**Laurea in Informatica**
**A.A. 03/04**
**Corso di RICERCA OPERATIVA***

**CS - 21 giugno 2004**

**Esercizio n.1** (punti: 6)
In una centrale di polizia si deve provvedere alla riorganizzazione dei turni giornalieri, ciascuno dei quale è composto da 4 ore. Ogni poliziotto deve lavorare giornalmente per due turni alternati: ad esempio, chi inizia a lavorare al primo turno lavorerà anche al terzo, chi inizia a lavorare al secondo turno lavorerà anche al quarto, e così via. Nella seguente tabella, per ogni turno, sono riportati il numero minimo di poliziotti che devono essere presenti in quel turno:

| Turno | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Orario | 0-4 | 4-8 | 8-12 | 12-16 | 16-20 | 20-24 |
| Numero minimo | 30 | 25 | 50 | 40 | 60 | 35 |

Ad esempio, nel turno 4, cioè dalle ore 12 alle ore 16, devono essere in servizio almeno 40 poliziotti.
La retribuzione dei poliziotti è di 14 euro all'ora per i turni diurni (3, 4, 5 e 6) e di 20 euro all'ora per i turni notturni (1 e 2). Formulare il problema come problema di ottimizzazione, con l'obiettivo di minimizzare le spese giornaliere di retribuzione dei poliziotti.
*Risoluzione*

- Variabili decisionali: $x_i$=numero di poliziotti che iniziano a lavorare nel turno $i$, con $i = 1, \dots, 6$.
- funzione obiettivo:

$$\begin{array}{l} \min_x z = (4 \cdot 20 + 4 \cdot 14)x_1 + (4 \cdot 20 + 4 \cdot 14)x_2 + (4 \cdot 14 + 4 \cdot 14)x_3 + \\ (4 \cdot 14 + 4 \cdot 14)x_4 + (4 \cdot 20 + 4 \cdot 14)x_5 + (4 \cdot 20 + 4 \cdot 14)x_6 \end{array}$$

- vincoli:

$$\begin{array}{l} x_1 + x_5 \geq 30 \\ x_2 + x_6 \geq 25 \\ x_3 + x_1 \geq 50 \\ x_4 + x_2 \geq 40 \\ x_5 + x_3 \geq 60 \\ x_6 + x_4 \geq 35 \\ x_i \geq 0 \quad i = 1, \dots, 6 \\ x_i \text{ int} \quad i = 1, \dots, 6 \end{array}$$

**Esercizio n.2** (punti: 11)
Dato il seguente problema $(P)$ di Programmazione Lineare:

$$(P) \left\{ \begin{array}{lrrl} \min z = & 2x_1 & -3x_2 & \\ & -4x_1 & -3x_2 & \geq 10 \\ & -2x_1 & +4x_2 & \leq 6 \\ & & x_2 & \geq 0 \end{array} \right. ,$$

sia $\hat{x} = [B^{-1}b,\ 0]^T$ una soluzione di base ammissibile non ottima per $(P)$ in corrispondenza dell'insieme di indici di base $\beta = \{1^-, 4\}$, con

$$B^{-1} = \begin{bmatrix} 1/4 & 0 \\ -1/2 & 1 \end{bmatrix}$$

e $b$ il vettore dei termini noti del sistema di vincoli di $(P)$.

1. (punti: 6) Determinare una soluzione ottima per $(P)$ applicando il metodo del simplesso a partire da $\hat{x}$;
2. (punti: 2) costruire il duale $(D)$ di $(P)$;

*NOTA: La prova scritta si intende superata se si realizza un punteggio complessivo pari almeno a 17**

3. (punti: 3) applicando le relazioni di scarto complementare (complementary slackness), risolvere $(D)$ a partire dalla soluzione ottima di $(P)$.

*Risoluzione*

Per applicare il simplesso bisogna riscrivere il problema in forma standard e costruire la tabella $T$ in forma canonica, corrispondente a $\beta$. Ricordiamo che, poichè $x_1$ è libera in segno, essa viene sostituita nella forma standard dalle variabili positive $x_1^+$ e $x_1^-$, ponendo $x_1 = x_1^+ - x_1^-$. La prima tabella in forma canonica è la seguente:

$$\begin{array}{ccccc|c} -1 & 1 & -3/4 & -1/4 & 0 & 5/2 \\ 0 & 0 & 11/2 & 1/2 & 1 & 1 \\ \hline 0 & 0 & -9/2 & -1/2 & 0 & 5 \end{array}.$$

Facendo pivot sull'elemento $T_{23} = 11/2$, si ottiene

$$\begin{array}{ccccc|c} -1 & 1 & 0 & -2/11 & 3/22 & 29/11 \\ 0 & 0 & 1 & 1/11 & 2/11 & 2/11 \\ \hline 0 & 0 & 0 & -1/11 & 9/11 & 64/11 \end{array}.$$

Infine, facendo pivot sull'elemento $T_{24} = 1/11$, si ottiene la seguente tabella ottima:

$$\begin{array}{ccccc|c} -1 & 1 & 2 & 0 & 1/2 & 3 \\ 0 & 0 & 11 & 1 & 2 & 2 \\ \hline 0 & 0 & 1 & 0 & 1 & 6 \end{array},$$

corrispondente a $x^* = [0, 3, 0, 2, 0]^T$ e $z^* = -6$. Tenendo conto che $x_1 = x_1^+ - x_1^-$, la soluzione ottima del problema di partenza è $x^* = [-3, 0]^T$.

Per costruire il duale $(D)$ di $(P)$, moltiplichiamo prima per -1 il II vincolo di $(P)$. Quindi il duale è:

$$(D) \begin{cases} \max w = & 10\pi_1 & -6\pi_2 & \\ & -4\pi_1 & +2\pi_2 & = 2 \\ & -3\pi_1 & -4\pi_2 & \leq -3 \\ & \pi_1, & \pi_2 & \geq 0 \end{cases},$$

Imponendo le relazioni di scarto complementare, si ottiene $\pi^* = [0, 1]^T$.

**Esercizio n.3** (punti: 7)

Dato il seguente problema $(PLI)$ di Programmazione Lineare Intera

$$(PLI) \begin{cases} \max z = & -x_1 & +\frac{5}{2}x_2 & \\ & & 2x_2 & \leq 5 \\ & x_1 & & \leq 0 \\ & x_1 & & \geq -2 \\ & & x_2 & \geq 0 \\ & x_1, & x_2 & \text{int.} \end{cases},$$

1. (punti: 3) determinare la formulazione ideale (funzione obiettivo e vincoli) di $(PLI)$;
2. (punti: 4) risolvere $(PLI)$ applicando il metodo Branch & Bound e risolvendo tutti i rilassamenti continui per via grafica.

*Risoluzione*

La formulazione ideale di $(PLI)$ è la seguente:

$$(PLI_{id}) \begin{cases} \max z = & -x_1 & +\frac{5}{2}x_2 & \\ & & x_2 & \leq 2 \\ & x_1 & & \leq 0 \\ & x_1 & & \geq -2 \\ & & x_2 & \geq 0 \\ & x_1, & x_2 & \text{int.} \end{cases},$$

Risolvendo il problema $(PLI)$ con l'algoritmo di Branch & Bound e scegliendo una qualsiasi delle due strategie di visita dell'albero di Branch & Bound, vengono generati a partire da $(PLI)$ due soli sottoproblemi: uno viene chiuso per interezza della soluzione (si aggiornano $x_{opt}$ e $z_{opt}$) del rilassamento continuo, l'altro per inammissibilità del rilassamento continuo. La soluzione ottima di $(PLI)$ è $x^*_{PLI} = [-2, 2]^T$, con valore di funzione obiettivo pari a 7.

**Esercizio n.4** (punti: 6)
Una casa editrice deve effettuare il trasporto di libri da 3 depositi ($D_1$, $D_2$, $D_3$) a 4 librerie ($L_1$, $L_2$, $L_3$, $L_4$). Nella seguente tabella sono riportati i costi unitari (espressi in euro) $c_{ij}$ di trasporto da ciascun deposito $D_i$ a ciascuna libreria $L_i$, le quantità di libri $a_i$ $(i = 1, 2, 3)$ disponibili nei depositi e le quantità $b_j$ $(j = 1, 2, 3, 4)$ richieste dalle singole librerie:

| | $L_1$ | $L_2$ | $L_3$ | $L_4$ | $a_i$ |
|---|---|---|---|---|---|
| $D_1$ | 0.5 | 0.8 | 1 | 1.5 | 50 |
| $D_2$ | 0.7 | 2 | 0.8 | 0.5 | 100 |
| $D_3$ | 1 | 0.5 | 1.5 | 0.6 | 40 |
| $b_j$ | 30 | 70 | 45 | 45 | |

Ad esempio, il deposito $D_1$ ha una disponibilità di 50 libri e la libreria $L_3$ ne richiede almeno 45; inoltre trasportare un libro da $D_1$ a $L_4$ costa 1.5 euro.

Poichè i costi di trasporto sono a carico delle librerie, l'obiettivo è quello di minimizzare il massimo fra i costi di trasporto sostenuti da ciascuna libreria e nel contempo soddisfare i vincoli di domanda e di offerta.

Indicando con $x_{ij}$ (variabili decisionali) il numero di libri trasportati dal deposito $D_i$ alla libreria $L_j$ $(i = 1, 2, 3$ e $j = 1, 2, 3, 4)$, il problema può essere formulato come problema $(P)$ di ottimizzazione nel seguente modo:

$$(P) \begin{cases} \min\limits_{x} \quad \max\limits_{j=1,2,3,4} \displaystyle\sum_{i=1}^{3} c_{ij}x_{ij} \\ \displaystyle\sum_{j=1}^{4} x_{ij} \le a_i \quad i = 1, 2, 3 \\ \displaystyle\sum_{i=1}^{3} x_{ij} \ge b_j \quad j = 1, 2, 3, 4 \\ x_{ij} \ge 0 \quad i = 1, 2, 3 \quad j = 1, 2, 3, 4 \\ x_{ij} \text{ int} \quad i = 1, 2, 3 \quad j = 1, 2, 3, 4 \end{cases} .$$

- Trasformare il problema $(P)$ in un problema equivalente $(P_1)$, avente funzione obiettivo lineare;
- scrivere un programma LINGO (Linear, INteractive and General Optimizer) che consenta di risolvere $(P_1)$, facendo uso degli oggetti "SETS" e "DATA".

*Risoluzione*
Il problema $(P_1)$ (linearizzato di $(P)$), è il seguente:

$$(P_1) \begin{cases} \min\limits_{x,v} \quad v \\ \displaystyle\sum_{i=1}^{3} c_{ij}x_{ij} \le v \quad j = 1, 2, 3, 4 \\ \displaystyle\sum_{j=1}^{4} x_{ij} \le a_i \quad i = 1, 2, 3 \\ \displaystyle\sum_{i=1}^{3} x_{ij} \ge b_j \quad j = 1, 2, 3, 4 \\ x_{ij} \ge 0 \quad \imath = 1, 2, 3 \quad j = 1, 2, 3, 4 \\ x_{ij} \text{ int} \quad i = 1, 2, 3 \quad j = 1, 2, 3, 4 \end{cases} .$$

Facendo riferimento alla notazione usata in $(P_1)$, un possibile modo di scrivere il programma LINGO è il seguente:

```
SETS:
DEPOSITI / D1 D2 D3 / : A;
LIBRERIE / L1 L2 L3 L4/ : B;
LIBRITRASP (DEPOSITI, LIBRERIE): X, C;
ENDSETS

DATA:
C = 0.5 0.8 1 1.5 0.7 2 0.8 0.5 1 0.5 1.5 0.6;
A = 50 100 40;
B= 30 70 45 45;
ENDDATA

MIN = V;

@FOR(LIBRERIE(J): V >= @SUM(DEPOSITI(I): C(I,J)*X(I,J)));
@FOR(DEPOSITI(I): @SUM(LIBRERIE(J): X(I,J)) <= A (I));
@FOR(LIBRERIE(J): @SUM(DEPOSITI(I): X(I,J)) >= B (J));
@FOR(DEPOSITI(I): @FOR(LIBRERIE(J): @GIN(X(I,J))));
```