Prova Scritta di RICERCA OPERATIVA

13 Gen. 2003

Nome e Cognome:

Esercizio 1. (6 punti) Una azienda agricola coltiva mais e alleva vitelli, usando tre diversi procedimenti. Con il primo procedimento vengono allevati 1000 vitelli utilizzando 30 mesi/uomo di lavoro, 20 ettari di terreno, 200 tonnellate di mais. Con il secondo procedimento vengono prodotte 100 tonnellate di mais utilizzando 20 mesi/uomo di lavoro e 40 ettari di terreno. Con il terzo procedimento vengono prodotti 200 tonnellate di mais e 500 vitelli utilizzando 40 mesi/uomo di lavoro e 50 ettari di terreno.
Il mais si vende a 500 Euro per tonnellata, i vitelli a 200 Euro ciascuno.
L'azienda possiede 70 ettari di terreno, dispone di 50 mesi/uomo di lavoro e deve essere autosufficiente per quanto riguarda il mais. Lo scopo dell'azienda è massimizzare il profitto.

Domanda a) (4 punti) Scrivere il modello lineare per trovare quanto produrre secondo ogni procedimento, tenendo conto che ogni procedimento può essere applicato più volte e anche parzialmente (es. il primo procedimento applicato 0,5 volte vuol dire produrre 500 vitelli utilizzando 15 mesi/uomo, 10 ettari e 100 tonnellate di mais.) Poiché il numero di vitelli è elevato, e si parla di valori medi, non occorre preoccuparsi del fatto che tale numero risulti intero.
Domanda b) (2 punti) Si supponga che non si possa usare contemporaneamente il secondo e il terzo procedimento. Modificare la formulazione ottenuta al punto (a) in modo da tener conto di questo ulteriore vincolo.

Soluzione a) Conviene fare uno schema delle 3 possibilità:

	Produce		Utilizza		
1	1000 vitelli		30 mesi/uomo	20 ettari	200 t. mais
2	100 t. mais		20 mesi/uomo	40 ettari	
3	200 t. mais	500 vitelli	40 mesi/uomo	50 ettari	

Si introducono le seguenti variabili:

x_i = quantità di procedimento i adottata.

Come detto, ogni procedimento può essere applicato più volte ed anche parzialmente. Le variabili sono perciò reali (ma non negative).
Il profitto dell'azienda è dato dalla quantità di vitelli e di mais venduti, ovvero quelle prodotte meno quelle utilizzate. Il profitto è quindi

$$200\,(1000\,x_1 + 500\,x_3) + 500\,(100\,x_2 + 200\,x_3 - 200\,x_1)$$

Due vincoli sono quelli sulla quantità di ettari di terreno e di mesi/uomo di lavoro disponibili. Da notare che risparmiare su queste grandezze non fornisce alcun guadagno. Un terzo vincolo è dato dal fatto che la quantità di mais utilizzata non può superare quella prodotta (autosufficienza). Il modello lineare è quindi:

$$\max 200\,(1000\,x_1 + 500\,x_3) + 500\,(100\,x_2 + 200\,x_3 - 200\,x_1)$$

$$\begin{aligned}
&20\,x_1 + 40\,x_2 + 50\,x_3 \le 70\\
&30\,x_1 + 20\,x_2 + 40\,x_3 \le 50\\
&-200\,x_1 + 100\,x_2 + 200\,x_3 \ge 0\\
\\
&x_1, x_2, x_3 \ge 0\\
&x_1, x_2, x_3 \in R
\end{aligned}$$

Soluzione b) Dato che le variabili del problema sono reali, per esprimere la mutua esclusività occorre introdurre una variabile binaria

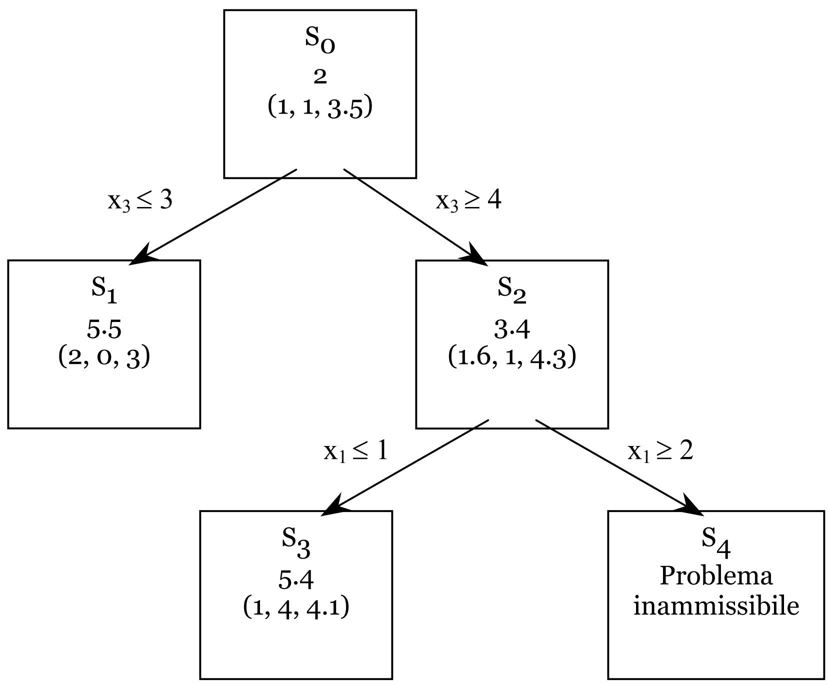
$$y = \begin{cases} 1 & \text{se uso (almeno in parte) il procedimento 2 (e quindi non posso usare il procedimento 3)} \\ 0 & \text{altrimenti (non uso il procedimento 2 e quindi posso usare il procedimento 3)} \end{cases}$$

i vincoli da aggiungere al modello sono allora, con M costante più grande di ogni possibile valore di x_2 e x_3 :

$$x_2 \le My \qquad x_3 \le M(1\text{-}y)$$

Esercizio 2. (5 punti) Sia dato un problema di Programmazione Lineare Intera nelle variabili $x_1, x_2, x_3 \in Z_+$.

Per trovare la soluzione di tale problema viene applicato il metodo del "Branch and Bound". Alcune iterazioni di tale metodo sono riassunte nel seguente albero di enumerazione. In tale albero, per ogni sottoproblema S_i è indicato la soluzione ottima del rilassamento lineare.



Dire, motivando la risposta, se il metodo del "Branch and Bound" ha già determinato la soluzione ottima arrestandosi. In caso affermativo indicare la soluzione ottima del problema di partenza. In caso negativo dire su quali problemi e su quali variabili si può fare il branching. Commentare l'esplorazione dell'albero.

Soluzione Dato che aggiungendo vincoli notiamo un incremento del valore della funzione obiettivo, deve essere un problema di minimizzazione. La soluzione del primo rilassamento non è intera e viene fatto il branching sulla veriabile frazionaria x_3 . Il sottoproblema S_1 ha soluzione intera e diventa l'ottimo corrente. Il sottoproblema S_2 ha soluzione frazionaria e valore della funzione obiettivo migliore (inferiore) dell'ottimo corrente, quindi viene fatto il branching sulla variabile frazionaria x_1 . Il sottoproblema S_3 ha soluzione frazionaria e valore della funzione obiettivo migliore (inferiore) dell'ottimo corrente, quindi bisognerebbe proseguire il branching sulla variabile frazionaria x_3 . Il sottoproblema S_4 è invece inammissibile, e può quindi essere chiuso. Dato che occorre proseguire con il branching, non è possibile dire se l'ottimo è stato già raggiunto o meno, e quindi la soluzione ottima non è stata determinata.

Esercizio 3. (4 punti) Fornire un semplice esempio di modellazione di un problema di costo fisso, descrivendo un problema ed il corrispondente modello matematico.

Soluzione Un esempio consiste nello spedire del materiale con un autocarro, dove si paga un costo f per il fatto di usare l'autocarro (costo fisso) e un costo c per ogni unità di merce trasportata.
Spedire un quantità x di merce costa quindi:

$$\text{costo} = \begin{cases} f + cx & \text{se } x > 0 \\ 0 & \text{se } x = 0 \end{cases}$$

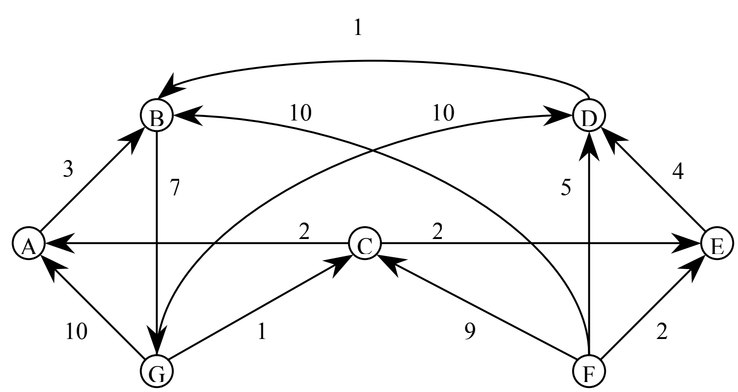
Per rappresentare questa discontinuità con un modello lineare occorre introdurre una variabile binaria

$$y = \begin{cases} 1 & \text{se } x > 0 \\ 0 & \text{se } x = 0 \end{cases}$$

Per rendere l'esempio molto semplice (anche se non troppo realistico), immaginiamo di dover minimizzare il costo di spedizione, con dei vincoli di tipo $ax \geq b$. Indicando con M una costante più grande di ogni possibile valore di x, il modello che rappresenta questo problema è allora:

$$\begin{aligned} &\min fy + cx \\ &\quad ax \geq b \\ &\quad My \geq x \\ &\quad x \geq 0 \\ &\quad x \in R \\ &\quad y \in \{0,1\} \end{aligned}$$

Esercizio 4. (4 punti) Sia dato il seguente grafo, dove i pesi sugli archi rappresentano distanze.

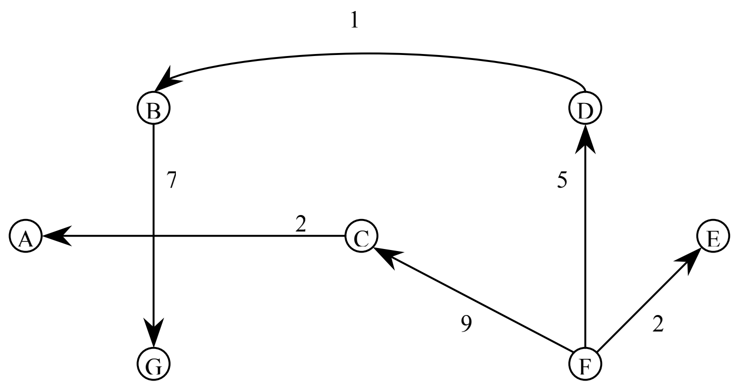


Calcolare, usando l'opportuno algoritmo, l'albero dei cammini minimi dal nodo F a tutti gli altri.

Soluzione Intanto il problema ammette soluzione, dato che tutti i nodi sono raggiungibili dal nodo F (altrimenti sarebbe stato un problema inammissibile), e non contiene cicli di lunghezza negativa (altrimenti il problema sarebbe stato illimitato inferiormente). Occorre allora scegliere l'opportuno algoritmo.
Dato che il grafo contiene cicli, non è possibile utilizzare l'algoritmo per grafi aciclici. Essendo tutti positivi i pesi degli archi, si può utilizzare l'algoritmo di Dijkstra, rappresentato nella seguente tabella.

A		B		C		D		E		G	
dist.	predec.	dist.	predec.	dist.	predec.	dist.	predec.	dist.	predec.	dist.	predec.
∞	-	10	F	9	F	5	F	2	F	∞	-
∞	-	10	F	9	F	5	F			∞	-
∞	-	6	D	9	F					∞	-
∞	-			9	F					13	B
11	C									13	B
										13	B

L'albero dei cammini minimi cercato è allora



Esercizio 5. (6 punti) Dato un grafo connesso non orientato G = (V,E), un suo albero ricoprente, come noto, è un albero T = (W,F) avente W=V e $F \subseteq E$. Se ogni arco e_i ha un peso p_i, il peso dell'albero ricoprente è la somma dei pesi dei suoi archi.

Domanda a) (2 punti) Fornire la definizione di albero.
Domanda b) (4 punti) Definire un algoritmo di tipo Greedy per trovare, dato un grafo, l'albero ricoprente di peso minimo (suggerimento: riflettere sulla presenza o assenza di cicli nelle soluzioni parziali).

Soluzione a) Un albero è un grafo aciclico connesso.
Soluzione b) Un algoritmo Greedy costruisce la soluzione passo per passo, partendo dalla soluzione vuota e generando soluzioni parziali fino al raggiungimento di un soluzione complessiva. È una euristica, nel senso che non garantisce di trovare la soluzione ottima, ma dovrebbe trovare una soluzione abbastanza buona in tempi che dovrebbero essere più rapidi di quelli di un algoritmo esatto. Può risolvere problemi di ottimizzazione combinatoria, ovvero problemi in cui si debba scegliere, tra tanti possibili sottoinsiemi di un insieme dati, il migliore secondo un certo criterio.
Dato che l'albero ricoprente di un grafo non orientato è ottenuto prendendo un sottoinsieme degli archi del grafo di partenza, appare evidente che si debba ad ogni passo scegliere un arco. Dato che vogliamo l'albero ricoprente di peso minimo, un algoritmo Greedy sceglie ad ogni passo l'arco di peso inferiore tra quelli disponibili. Rimane a questo punto da definire quali sono, ad ogni passo, gli archi disponibili.
Intanto gli archi già scelti non saranno più disponibili. Poi, una soluzione parziale deve in questo caso essere un insieme di archi che possa essere esteso fino a diventare un albero ricoprente. Dato che un albero è aciclico, tutte le soluzioni parziali devono essere prive di cicli. In definitiva, ad ogni passo, l'insieme degli archi disponibili è dato da quelli non ancora scelti e tali da non creare cicli nella soluzione parziale corrente. L'algoritmo è quindi:

inizializzazione: l'insieme degli archi scelti F_0 è vuoto
iterazione i-esima: scegli l'arco $e_j \in E \backslash F_i$ tale che $e_j \cup F_i$ non contenga cicli ed e_j abbia peso minimo
terminazione: se T_i è albero ricoprente (raggiunge tutti i nodi) STOP, altrimenti poni i=i+1 e ripeti **iterazione**

Esercizio 6. (7 punti) Il seguente modello matematico rappresenta il problema di un progettista che deve progettare i collegamenti tra n oggetti (le variabili binarie x_{ij} valgono 1 quando l'oggetto i viene collegato all'oggetto j, si noti che il collegamento da i a j è diverso dal collegamento da j a i) rispettando alcuni vincoli.

$$\max \sum_i \sum_j x_{ij}$$

$$\sum_i x_{ij} \le b_1 \quad \forall j \quad \text{(numero di collegamenti che partono da ogni oggetto)}$$
$$\sum_j x_{ij} \le b_2 \quad \forall i \quad \text{(numero di collegamenti che arrivano ad ogni oggetto)}$$
$$\sum_i \sum_j c_{ij} x_{ij} \le b_3 \quad \text{(costo complessivo)}$$

$$x_{ij} \in \{0,1\}$$

$$\text{con } c = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \\ 3 & 1 & 4 & 2 \\ 4 & 3 & 2 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 4 \\ 3 \\ 10 \end{pmatrix}$$

Domanda a) (3 punti) Scrivere il file del modello **.mod** e dei dati **.dat** che nella sintassi di AMPL rappresentino correttamente il problema.
Domanda b) (2 punti) Modificare quanto al punto **a)** nelle ipotesi che i collegamenti tra un oggetto e se stesso non siano possibili.
Domanda c) (2 punti) Modificare quanto ai punti **a)** e **b)** nelle ipotesi che, per ogni coppia i e j, si possa attuare uno solo tra i collegamenti da i a j e da j a i.

Soluzione a) Per avere un modello valido per n oggetti bisogna dichiarare un parametro n e usarlo per definire l'insieme degli oggetti. Il resto del modello è abbastanza standard. I file di modello e di dati sono allora:

oggetti.mod

```
param n;           # numero di oggetti
set O := 1..n;     # oggetti

param c{O,O} ;     # costi
param b{1..3};     # termini noti

var x{O,O} >= 0, binary; # collegamenti attivabili

maximize obj: sum{i in O,j in O} x[i,j];

s.t. da{j in O}: sum{i in O} x[i,j] <= b[1];
s.t. ad{i in O}: sum{j in O} x[i,j] <= b[2];
s.t. cost: sum{i in O,j in O} c[i,j]*x[i,j] <= b[3];
```

oggetti.dat

```
param n := 4;

param b := 1 4
           2 3
           3 10;

param c:  1  2  3  4 :=
       1  1  2  3  4
       2  2  4  1  3
       3  3  1  4  2
       4  4  3  2  1;
```

Soluzione b) Per escludere i collegamenti tra un oggetto e se stesso conviene non creare proprio le variabili. Occorre poi ricordarsi che le variabili con i due indici uguali non esistono in tutto il resto del modello. Il file di modello diventa quindi:

```
param n;          # numero di oggetti
set O := 1..n;    # oggetti

param c{O,O} ;    # costi
param b{1..3};    # termini noti

var x{i in O, j in O: i<>j} >= 0, binary; # collegamenti attivabili

maximize obj: sum{i in O,j in O: i<>j} x[i,j];

s.t. da{j in O}: sum{i in O: i<>j} x[i,j] <= b[1];
s.t. ad{i in O}: sum{j in O: i<>j} x[i,j] <= b[2];
s.t. cost: sum{i in O,j in O: i<>j} c[i,j]*x[i,j] <= b[3];
```

Soluzione c) Per escludere che possano essere attivati entrambe i collegamenti da `i` a `j` e da `j` a `i` occorre introdurre altri vincoli del tipo $x_{ij} + x_{ji} \leq 1$ per tutte le opportune coppie (i,j). Il file di modello diventa quindi:

```
param n;          # numero di oggetti
set O := 1..n;    # oggetti

param c{O,O} ;    # costi
param b{1..3};    # termini noti

var x{i in O, j in O: i<>j} >= 0, binary; # collegamenti attivabili

maximize obj: sum{i in O,j in O: i<>j} x[i,j];

s.t. da{j in O}: sum{i in O: i<>j} x[i,j] <= b[1];
s.t. ad{i in O}: sum{j in O: i<>j} x[i,j] <= b[2];
s.t. cost: sum{i in O,j in O: i<>j} c[i,j]*x[i,j] <= b[3];
s.t. esclusione{i in O, j in O: i<j}: x[i,j] + x[j,i] <= 1;
```