$$\begin{cases} x_1 + x_2 - x_3 = 3 \\ -2x_1 + x_3 + x_4 = 2 \\ 3x_1 - 2x_3 + x_5 = 11 \\ x \geq 0 \end{cases}$$

Esercizio 2



$$\min \quad -x_1 - 3x_2$$

$$\begin{cases} x_1 - x_2 \leq 4 \\ 2x_1 + x_2 \geq 5 \\ -x_1 + 2x_2 \leq 4 \\ x_1, x_2 \geq 0 \end{cases}$$

Esercizio 3

Dal problema duale in figura si ottiene la condizione di
 ortogonalità $u_2 = -4$ che è incompatibile con il segno
 di u_2 ; per cui non esiste soluzione ottima tale che
 $x_2 > 0$. In effetti il duale è impossibile ed il primale
 illimitato, per cui non esiste alcuna soluzione ottima.

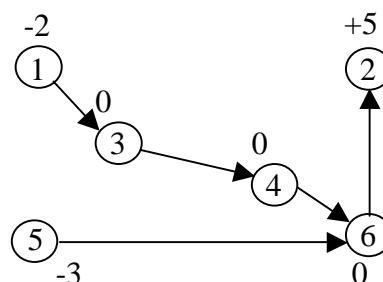
$$\max \quad 4u_1 + 2u_2$$

$$\begin{cases} -u_1 \geq 2 \\ u_2 \leq -4 \\ u_1 + u_2 = 5 \\ u_1 \leq 0, \quad u_2 \geq 0 \end{cases}$$

Esercizio 4

A partire dalla base in figura entra (4,5) ed esce (4,6). Alla
 successiva iterazione entra (5,1) [oppure (2,3)] ed il problema
 risulta illimitato.

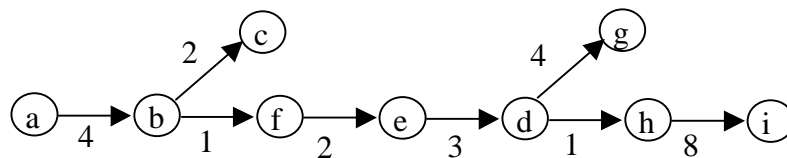
Archi	(1,3)	(2,1)	(2,3)	(3,4)	(4,5)	(4,6)	(5,1)	(5,6)	(6,2)
Costi	-3	5	4	-6	2	4	-2	-3	1
Flussi	2	0	0	2	0	2	0	3	5



Esercizio 5

In tabella è riportato il costo di percorrenza degli archi di un grafo con 9 nodi **a...i**. Trovare l'albero dei cammini
 minimi dal nodo **a** a tutti gli altri nodi utilizzando l'algoritmo di Dijkstra. Evidenziare il cammino minimo dal nodo **a** al
 nodo **i**.

Archi	(a,b)	(a,c)	(a,e)	(b,c)	(b,e)	(b,f)	(c,d)	(d,h)	(d,g)	(e,c)	(e,h)	(e,d)	(e,g)	(f,a)	(f,d)	(f,e)	(f,g)	(g,h)	(g,i)	(h,i)
Costi	4	7	9	2	4	1	7	1	4	1	10	3	10	4	9	2	10	6	12	8



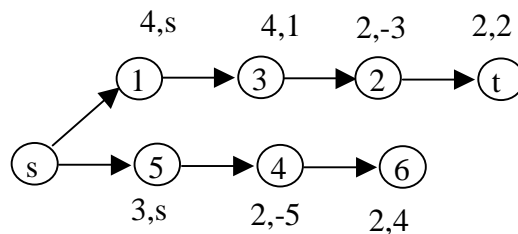
In figura è rappresentato l'albero dei cammini minimi dal nodo **a** a tutti gli altri nodi. In particolare il cammino minimo da **a** a **i** è il cammino (**a,b,f,e,d,h,i**) di costo 19.

Esercizio 6

In tabella sono riportati gli archi di un grafo con 8 nodi, e sono dati i valori di capacità degli archi ed un flusso ammissibile. A partire dal flusso dato trovare il massimo flusso inviabile da **s** a **t** con l'algoritmo di Ford e Fulkerson.

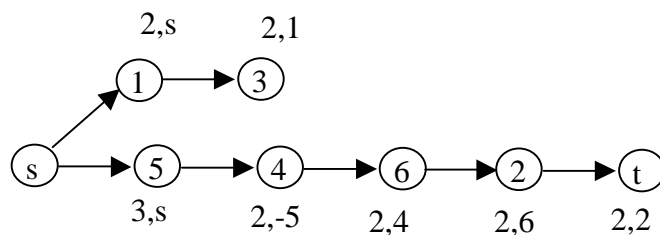
Archi	(s,1)	(s,5)	(1,2)	(1,3)	(2,3)	(2,t)	(3,4)	(4,5)	(4,6)	(5,1)	(5,6)	(6,2)	(6,t)
Capacità	6	3	2	9	4	6	2	4	3	5	2	5	2
Flussi	2	0	2	0	2	0	2	2	0	0	2	0	2

La ricerca del primo cammino aumentante porta a costruire il seguente albero:



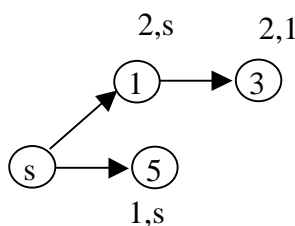
Archi	(s,1)	(s,5)	(1,2)	(1,3)	(2,3)	(2,t)	(3,4)	(4,5)	(4,6)	(5,1)	(5,6)	(6,2)	(6,t)
Capacità	6	3	2	9	4	6	2	4	3	5	2	5	2
Flussi	4	0	2	2	0	2	2	2	0	0	2	0	2

La ricerca del secondo cammino aumentante porta a costruire il seguente albero:



Archi	(s,1)	(s,5)	(1,2)	(1,3)	(2,3)	(2,t)	(3,4)	(4,5)	(4,6)	(5,1)	(5,6)	(6,2)	(6,t)
Capacità	6	3	2	9	4	6	2	4	3	5	2	5	2
Flussi	4	2	2	2	0	4	2	0	2	0	2	2	2

La ricerca del terzo cammino aumentante porta a costruire il seguente albero che individua un taglio di capacità 6, pari alla capacità degli archi uscenti dal taglio: (1,2),(3,4),(5,6). Il Flusso attuale è dunque massimo.



Esercizio 7

In tabella sono riportate le 14 attività di un progetto, con durate e vincoli di precedenza tra attività. Formulare come problema di PL il problema di determinare la durata minima del progetto, senza risolvere il problema stesso.

Attività	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀	A ₁₁	A ₁₂	A ₁₃	A ₁₄
Durata	7	9	14	16	15	14	8	3	5	7	3	9	11	5
Predecessori	-	-	-	A ₁	A ₁	A ₁	A ₄	A ₅	A ₄	A ₉	A ₇	A ₁₁	A ₈	A ₁₀
				A ₃	A ₂	A ₃	A ₆		A ₆		A ₁₀		A ₉	A ₁₃

Basta scegliere come variabili l'istante di inizio di ogni attività ($x_1 \dots x_{14}$), più una variabile x_{15} per indicare la fine del progetto. Si ottiene così un problema con 15 variabili. La funzione obiettivo è semplicemente la minimizzazione di x_{15} , mentre i vincoli devono solo imporre che una attività A_i non può iniziare se un'attività A_h che la precede non è stata completata (cioè $x_i - x_h \geq d_h$, se d_h è la durata dell'attività A_h), includendo anche il fatto che la fine del progetto avviene solo dopo che tutte le attività prive di successori (A_{12} e A_{14}) sono state completate. Infine nessuna attività può iniziare prima del tempo 0. Si ottiene un problema con 19 vincoli del tipo $x_i - x_h \geq d_h$, per ogni A_h predecessore di A_i . Ci sono poi i vincoli $x_i \geq 0$, per ogni attività A_i .

min x_{15}

$$\begin{cases} x_4 - x_1 \geq 7 & x_4 - x_3 \geq 14 & x_5 - x_1 \geq 7 & x_5 - x_2 \geq 15 & x_6 - x_1 \geq 7 & x_6 - x_3 \geq 14 \\ x_7 - x_4 \geq 16 & x_7 - x_6 \geq 14 & x_8 - x_5 \geq 15 & x_9 - x_4 \geq 16 & x_9 - x_6 \geq 14 & x_{10} - x_9 \geq 5 \\ x_{11} - x_7 \geq 8 & x_{11} - x_9 \geq 5 & x_{12} - x_{11} \geq 3 & x_{13} - x_8 \geq 3 & x_{13} - x_9 \geq 5 & x_{14} - x_{10} \geq 7 \\ x_{14} - x_{13} \geq 11 & x_{15} - x_{12} \geq 9 & x_{15} - x_{14} \geq 5 \\ x \geq 0 \end{cases}$$

Domanda 8

Illustrare i problemi di programmazione convessa, dimostrando in particolare che in questi problemi un punto di minimo locale è punto di minimo globale.

SOLUZIONI

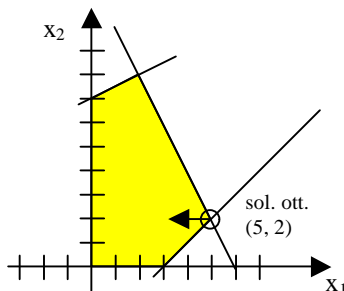
Esercizio 1

Partendo dalla base $B = [A_5 A_3 A_2]$ entra A_1 esce A_5 ;
alla successiva iterazione entra A_4 esce A_3 . La base
corrente è ottima.
La soluzione iniziale è: $x^T = (0 \ 4 \ 6 \ 0 \ 12)$
La soluzione ottima è: $x^T = (6 \ 2 \ 0 \ 4 \ 0)$

$$\begin{aligned} \min \quad & x_2 - 2x_4 \\ \text{s.t.} \quad & \begin{cases} 4x_1 - 3x_4 + x_5 = 12 \\ -x_1 + x_3 + 3x_4 = 6 \\ x_1 + x_2 - x_4 = 4 \\ x \geq 0 \end{cases} \end{aligned}$$

Esercizio

$$\begin{aligned} \min \quad & -x_1 \\ \text{s.t.} \quad & \begin{cases} x_1 - x_2 \leq 3 \\ 2x_1 + x_2 \leq 12 \\ -x_1 + 2x_2 \leq 14 \\ x_1, x_2 \geq 0 \end{cases} \end{aligned}$$



Esercizio 3

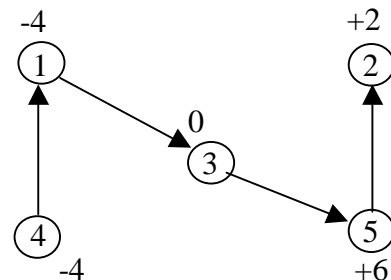
Dal problema duale in figura, e applicando le
condizioni di ortogonalità, si ottiene la soluzione
ammissibile duale $u^T = (0 \ 1)$. Pertanto la soluzione
 $x^T = (1 \ -5 \ 0)$ è soluzione ottima del problema
primale.

$$\begin{aligned} \min \quad & -3u_1 + 6u_2 \\ \text{s.t.} \quad & \begin{cases} 2u_1 + u_2 \geq 1 \\ u_1 - u_2 = -1 \\ -u_1 \leq 1 \\ u_2 \geq 0 \end{cases} \end{aligned}$$

Esercizio 4

A partire dalla base data (in figura) entra (1,2) ed esce (5,2) scarico; alla
successiva iterazione entra (5,4). Poiché tutti gli archi del ciclo sono
concordi, il problema è illimitato inferiormente. Infatti il ciclo 1,3,5,4,1
ha costo $-4 -2 +3 +1 = -2 < 0$.

Archì	(1,2)	(1,3)	(2,3)	(3,4)	(3,5)	(4,1)	(5,2)	(5,4)
Capacità	3	14	4	6	10	11	6	7
Costi	2	-4	4	3	-2	1	16	3
Flussi	0	8	0	0	8	4	2	0



Esercizio 5

Un progetto richiede 14 attività, con durate e vincoli di precedenza dati in tabella. Determinare la durata minima del
progetto, evidenziando le attività critiche. Esprimere, per tutte le attività non critiche, l'intervallo [minimo inizio,
massima fine].

Attività	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀	A ₁₁	A ₁₂	A ₁₃	A ₁₄
Durata	7	9	12	16	15	14	6	3	5	7	8	9	11	6
Predecessori	-	-	-	A ₁ A ₂ A ₃	A ₁ A ₂	A ₁ A ₂	A ₄ A ₆	A ₅ A ₆	A ₆	A ₉	A ₇	A ₁₀ A ₁₁	A ₈ A ₉	A ₁₀ A ₁₃

Attività	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀	A ₁₁	A ₁₂	A ₁₃	A ₁₄	Fine
Durata	7	9	12	16	15	14	6	3	5	7	8	9	11	6	
Min iniz.	0	0	0	9	12	9	25	27	23	28	31	39	30	41	48
Max fine	9	9	13	25	28	25	31	31	31	39	39	48	42	48	
Critica		SI		SI			SI				SI	SI			

Esercizio 6

State applicando l'algoritmo di Floyd e Warshall ad un grafo con 5 nodi. Alla fine del passo 3 ottenete le matrici in figura (quella di sinistra indica i cammini minimi, quella di destra i predecessori). Effettuate i passi 4 e 5 dell'algoritmo e mostrare i cammini minimi dal nodo 3 al nodo 2 e dal nodo 5 al nodo 4. In presenza di cicli negativi arrestate l'algoritmo e mostrate un ciclo negativo.

0	2	3	6	-1
∞	0	1	4	-3
∞	∞	0	3	∞
∞	8	9	0	2
-2	0	1	4	0

1	1	2	3	2
2	2	2	3	2
3	3	3	3	3
4	4	2	4	4
5	1	2	3	5

Alla fine del passo 4 si trovano le matrici

0	2	3	6	-1
∞	0	1	4	-3
∞	11	0	3	5
∞	8	9	0	2
-2	0	1	4	0

1	1	2	3	2
2	2	2	3	2
3	4	3	3	4
4	4	2	4	4
5	1	2	3	5

Alla fine del passo 5 si trovano le matrici

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

5	1	2	3	2
5	1	2	3	2
5	1	3	3	4
5	1	2	4	4
5	1	2	3	5

Che rivelano la presenza dei cicli negativi (1,2,5,1) e (2,5,1,2), per cui l'algoritmo si ferma.

Esercizio 7

In tabella sono riportati gli archi di un grafo con 8 nodi, e sono dati i valori di capacità degli archi. Formulare come problema di PL il problema di inviare il massimo flusso dal nodo s al nodo t, senza risolvere il problema

Archì	(s,1)	(s,5)	(1,4)	(1,3)	(2,5)	(2,t)	(3,6)	(4,5)	(4,2)	(5,1)	(5,6)	(6,2)	(6,t)
Capacità	6	3	5	9	3	6	2	4	6	5	2	7	2

Un problema di massimo flusso può essere facilmente formulato aggiungendo un arco fittizio di capacità infinita dal nodo t al nodo s, introducendo una variabile di flusso per ogni arco e massimizzando il flusso sull'arco fittizio. I vincoli devono semplicemente imporre la non negatività dei flussi su ciascun arco, il rispetto dei vincoli di capacità, e l'equilibrio dei flussi ad ogni nodo. Si ottiene:

$$\text{variabili: } x^T = (x_{s1} \ x_{s5} \ x_{14} \ x_{13} \ x_{25} \ x_{2t} \ x_{36} \ x_{45} \ x_{42} \ x_{51} \ x_{56} \ x_{62} \ x_{6t} \ x_{ts})$$

$$\max \quad x_{ts}$$

$$\begin{cases} x_{ts} - x_{s1} - x_{s5} = 0 \\ x_{s1} + x_{51} - x_{14} - x_{13} = 0 \\ x_{42} + x_{62} - x_{25} - x_{2t} = 0 \\ x_{13} - x_{36} = 0 \\ x_{14} - x_{45} - x_{42} = 0 \\ x_{s5} + x_{25} + x_{45} - x_{51} - x_{56} = 0 \\ x_{36} + x_{56} - x_{62} - x_{6t} = 0 \\ x_{2t} + x_{6t} - x_{ts} = 0 \\ 0^T \leq x^T \leq (6 \ 3 \ 5 \ 9 \ 3 \ 6 \ 2 \ 4 \ 6 \ 5 \ 2 \ 7 \ 2 \ \infty) \end{cases}$$

Domanda 8

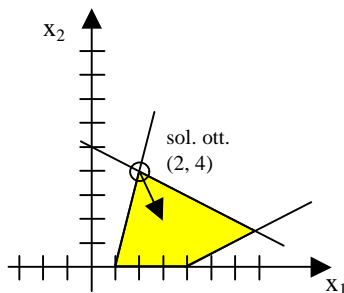
Illustrare la teoria della dualità, dimostrando che i problemi di programmazione lineare godono della proprietà di dualità forte.

SOLUZIONI

Esercizio 1

Partendo dalla base $B = [A_3 A_2 A_5]$ entra A_4 esce A_3 ;
alla successiva iterazione entra A_1 esce A_5 .
All'iterazione successiva entra A_3 e il problema risulta
illimitato inferiormente.
La soluzione iniziale è: $x^T = (0 \ 2 \ 3 \ 0 \ 10)$

$$\begin{aligned} \min \quad & x_3 - x_4 \\ \text{s.t.} \quad & \begin{cases} -x_1 + x_3 + x_4 = 3 \\ x_1 + x_2 - 2x_4 = 2 \\ -2x_1 + 3x_4 + x_5 = 10 \\ x \geq 0 \end{cases} \end{aligned}$$



Esercizio 2

$$\begin{aligned} \min \quad & x_1 - 2x_2 \\ \text{s.t.} \quad & \begin{cases} 4x_1 - x_2 \geq 4 \\ x_1 + 2x_2 \leq 10 \\ x_1 - x_2 \leq 4 \\ x_1, x_2 \geq 0 \end{cases} \end{aligned}$$

Esercizio 3

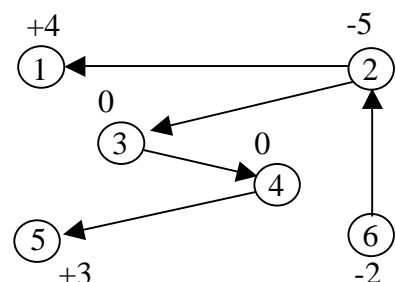
Dal problema duale in figura, e applicando le
condizioni di ortogonalità, si ottiene la soluzione
ammissibile duale $u^T = (0 \ 1)$. Pertanto la soluzione
 $x^T = (0 \ 0 \ 3)$ è soluzione ottima del problema
primale.

$$\begin{aligned} \max \quad & 5u_1 + 3u_2 \\ \text{s.t.} \quad & \begin{cases} u_1 \leq 2 \\ u_2 \geq -4 \\ u_1 + u_2 = 1 \\ u_1 \leq 0 \quad u_2 \geq 0 \end{cases} \end{aligned}$$

Esercizio 4

A partire dalla base data (in figura) entra (1,3) ed esce (2,3); alla
successiva iterazione entra (6,5) ed esce (6,2). La soluzione ottenuta (in
tabella) è ottima.

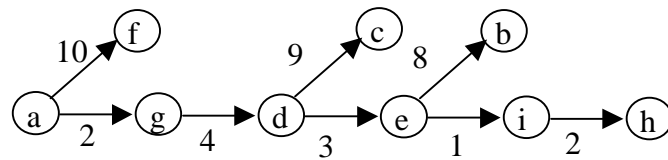
Archi	(1,3)	(2,1)	(2,3)	(3,4)	(4,5)	(4,6)	(5,1)	(6,2)	(6,5)
Costi	2	-6	1	3	-6	-2	8	14	2
Flussi	0	4	3	3	3	0	0	2	0
Sol ott.	1	5	0	1	1	0	0	0	2



Esercizio 5

In tabella è riportato il costo di percorrenza degli archi di un grafo con 9 nodi **a...i**. Trovare l'albero dei cammini
minimi dal nodo **a** a tutti gli altri nodi utilizzando l'algoritmo di Dijkstra. Evidenziare il cammino minimo dal nodo **a** al
nodo **h**.

Archi	(a,f)	(a,g)	(b,a)	(b,c)	(b,f)	(c,h)	(d,c)	(d,e)	(d,h)	(d,i)	(e,b)	(e,f)	(e,i)	(f,c)	(g,d)	(g,e)	(h,f)	(i,h)
Costi	10	2	1	1	1	1	9	3	15	14	8	2	1	12	4	8	1	2



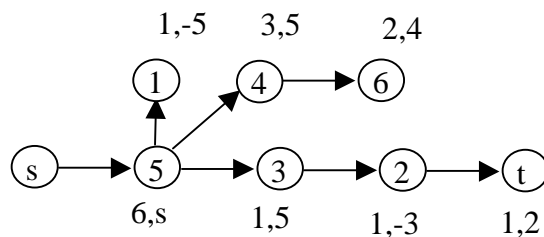
In figura è rappresentato l'albero dei cammini minimi dal nodo **a** a tutti gli altri nodi. In particolare il cammino minimo da **a** a **h** è il cammino **(a,g,d,e,i,h)** di costo 12.

Esercizio 6

In tabella sono riportati gli archi di un grafo con 8 nodi, e sono dati i valori di capacità degli archi ed un flusso ammissibile. A partire dal flusso dato trovare il massimo flusso inviabile da **s** a **t** con l'algoritmo di Ford e Fulkerson.

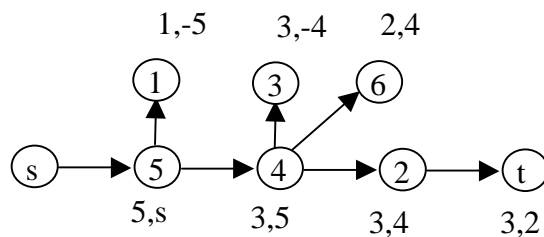
Archi	(s,1)	(s,5)	(1,2)	(1,5)	(2,3)	(2,t)	(3,1)	(3,4)	(4,2)	(4,6)	(5,3)	(5,4)	(6,t)
Capacità	3	6	2	2	5	8	4	3	5	5	2	3	3
Flussi	3	0	2	1	2	0	0	3	0	3	1	0	3

La ricerca del primo cammino aumentante porta a costruire il seguente albero:



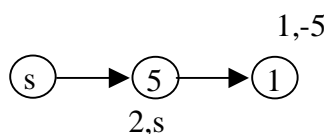
Archi	(s,1)	(s,5)	(1,2)	(1,5)	(2,3)	(2,t)	(3,1)	(3,4)	(4,2)	(4,6)	(5,3)	(5,4)	(6,t)
Capacità	3	6	2	2	5	8	4	3	5	5	2	3	3
Flussi	3	1	2	1	1	1	0	3	0	3	2	0	3

La ricerca del secondo cammino aumentante porta a costruire il seguente albero:



Archi	(s,1)	(s,5)	(1,2)	(1,5)	(2,3)	(2,t)	(3,1)	(3,4)	(4,2)	(4,6)	(5,3)	(5,4)	(6,t)
Capacità	3	6	2	2	5	8	4	3	5	5	2	3	3
Flussi	3	4	2	1	1	4	0	3	3	3	2	3	3

La ricerca del terzo cammino aumentante porta a costruire il seguente albero che individua un taglio di capacità 7, pari alla capacità degli archi uscenti dal taglio: (1,2),(5,3),(5,4). Il Flusso attuale è dunque massimo.



Esercizio 7

In tabella sono riportate le 14 attività di un progetto, con durate e vincoli di precedenza tra attività. Formulare come problema di PL il problema di determinare la durata minima del progetto, senza risolvere il problema stesso.

Attività	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀	A ₁₁	A ₁₂	A ₁₃	A ₁₄
Durata	3	8	9	9	12	7	4	8	6	10	6	4	17	3
Predecessori	-	-	A ₁	A ₁	A ₁	A ₁	A ₅	A ₇	A ₃	A ₄	A ₉	A ₉	A ₂	A ₉
						A ₂		A ₆	A ₅	A ₇		A ₁₁		A ₁₃
						A ₃								

Basta scegliere come variabili l'istante di inizio di ogni attività ($x_1 \dots x_{14}$), più una variabile x_{15} per indicare la fine del progetto. Si ottiene così un problema con 15 variabili. La funzione obiettivo è semplicemente la minimizzazione di x_{15} , mentre i vincoli devono solo imporre che una attività A_i non può iniziare se un'attività A_h che la precede non è stata completata (cioè $x_i - x_h \geq d_h$, se d_h è la durata dell'attività A_h), includendo anche il fatto che la fine del progetto avviene solo dopo che tutte le attività prive di successori (A_8, A_{10}, A_{12} e A_{14}) sono state completate. Infine nessuna attività può iniziare prima del tempo 0. Si ottiene un problema con 19 vincoli del tipo $x_i - x_h \geq d_h$, per ogni A_h predecessore di A_i . Ci sono poi i vincoli $x_i \geq 0$, per ogni attività A_i .

min x_{15}

$$\begin{cases} x_3 - x_1 \geq 3 & x_4 - x_1 \geq 3 & x_5 - x_1 \geq 3 & x_6 - x_1 \geq 3 & x_6 - x_2 \geq 8 & x_6 - x_3 \geq 9 \\ x_7 - x_5 \geq 12 & x_8 - x_6 \geq 7 & x_8 - x_7 \geq 4 & x_9 - x_3 \geq 9 & x_9 - x_5 \geq 9 & x_{10} - x_4 \geq 9 \\ x_{10} - x_7 \geq 4 & x_{11} - x_9 \geq 6 & x_{12} - x_9 \geq 6 & x_{12} - x_{11} \geq 6 & x_{13} - x_2 \geq 8 & x_{14} - x_9 \geq 6 \\ x_{14} - x_{13} \geq 17 & x_{15} - x_8 \geq 8 & x_{15} - x_{10} \geq 10 & x_{15} - x_{12} \geq 4 & x_{15} - x_{14} \geq 3 \\ x \geq 0 \end{cases}$$

Domanda 8

Illustrare i problemi di flusso su reti e descrivere l'algoritmo del simplesso su reti non capacitate, illustrando in dettaglio le particolarità di questo rispetto al simplesso rivisto.

SOLUZIONI

Esercizio 1

Partendo dalla base $B = [A_5 A_3 A_2]$ entra A_1 esce A_5 ;
alla successiva iterazione entra A_4 esce A_3 . La base
corrente è ottima.

La soluzione iniziale è: $x^T = (0 \ 14 \ 12 \ 0 \ 3)$

La soluzione ottima è: $x^T = (5 \ 15 \ 0 \ 2 \ 0)$

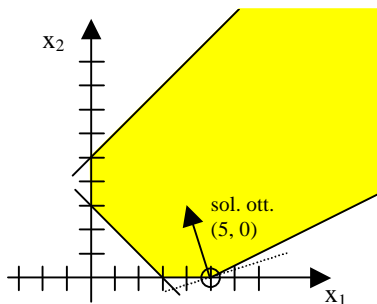
$$\min \quad x_1 + x_3 + x_4$$

$$\begin{cases} x_1 - x_4 + x_5 = 3 \\ 2x_1 + x_3 + x_4 = 12 \\ -x_1 + x_2 + 2x_4 = 14 \\ x \geq 0 \end{cases}$$

Esercizio 2

$$\min \quad -x_1 + 3x_2$$

$$\begin{cases} x_1 + x_2 \geq 3 \\ -x_1 + x_2 \leq 5 \\ x_1 - 2x_2 \leq 5 \\ x_1, x_2 \geq 0 \end{cases}$$



Esercizio 3

$$\min \quad 6u_1 + 3u_2$$

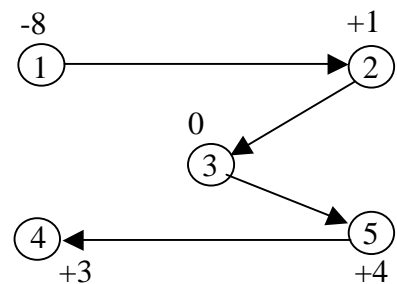
$$\begin{cases} 3u_1 + u_2 \geq 1 \\ -2u_1 + u_2 = 1 \\ u_1 + u_2 \leq 0 \\ u_1 \geq 0 \end{cases}$$

Dal problema duale in figura, e applicando le
condizioni di ortogonalità, si ottiene la soluzione duale
 $u^T = (0 \ 1)$ che viola il terzo vincolo duale. Pertanto
la soluzione $x^T = (2 \ 1 \ 0)$ NON è soluzione
ottima del problema primale.

Esercizio 4

A partire dalla base data (in figura) entra (1,3) scarico ed esce (1,3) saturo;
alla successiva iterazione entra (3,4) scarico ed esce (3,4) saturo; alla
successiva iterazione entra (4,1) scarico ed esce (4,1) saturo. La soluzione
così ottenuta (in tabella) è ottima.

Archi	(1,2)	(1,3)	(2,3)	(3,4)	(3,5)	(4,1)	(5,2)	(5,4)
Capacità	10	2	9	2	12	1	4	7
Costi	4	3	3	-2	1	-10	4	1
Flussi	8	0	7	0	7	0	0	3
Sol. Ott.	7	2	6	2	6	1	0	2



Esercizio 5

Un progetto richiede 14 attività, con durate e vincoli di precedenza dati in tabella. Determinare la durata minima del progetto, evidenziando le attività critiche. Esprimere, per tutte le attività non critiche l'intervallo [minimo inizio, massima fine].

Attività	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀	A ₁₁	A ₁₂	A ₁₃	A ₁₄
Durata	4	8	10	9	10	7	4	6	6	10	6	3	17	8
Predecessori	-	-	A ₁	A ₁	A ₁	A ₁	A ₅	A ₇	A ₃	A ₄	A ₉	A ₉	A ₂	A ₈
						A ₂	A ₆		A ₇	A ₇		A ₁₀		A ₉
														A ₁₃

Attività	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀	A ₁₁	A ₁₂	A ₁₃	A ₁₄	Fine
Durata	4	8	10	9	10	7	4	6	6	10	6	3	17	8	
Min iniz.	0	0	4	4	4	8	15	19	19	19	25	29	8	25	33
Max fine	5	8	19	20	15	15	19	25	25	30	33	33	25	33	
Critica		SI				SI	SI	SI	SI				SI	SI	

Esercizio 6

State applicando l'algoritmo di Floyd e Warshall ad un grafo con 5 nodi. Alla fine del passo 3 ottenete le matrici in figura (quella di sinistra indica i cammini minimi, quella di destra i predecessori). Effettuate i passi 4 e 5 dell'algoritmo e mostrare i cammini minimi dal nodo 3 al nodo 2 e dal nodo 5 al nodo 4. In presenza di cicli negativi arrestate l'algoritmo e mostrate un ciclo negativo.

0	∞	∞	-1	4
1	0	∞	0	-3
-7	-8	0	-4	-11
-1	-2	6	0	-5
5	4	∞	6	0

1	1	1	1	1
2	2	2	1	2
2	3	3	2	2
2	3	4	4	2
2	3	5	5	5

Alla fine del passo 4 si trovano le matrici

-2	-3	5	-1	-6
-1	-2	6	0	-5
-7	-8	0	-4	-11
-1	-2	6	0	-5
5	4	12	6	0

2	3	4	1	2
2	3	4	1	2
2	3	3	2	2
2	3	4	4	2
2	3	4	5	5

Che rivelano la presenza dei cicli negativi (1,4,3,2,1) e (2,1,4,3,2), per cui l'algoritmo si ferma.

Esercizio 7

In tabella sono riportati gli archi di un grafo con 8 nodi, e sono dati i valori di capacità degli archi. Formulare come problema di PL il problema di inviare il massimo flusso dal nodo **s** al nodo **t**, senza risolvere il problema

Archì	(s,1)	(s,5)	(1,2)	(1,3)	(2,3)	(2,t)	(3,4)	(3,6)	(4,2)	(4,3)	(5,2)	(5,4)	(6,t)
Capacità	2	6	2	2	4	8	4	3	7	5	2	1	3

Un problema di massimo flusso può essere facilmente formulato aggiungendo un arco fittizio di capacità infinita dal nodo **t** al nodo **s**, introducendo una variabile di flusso per ogni arco e massimizzando il flusso sull'arco fittizio. I vincoli devono semplicemente imporre la non negatività dei flussi su ciascun arco, il rispetto dei vincoli di capacità, e l'equilibrio dei flussi ad ogni nodo. Si ottiene:

$$\text{variabili: } x^T = (x_{s1} \ x_{s5} \ x_{12} \ x_{13} \ x_{23} \ x_{2t} \ x_{34} \ x_{36} \ x_{42} \ x_{43} \ x_{52} \ x_{54} \ x_{6t} \ x_{ts})$$

$$\max \quad x_{ts}$$

$$\begin{cases} x_{ts} - x_{s1} - x_{s5} = 0 \\ x_{s1} - x_{12} - x_{13} = 0 \\ x_{12} + x_{42} + x_{52} - x_{23} - x_{2t} = 0 \\ x_{13} + x_{23} + x_{43} - x_{34} - x_{36} = 0 \\ x_{34} + x_{54} - x_{42} - x_{43} = 0 \\ x_{s5} - x_{52} - x_{54} = 0 \\ x_{36} - x_{6t} = 0 \\ x_{2t} + x_{6t} - x_{ts} = 0 \\ 0^T \leq x^T \leq (2 \ 6 \ 2 \ 2 \ 4 \ 8 \ 4 \ 3 \ 7 \ 5 \ 2 \ 1 \ 3 \ \infty) \end{cases}$$

Domanda 8

Illustrare gli algoritmi su grafo visti nel corso, dimostrando in particolare che l'algoritmo di Dijkstra trova una soluzione ottima per il problema di cammino minimo.