

POLITECNICO DI MILANO



# Esercizi e Laboratori di Ricerca Operativa

*Per i corsi di Ricerca Operativa*

Leo Liberti

Ultimo aggiornamento: 9 novembre 2005



# Indice

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Esercizi teorici</b>                          | <b>5</b> |
| 1.1      | Algoritmo di Dijkstra . . . . .                  | 5        |
| 1.2      | Algoritmo di Bellman-Ford . . . . .              | 5        |
| 1.3      | Taglio minimo in un grafo . . . . .              | 6        |
| 1.4      | Rinnovo dei macchinari . . . . .                 | 6        |
| 1.5      | Progetto con precedenza sulle attività . . . . . | 6        |
| 1.6      | Dualità . . . . .                                | 6        |
| 1.7      | Metodo del simplesso . . . . .                   | 7        |
| 1.8      | Scarti complementari . . . . .                   | 7        |
| 1.9      | Analisi di sensitività . . . . .                 | 8        |
| 1.10     | Obiettivo lineare a tratti . . . . .             | 8        |
| 1.11     | Tagli di Gomory . . . . .                        | 8        |
| 1.12     | Branch and Bound . . . . .                       | 8        |
| <b>2</b> | <b>Problemi facili</b>                           | <b>9</b> |
| 2.1      | Mix produttivo . . . . .                         | 9        |
| 2.2      | Multiperiodo . . . . .                           | 9        |
| 2.3      | Trasporto . . . . .                              | 10       |
| 2.4      | Guardiani di un museo . . . . .                  | 11       |
| 2.5      | Divisione di eredità . . . . .                   | 12       |
| 2.6      | Islandia . . . . .                               | 12       |
| 2.7      | Scheduling eterogeneo . . . . .                  | 12       |
| 2.8      | Scheduling sequenziale . . . . .                 | 13       |
| 2.9      | Parcheggio . . . . .                             | 13       |

---

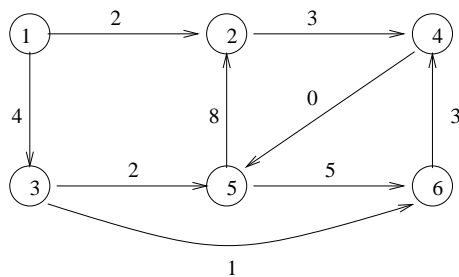
|          |  |           |
|----------|--|-----------|
| <b>3</b> | <b>Problemi difficili</b>                          | <b>15</b> |
| 3.1      | Checksum sui dati . . . . .                        | 15        |
| 3.2      | Problema delle otto regine . . . . .               | 17        |
| 3.3      | Gestione di un'impresa . . . . .                   | 17        |
| 3.4      | Pianificazione della produzione alla DEC . . . . . | 17        |
| 3.5      | Problema del commesso viaggiatore . . . . .        | 18        |
| 3.6      | Controllo ottimo di un razzo . . . . .             | 19        |
| 3.7      | Doppio monopolio . . . . .                         | 19        |
| <b>4</b> | <b>Applicazioni in ambito telecomunicazioni</b>    | <b>21</b> |
| 4.1      | Instradamento di pacchetti . . . . .               | 21        |
| 4.2      | Affidabilità di una rete . . . . .                 | 22        |
| 4.3      | Network Design . . . . .                           | 22        |
| 4.4      | Network Routing . . . . .                          | 22        |
| <b>5</b> | <b>Problemi non lineari</b>                        | <b>25</b> |
| 5.1      | Codici correttori di errore . . . . .              | 25        |
| 5.2      | Manutenzione aerea . . . . .                       | 26        |
| 5.3      | Miscelazione di greggio . . . . .                  | 26        |
| 5.4      | Controllo ottimale di un razzo . . . . .           | 27        |
| 5.5      | Lotti ottimi . . . . .                             | 27        |

# Capitolo 1

## Esercizi teorici

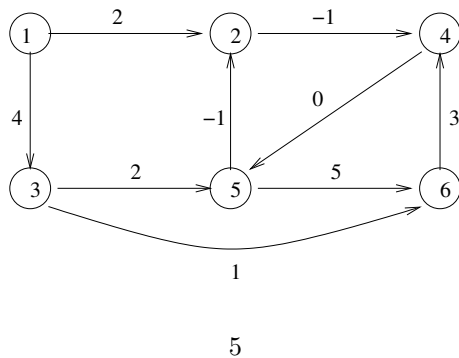
### 1.1 Algoritmo di Dijkstra

Costruire l'albero dei cammini minimi del grafo sotto a partire dal nodo 1, utilizzando l'algoritmo di Dijkstra.



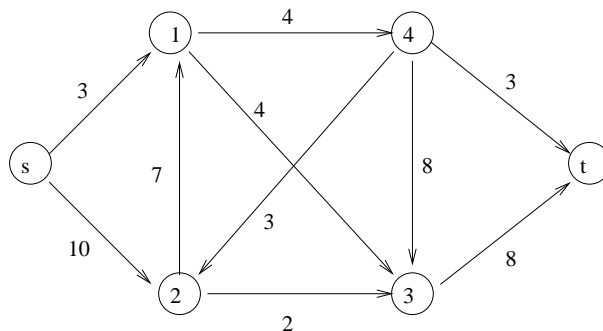
### 1.2 Algoritmo di Bellman-Ford

Verificare la presenza di cicli negativi nel grafo sotto, utilizzando l'algoritmo di Bellman-Ford a partire dal nodo 1.



### 1.3 Taglio minimo in un grafo

Trovare il taglio minimo nel grafo sotto e specificare quale algoritmo si è usato (i numeri sugli archi indicano le capacità).



### 1.4 Rinnovo dei macchinari

Un'impresa acquista un nuovo macchinario per 12000 euro, di cui si conosce il costo annuale di manutenzione per i prossimi 5 anni.

| età (anni) | manutenzione (keuro) | ricavato (keuro) |
|------------|----------------------|------------------|
| 0          | 2                    | -                |
| 1          | 4                    | 7                |
| 2          | 5                    | 6                |
| 3          | 9                    | 2                |
| 4          | 12                   | 1                |

Per evitare i costi elevati di un macchinario vecchio, si può ridarlo indietro all'inizio del secondo, terzo, quarto o quinto anno, e comprarne uno nuovo. Determinare un piano di rinnovo del macchinario che minimizzi il costo totale netto (acquisti + manutenzione + ricavato) per un periodo di 5 anni. [Amaldi]

### 1.5 Progetto con precedenze sulle attività

Un progetto è decomponibile nelle seguenti 7 attività la cui durata in giorni è indicata tra parentesi:  $A$  (4),  $B$  (3),  $C$  (5),  $D$  (2),  $E$  (10),  $F$  (10),  $G$  (1). Nello svolgere le attività devono venire rispettate le seguenti precedenze:  $A \rightarrow G, D$ ;  $E, G \rightarrow F$ ;  $D, F \rightarrow C$ ;  $F \rightarrow B$ . Ogni giorno di lavoro comporta un costo di 1000 euro, inoltre dall'inizio dell'attività  $A$  alla fine dell'attività  $B$  deve venire affittata una apparecchiatura il cui costo è di 5000 euro al giorno. Formulare il problema in termini di programmazione lineare (individuare variabili decisionali, vincoli e funzione obiettivo). Domanda opzionale: descrivere brevemente un algoritmo per risolvere il problema. [Malucelli]

### 1.6 Dualità

Si scriva il duale dei seguenti problemi:

1.

$$\min_x \left. \begin{array}{l} 3x_1 + 5x_2 - x_3 \\ x_1 - x_2 + x_3 \leq 3 \\ 2x_1 - 3x_2 \leq 4 \\ x \geq 0 \end{array} \right\} \quad (1.1)$$

2.

$$\min_x \left. \begin{array}{l} x_1 - x_2 - x_3 \\ -3x_1 - x_2 + x_3 \leq 3 \\ 2x_1 - 3x_2 - 2x_3 \geq 4 \\ x_1 - x_3 = 2 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{array} \right\} \quad (1.2)$$

3.

$$\max_x \left. \begin{array}{l} x_1 - x_2 - 2x_3 + 3 \\ -3x_1 - x_2 + x_3 \leq 3 \\ 2x_1 - 3x_2 \geq 4x_3 \\ x_1 - x_3 = x_2 \\ x_1 \geq 0 \\ x_2 \leq 0 \end{array} \right\} \quad (1.3)$$

## 1.7 Metodo del simplesso

Si risolva il seguente problema:

$$\max_x \left. \begin{array}{l} x_1 + x_2 \\ -x_1 + x_2 \leq 1 \\ 2x_1 + x_2 \leq 4 \\ x_1 \geq 0 \\ x_2 \leq 0 \end{array} \right\}$$

mediante il metodo del simplesso, a partire dal punto ammissibile  $\bar{x} = (1, 0)$ . [*Malucelli*]

## 1.8 Scarti complementari

Si consideri

$$\begin{array}{rcl} \max & 2x_1 & + \quad x_2 \\ & x_1 & + \quad 2x_2 \leq 14 \\ & 2x_1 & - \quad x_2 \leq 10 \\ & x_1 & - \quad x_2 \leq 3 \\ & x_1 & , \quad x_2 \geq 0 \end{array}$$

1. Scrivere il problema duale;
2. Verificare che  $\bar{x} = (\frac{20}{3}, \frac{11}{3})$  è soluzione ammissibile;
3. Dimostrare che  $\bar{x}$  è anche una soluzione ottima per mezzo del teorema degli scarti complementari, e determinare la soluzione ottima del duale. [*Belotti*]

## 1.9 Analisi di sensitività

Si consideri il problema:

$$\min \left. \begin{array}{l} x_1 - 5x_2 \\ -x_1 + x_2 \leq 5 \\ x_1 + 4x_2 \leq 40 \\ 2x_1 + x_2 \leq 20 \\ x_1, x_2 \geq 0. \end{array} \right\}$$

1. Si verifichi che la soluzione ammissibile  $x^* = (4, 9)$  è anche ottima.
2. Interpretando la funzione obiettivo come funzione di costo di un problema decisionale, e i termini noti dei vincoli come limiti sulla produzione / vendita di un bene, quale dei tre limiti sarebbe opportuno incrementare per migliorare ulteriormente il valore della funzione obiettivo (supponendo che la variazione del limite non cambi la base ottima)?

## 1.10 Obiettivo lineare a tratti

Si riformuli il problema  $\min\{f(x) \mid x \in \mathbb{R}_{\geq 0}\}$ , dove

$$f(x) = \begin{cases} -x + 1 & 0 \leq x < 1 \\ x - 1 & 1 \leq x < 2 \\ \frac{1}{2}x & 2 \leq x \leq 3 \end{cases}$$

come problema di programmazione lineare a variabili miste intere e continue. [Malucelli]

## 1.11 Tagli di Gomory

Si risolva il problema seguente:

$$\min \left. \begin{array}{l} x_1 - 2x_2 \\ -4x_1 + 6x_2 \leq 9 \\ x_1 + x_2 \leq 4 \\ x \geq 0, \quad x \in \mathbb{Z}^2 \end{array} \right\}$$

mediante l'algoritmo dei piani di taglio di Gomory. [Problema tratto da Bertsimas e Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, Belmont, 1997.]

## 1.12 Branch and Bound

Si risolva il problema seguente:

$$\max \left. \begin{array}{l} 2x_1 + 3x_2 \\ x_1 + 2x_2 \leq 3 \\ 6x_1 + 8x_2 \leq 15 \\ x_1, x_2 \in \mathbb{Z}_+ \end{array} \right\}$$

mediante l'algoritmo Branch and Bound; risolvere ogni sottoproblema per via grafica.



## Capitolo 2

# Problemi facili

### 2.1 Mix produttivo

Un'azienda vuole pianificare il livello di produzione di 3 prodotti ( $A_1, A_2, A_3$ ). Vi sono in totale 22 giorni di produzione disponibili in un mese. Nella seguente tabella sono indicati: la domanda massima (in quintali), il prezzo di vendita (in dollari/quintale), il costo di produzione per quintale di prodotto e la quota di produzione (cioè il massimo numero di quintali prodotti in 1 giorno se tutte le risorse a disposizione fossero usate per un solo tipo di prodotto).

| Prodotto            | $A_1$   | $A_2$   | $A_3$   |
|---------------------|---------|---------|---------|
| Domanda massima     | 5300    | 4500    | 5400    |
| Prezzo di vendita   | \$124   | \$109   | \$115   |
| Costo di produzione | \$73.30 | \$52.90 | \$65.40 |
| Quota di produzione | 500     | 450     | 550     |

1. Formulare un modello AMPL per determinare il piano di produzione che massimizza il guadagno totale. Come cambiano il modello matematico, il modello e i dati AMPL e la soluzione, se per ogni prodotto introduciamo un costo fisso di attivazione della linea di produzione:

| Prodotto             | $A_1$    | $A_2$    | $A_3$    |
|----------------------|----------|----------|----------|
| Costo di attivazione | \$170000 | \$150000 | \$100000 |

2. Come cambiano il modello matematico, il modello e i dati AMPL e la soluzione, se oltre al costo fisso di attivazione aggiungiamo anche il vincolo che se una linea di produzione viene attivata, è necessaria la produzione di un lotto minimo di prodotto:

| Prodotto     | $A_1$ | $A_2$ | $A_3$ |
|--------------|-------|-------|-------|
| Lotto minimo | 20    | 20    | 16    |

[Amaldi]

### 2.2 Multiperiodo

Si vuole pianificare la produzione di tre prodotti  $A_1, A_2, A_3$ , ma ammettendo che il livello di produzione cambi di mese in mese, con un orizzonte temporale di quattro mesi, da Gennaio ad Aprile.

La domanda cambia non solo da un prodotto a un altro, ma anche da un mese all'altro:

| Domanda di produzione | Gennaio | Febbraio | Marzo | Aprile |
|-----------------------|---------|----------|-------|--------|
| $A_1$                 | 5300    | 1200     | 7400  | 5300   |
| $A_2$                 | 4500    | 5400     | 6500  | 7200   |
| $A_3$                 | 4400    | 6700     | 12500 | 13200  |

I prezzi di vendita, costi di produzione, quote di produzione, costi di attivazione e lotti minimi sono:

| Produzione           | $A_1$    | $A_2$    | $A_3$    |
|----------------------|----------|----------|----------|
| Prezzo di vendita    | \$124    | \$109    | \$115    |
| Costo di attivazione | \$150000 | \$150000 | \$100000 |
| Costo di produzione  | \$73.30  | \$52.90  | \$65.40  |
| Quota di produzione  | 500      | 450      | 550      |
| Lotto minimo         | 20       | 20       | 16       |

I giorni di produzione disponibili variano ogni mese: 23 giorni a Gennaio, 20 a Febbraio, 23 a Marzo e 22 ad Aprile. La validità della decisione sull'attivazione o meno di una linea di produzione dura un mese. I lotti minimi sono da intendersi mensili.

Inoltre abbiamo la disponibilità di un magazzino con costo unitario mensile di \$3.50 per  $A_1$ , \$4.00 per  $A_2$  e \$3.00 per  $A_3$ . Ogni prodotto occupa lo stesso volume e il volume totale disponibile nel magazzino è di 800 unità.

Formulare un modello per determinare un piano di produzione che massimizza il guadagno totale. [Amaldi]

## 2.3 Trasporto

Una ditta di trasporto deve trasferire container vuoti dai propri 6 magazzini, situati a Verona, Perugia, Roma, Pescara, Taranto e Lamezia, ai principali porti nazionali (Genova, Venezia, Ancona, Napoli, Bari).

Le disponibilità di container vuoti ai Magazzini sono le seguenti

|         | Container vuoti |
|---------|-----------------|
| Verona  | 10              |
| Perugia | 12              |
| Roma    | 20              |
| Pescara | 24              |
| Taranto | 18              |
| Lamezia | 40              |

e le richieste ai Porti sono le seguenti:

|         | Container richiesti |
|---------|---------------------|
| Genova  | 20                  |
| Venezia | 15                  |
| Ancona  | 25                  |
| Napoli  | 33                  |
| Bari    | 21                  |

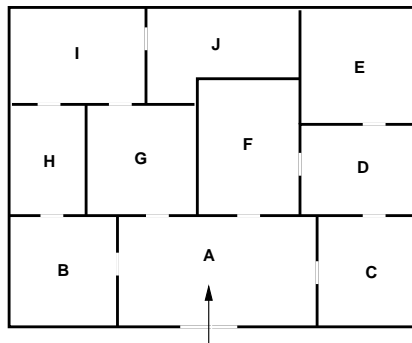
Trasportare i container dai magazzini ai porti costa. Il costo di trasporto di ciascun container è proporzionale alla distanza percorsa dal camion che lo trasporta. Il costo di trasporto chilometrico per ogni container è di 300 lire / km. Si sa inoltre che ogni camion può trasportare al massimo 2 container. Le distanze fra magazzini e porti sono riportate nella seguente tabella:

|         | Genova  | Venezia | Ancona | Napoli | Bari   |
|---------|---------|---------|--------|--------|--------|
| Verona  | 290 km  | 115 km  | 355 km | 715 km | 810 km |
| Perugia | 380 km  | 340 km  | 165 km | 380 km | 610 km |
| Roma    | 505 km  | 530 km  | 285 km | 220 km | 450 km |
| Pescara | 655 km  | 450 km  | 155 km | 240 km | 315 km |
| Taranto | 1010 km | 840 km  | 550 km | 305 km | 95 km  |
| Lamezia | 1072 km | 1097 km | 747 km | 372 km | 333 km |

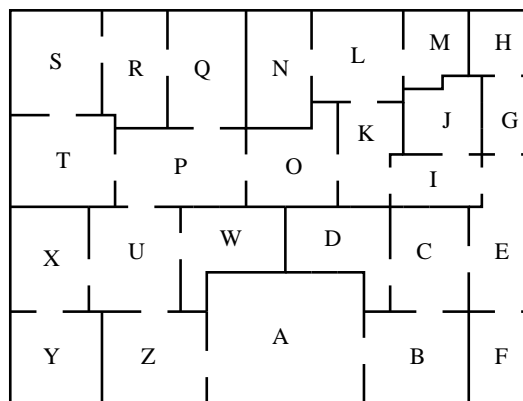
Si vuole determinare la politica di trasporto di costo complessivo minimo. [Amaldi]

## 2.4 Guardiani di un museo

Il direttore di un museo, prima dell'apertura di una nuova ala, deve decidere quante guardie assumere per vigilare sulle sale. Per motivi di budget decide che ogni guardiano stazionerà sulla porta di ogni sala, in modo che ognuno di loro controlli due sale. Sapendo che la mappa del museo è come in figura, formulare un modello di PLI per calcolare il numero di guardiani da assumere e risolverlo con AMPL.



Risolvere il problema anche sull'istanza sotto.



[Belotti]

## 2.5 Divisione di eredità

L'eredità di un ricco aristocratico, da poco passato a miglior vita, è costituita da:

- Una tela di Caillebotte del valore di 25.000
- Un busto di Diocleziano, valore 5.000
- Un vaso della dinastia Yuan valutato intorno ai 20.000
- Una Porsche 911 del valore di 40.000
- Tre diamanti del valore di 12.000 ciascuno;
- Un divano Luigi XV, valore 3.000
- Due preziosissimi cani di razza Jack Russell, da 3.000 ciascuno, che per volontà del defunto non possono essere divisi;
- Un bassorilievo del 200 d.C. valutato intorno ai 10.000
- Una barca a vela del valore di 15.000
- Una Harley Davidson da 10.000
- Un mobile appartenuto a Cavour, stimato sui 13.000

Deve essere diviso tra i due figli. Qual'è la divisione più equa, che minimizza la differenza tra i valori totali delle due parti? Proporre un modello di PLI e risolverlo con AMPL. [*Belotti*]

## 2.6 Islandia

Lo stato di Islandia ha quattro industrie esportatrici: acciaio, motori, elettronica e plastica. Il ministro dell'economia di questo stato vuole massimizzare il saldo esportazioni-importazioni. La moneta di Islandia è il klunz. I prezzi in klunz sul mercato mondiale per unità di acciaio, motori, elettronica e plastica sono rispettivamente 500, 1500, 300 e 1200. La produzione di una unità di acciaio richiede 0.02 unità di motori, 0.01 unità di plastica, 250 klunz di materie prime acquistate sul mercato mondiale e mezzo anno-uomo di manodopera. La produzione di una unità di motori richiede 0.8 unità di acciaio, 0.15 unità di elettronica, 0.11 unità di plastica, 300 klunz di materie prime acquistate sul mercato mondiale e un anno-uomo di manodopera. La produzione di una unità di prodotti elettronici richiede 0.01 unità di acciaio, 0.01 unità di motori, 0.05 unità di plastica, 50 klunz di materie prime acquistate sul mercato mondiale e mezzo anno-uomo di manodopera. La produzione di una unità di plastica richiede 0.03 unità di motori, 0.2 unità di acciaio, 0.05 unità di elettronica, 300 klunz di materie prime acquistate sul mercato mondiale e due anni-uomo di manodopera. La produzione di motori è limitata a 650000 unità, quella di plastica a 60000 unità. La manodopera totale disponibile in Islandia è di 830000 uomini per anno. Acciaio, motori, elettronica e plastica non possono essere importati, ma devono essere prodotti all'interno. [*Carello*]

## 2.7 Scheduling eterogeneo

È necessario 10 processi su 4 CPU da rispettivamente 1.33, 2, 2.66 e 4 GHz. Il numero di operazioni elementari dei processi (espresso in BI, che indica un miliardo di operazioni elementari) è come segue:

| processo | 1   | 2   | 3 | 4 | 5   | 6 | 7 | 8   | 9   | 10  |
|----------|-----|-----|---|---|-----|---|---|-----|-----|-----|
| BI       | 1.1 | 2.1 | 3 | 1 | 0.7 | 5 | 3 | 1.4 | 3.2 | 0.9 |

Si tratta di allocare i processi alle CPU in modo da minimizzare il tempo totale di completamento dei processi.

## 2.8 Scheduling sequenziale

Un impianto di colorazione dei tessuti consiste di 3 vasche di tinteggiatura. Ogni lotto di stoffa deve essere passato per ognuna delle vasche in un ordine predeterminato (si assume 1,2,3,4,5). L'impianto deve colorare 5 lotti di tessuto di diverse misure. Il tempo di colorazione del lotto  $i$  nella vasca  $j$  è dato dalla matrice  $(s_{ij})$  sotto (tempi in ore):

$$\begin{pmatrix} 3 & 1 & 1 \\ 2 & 1.5 & 1 \\ 3 & 1.2 & 1.3 \\ 2 & 2 & 2 \\ 2.1 & 2 & 3 \\ 1 & 1 & 4 \end{pmatrix}.$$

Si stabilisca il sequenziamento delle tinteggiature nelle diverse vasche con l'obiettivo di minimizzare il tempo di fine dell'ultima tinteggiatura.

## 2.9 Parcheggio

Su Dantzig Street si può parcheggiare da entrambi i lati della strada, e non c'è alcun divieto di sosta. Mr. Edmonds, che vive all'inizio della via, sta organizzando una festa a cui ha invitato una trentina di amici, che arriveranno scaglionati in  $n = 15$  automobili (l' $i$ -esima automobile ha lunghezza  $\lambda_i$  espressa in metri nella tabella sotto).

| $i$         | 1 | 2   | 3 | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  | 13  | 14  | 15 |
|-------------|---|-----|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|
| $\lambda_i$ | 4 | 4.5 | 5 | 4.1 | 2.4 | 5.2 | 3.7 | 3.5 | 3.2 | 4.5 | 2.3 | 3.3 | 3.8 | 4.6 | 3  |

Per evitare di dare fastidio ai vicini, Mr. Edmonds vorrebbe trovare una disposizione dei parcheggi sui due lati di Dantzig street in modo che il segmento della via occupato dalle automobili dei suoi ospiti sia di lunghezza minima. Si formuli un modello di PLI per risolvere il problema.

Come cambia il modello sapendo che su esattamente uno dei due lati la fila di automobili non può eccedere una determinata lunghezza  $L = 15\text{m}$ ?

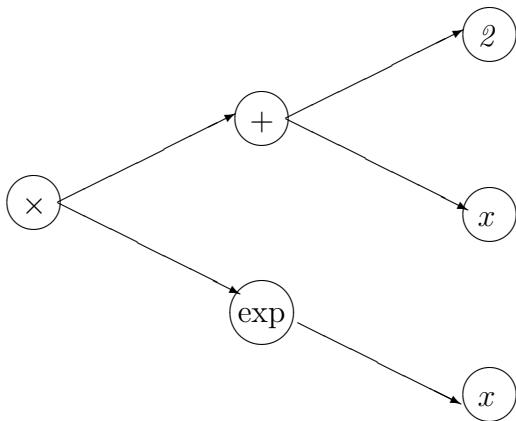


## Capitolo 3

# Problemi difficili

### 3.1 Checksum sui dati

Un *expression parser* è un programma che serve a leggere espressioni matematiche, e a calcolarne il valore su un insieme di dati numerici da assegnare alle variabili. Dalla lettura della stringa contenente l'espressione matematica risulta poi la creazione di una struttura dati ad *albero binario diretto* (senza cicli), in cui i nodi finali (quelli in cui la stella uscente è vuota) rappresentano le variabili o i coefficienti numerici, mentre tutti gli altri nodi, che hanno stelle uscenti di cardinalità 1 o 2, rappresentano gli operatori aritmetici (+, -, \*, /, potenza) e le funzioni trascendenti (sin, cos, tan, log, exp). Gli operatori matematici unari (ad es. sin(), che ha un solo argomento) vengono rappresentati con un nodo in cui la stella uscente ha cardinalità 1, gli operatori binari con un nodo in cui la stella uscente ha cardinalità 2. Per esempio, in figura sotto è riportato l'albero binario per l'espressione  $(x + 2)e^x$ .



L'*expression parser* consiste di diverse subroutine:

- `main()`: coordina tutta l'operazione;
- `parse()`: effettua la lettura della stringa contenente l'espressione matematica;
- `gettoken()`: ritorna il prossimo simbolo semantico dell'espressione matematica (variabile, numero, operatore aritmetico, funzione, parentesi) dal buffer;
- `ungettoken()`: rimette il prossimo simbolo semantico nel buffer;
- `readexpr()`: legge le operazioni di precedenza 4 (minore) (+,-);

- `readterm()`: legge le operazioni di precedenza 3 (\*,/);
- `readpower()`: legge le operazioni di precedenza 2 (potenza);
- `readprimitive()`: legge le operazioni di precedenza 1 (funzioni, espressioni dentro parentesi);
- `sum()`: crea un albero contenente i 2 addendi;
- `difference()`: crea un albero contenente i 2 termini della sottrazione;
- `product()`: crea un albero contenente i 2 fattori della moltiplicazione;
- `fraction()`: crea un albero con il numeratore e il denominatore;
- `power()`: crea un albero contenente la base e l'esponente;
- `minus()`: crea un albero contenente il termine  $f$  con cui si vuole esprimere  $-f$ ;
- `logarithm()`: crea un albero contenente l'argomento della funzione;
- `exponential()`: crea un albero contenente l'argomento della funzione;
- `sine()`: crea un albero contenente l'argomento della funzione;
- `cosine()`: crea un albero contenente l'argomento della funzione;
- `tangent()`: crea un albero contenente l'argomento della funzione;
- `variable()`: crea un nodo contenente una variabile;
- `number()`: crea un nodo contenente un numero;
- `readdata()`: legge i dati numerici da un file;
- `evaluate()`: attraversa ricorsivamente la struttura ad albero dell'espressione matematica sostituendo ad ogni variabile il valore numerico relativo e calcolando via via il risultato dell'espressione;
- `printresult()`: stampa i risultati.

La gerarchia di chiamata di queste subroutine (e la quantità di dati scambiati tra coppie di subroutine) è come segue:

- main: readdata (64KB), parse (2KB), evaluate (66KB), printresult(64KB)
- evaluate: evaluate (3KB)
- parse: gettoken (0.1KB), readexpr (1KB)
- readprimitive: gettoken (0.1KB), variable (0.5KB), number (0.2KB), logarithm (1KB), exponential (1KB), sine (1KB), cosine (1KB), tangent (1KB), minus (1KB), readexpr (2KB)
- readpower: power (2KB), readprimitive (1KB)
- readterm: readpower (2KB), product (2KB), fraction (2KB)
- readexpr: readterm (2KB), sum (2KB), difference (2KB)
- gettoken: ungettoken (0.1KB)



Ogni chiamata di subroutine comporta uno scambio dati bidirezionale tra la subroutine chiamante e quella chiamata. Per garantire l'integrità dei dati scambiati tra le subroutine, si vuole imporre un'operazione di checksum sui dati scambiati tra coppie di subroutine (una coppia di subroutine dove avviene il calcolo di checksum è chiamata una *coppia checksum*). Poiché il calcolo del checksum è costoso in termini di tempo CPU, ci si accontenta di calcolare i checksum in modo che nessuna subroutine faccia parte di più di una coppia checksum. Inoltre, per massimizzare la quantità di dati verificati, si vuole scegliere l'opportuno insieme di coppie checksum in modo che la quantità di dati scambiati tra le coppie sia massimo.

1. Si formuli un modello matematico che risolve il problema di ottimizzazione così espresso. Si scrivano i relativi file di modello e di dati in AMPL, e si risolva il problema con il software AMPL.
2. Come si devono modificare il modello matematico e i files di Mosel per assicurarsi che `readprimitive()` e `readexpr()` siano una coppia checksum? Come cambia la soluzione?

## 3.2 Problema delle otto regine

Formulare un modello di PLI che risolva il problema di posizionare otto regine sulla scacchiera facendo in modo che nessuna di esse sia minacciata da qualcuna delle altre. Risolvere con AMPL. [Belotti]

## 3.3 Gestione di un'impresa

Un'impresa di produzione produce un solo tipo di merce. Ci sono 40 operai, ciascuno dei quali produce 20 unità di merce al mese. La domanda fluttua nel corso di un semestre secondo la seguente tabella:

| Mese            | 1   | 2   | 3   | 4   | 5   | 6   |
|-----------------|-----|-----|-----|-----|-----|-----|
| Unità richieste | 700 | 600 | 500 | 800 | 900 | 800 |

Per adattare la produzione alle fluttuazioni, si impiega lavoro straordinario (ogni operaio può produrre al massimo 6 unità in più al mese, al costo unitario di 5 euro), si conserva merce in magazzino (l'uso del magazzino comporta un costo di 10 euro al mese per unità di merce), si assume o licenzia personale (il numero di operai può variare al massimo di 5 unità in più o in meno ogni mese, con un sovrapprezzo di 500 euro per ogni assunzione e 700 euro per ogni licenziamento).

All'inizio del periodo di gestione il magazzino è vuoto, e si richiede che debba essere vuoto anche alla fine del semestre. Si vuole pianificare la gestione dell'impresa in modo da massimizzare i guadagni. Risolvere il problema presentando una formulazione matematica, un modello di AMPL con il relativo file di dati, e la soluzione ottenuta con il solutore CPLEX. Qual è il valore della funzione obiettivo se si risolve un rilassamento continuo del problema? [Amaldi]

## 3.4 Pianificazione della produzione alla DEC

L'esempio considerato qui è un problema reale che la Digital Equipment Corporation (DEC) ha dovuto affrontare nell'autunno del 1988 per preparare la pianificazione di produzione per l'inverno.

Nella primavera dello stesso anno, la DEC aveva introdotto una nuova linea di workstation e server con CPU singola: GP-1, GP-2 e GP-3 (server con differenti caratteristiche) e WS-1 e WS-2 (workstation). Le caratteristiche sono riassunte nella tabella sotto.

| Sistema | Prezzo | Media dischi fissi | Moduli di RAM |
|---------|--------|--------------------|---------------|
| GP-1    | 60     | 0.3                | 4             |
| GP-2    | 40     | 1.7                | 2             |
| GP-3    | 30     | 0                  | 2             |
| WS-1    | 30     | 1.4                | 2             |
| WS-2    | 15     | 0                  | 1             |

Il prezzo è in migliaia di dollari, e la “media dischi fissi” è il numero medio di dischi fissi montati su ogni unità: per esempio, tutti i GP-2 hanno almeno un disco fisso, e 7 su 10 ne hanno 2.

Le consegne di questa nuova linea di prodotti erano iniziate durante l’estate ed incrementate lentamente nell’autunno. Per l’inverno, erano state anticipate le seguenti difficoltà.

- (i) La produzione interna di CPU aveva limitato la produzione a 7000 unità per via di problemi di debugging.
- (ii) La fornitura di dischi fissi era stata stimata nella gamma 3000-7000 unità.
- (iii) La fornitura dei moduli di RAM era limitata alla gamma 8000-16000 unità.

La sezione marketing aveva stabilito che la domanda durante l’inverno sarebbe stata: 1800 GP-1, 300 GP-3 e 3800 unità di tutta la classe “server”; e 3200 unità di tutta la classe “workstation”. Inclusi in queste proiezioni c’erano anche 500 ordini di GP-2, 500 di WS-1 e 400 di WS-2 che erano già stati ricevuti ma non ancora spediti ai clienti. Nei precedenti periodi di pianificazione, per far fronte alla limitazione di dischi fissi, la DEC aveva prodotto GP-1, GP-3 e WS-2 senza dischi fissi (anche se 3 clienti su 10 richiedevano GP-1 con il disco fisso) e GP-2 e WS-1 con soltanto un disco fisso. Questa configurazione viene chiamata “produzione vincolata”.

Per l’inverno, la DEC avrebbe dovuto prendere le decisioni seguenti.

- (a) Il piano di produzione per il periodo invernale.
- (b) Era opportuno continuare a costruire in “produzione vincolata” o era meglio cercare di soddisfare le richieste dei clienti?
- (c) Era meglio cercare di concentrare gli sforzi per produrre più dischi fissi o più moduli di memoria?

Prevedendo il caso peggiore sulla produzione di dischi fissi e moduli di RAM, si formuli un modelli di PL per prendere le decisioni (a) e (b) sopra. Si usino le tecniche di analisi di sensitività per prendere la decisione (c). I problemi di PL possono essere risolti numericamente mediante AMPL e CPLEX. [Problema sorto alla Digital Equipment Corporation, tratto da Bertsimas e Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, Belmont, 1997.]

[*Suggerimento*: per prendere la decisione (b), si risolva il problema con due distinti file di dati. La “produzione vincolata” corrisponde a valori di “media dischi fissi” diversi da quelli dati nella tabella sopra.]

### 3.5 Problema del commesso viaggiatore

Un commesso viaggiatore deve visitare 7 clienti in modo da minimizzare la distanza percorsa. Le distanze (in Km) tra ognuno dei clienti sono come segue:

|   | 1 | 2  | 3  | 4  | 5  | 6  | 7  |
|---|---|----|----|----|----|----|----|
| 1 | - | 86 | 49 | 57 | 31 | 69 | 50 |
| 2 |   | -  | 68 | 79 | 93 | 24 | 5  |
| 3 |   |    | -  | 16 | 7  | 72 | 67 |
| 4 |   |    |    | -  | 90 | 69 | 1  |
| 5 |   |    |    |    | -  | 86 | 59 |
| 6 |   |    |    |    |    | -  | 81 |

(la matrice delle distanze è simmetrica).

Formulare un modello di PLI e proporre un programma in AMPL che risolva il problema. Si proponga inoltre un possibile approccio euristico per la soluzione del problema, considerando che sulla matrice delle distanze vale la disuguaglianza triangolare.

### 3.6 Controllo ottimo di un razzo

Un razzo di massa  $m$  deve essere portato dal livello del mare a un'altezza  $H$  in tempo  $t$ . Sia  $y(t)$  l'altezza del razzo al tempo  $t$  e  $u(t)$  la forza agente in direzione verticale al tempo  $t$ , che non può eccedere un determinato valore  $b$ . Si assume che il razzo non perda massa durante il movimento, e che l'accelerazione di gravità  $g$  sia costante nell'intervallo  $[0, H]$ . Discretizzando il tempo  $t$  in un'orizzonte  $[0, T]$  in  $n$  intervalli, si scriva un modello di programmazione lineare che determini (per ogni  $k \leq n$ ) la forza  $u(t_k)$  agente sul razzo in modo che l'energia consumata sia minima. Sapendo che  $m = 2140\text{kg}$ ,  $H = 23\text{km}$ ,  $T = 1\text{min}$ ,  $b = 10000\text{N}$  e  $n = 20$ , si ricavi la soluzione del problema mediante AMPL.

Come cambia il modello se si tiene conto che il razzo, bruciando il combustibile, perde massa in ragione di  $\alpha u(t)$   $\text{kg s}^{-1}$ , e che la massa di combustibile iniziale è  $c$  (si assuma che la massa  $m$  del razzo è  $m_0 + c$ , dove  $m_0$  è la massa del razzo senza il combustibile)?

### 3.7 Doppio monopolio

Nel 2021, tutto il potere di credito economico mondiale è concentrato nella banca AA (Autorità Antidemocratiche). Nel 2022, un giudice eroico riesce a far applicare le leggi antitrust (ormai fuori moda, ma mai abrogate, un po' come i Regi Decreti che ancora popolano il corpus delle nostre leggi) alla banca AA prima di essere brutalmente sgozzato dai suoi scagnozzi. Si forma dunque una situazione di "doppio monopolio" dove nasce la banca BB (Banda Bassotti). In un vortice di atti in flagrante illegalità che passano tuttavia sotto silenzio dato che il grande pubblico è intento a seguire il programma televisivo "The International Big Brother 26", la banca AA riesce a far inserire un codicillo nella legge che prevede che la BB non può fare credito ad alcun cliente senza l'approvazione della AA. Tuttavia, in un'era in cui il conflitto di interessi è ormai largamente encomiato oltre che accettato, gli avvocati della AA sono gli stessi della BB, e riescono a far inserire lo stesso codicillo anche in favore della BB. A cose fatte, come in tutte le situazioni di concorrenza scomode per le grandi ditte, le banche AA e BB si riuniscono e mettono a punto dei metodi per intascare più soldi che si può senza pestarsi eccessivamente i piedi: viene deciso che ogni banca deve, indipendentemente dal codicillo, fare almeno un investimento. A operazioni iniziate, si presenta la seguente situazione: ci sono  $n = 6$  grandi clienti (evidentemente la situazione di monopolio si è estesa anche in altri settori industriali ed economici: come s'è visto sopra, anche negli studi di avvocati) e una miriade di piccoli risparmiatori che vengono progressivamente privati di tutti i loro risparmi, e che possono essere trascurati ai fini di questo esercizio e — diciamo — anche ad ogni altro fine. L'effetto del codicillo e del seguente accordo risulta in una situazione in cui la quantità di soldi guadagnati dalla banca AA in un dato investimento è esattamente la quantità di soldi persi da BB per non aver potuto fare lo stesso investimento. La seguente matrice  $A = (a_{ij})$ , di dimensioni  $n \times n$ ,

rappresenta i guadagni e le perdite delle due banche:

$$A = \begin{pmatrix} 1 & 2 & 1 & 3 & 5 & 2 \\ 4 & 1 & 1 & 3 & 2 & 1 \\ 1 & 10 & 6 & 3 & 1 & 4 \\ 2 & 2 & 1 & 3 & 3 & 8 \\ 12 & 1 & 3 & 4 & 2 & 1 \\ 3 & 3 & 1 & 4 & 2 & 2 \end{pmatrix}$$

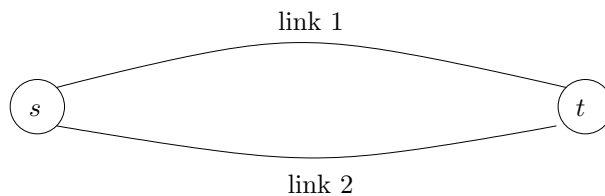
l'elemento  $a_{ij}$  rappresenta il guadagno di AA e la perdita di BB se AA dà tutto il credito al cliente  $i$  e BB al cliente  $j$  (ovviamente, la banca BB considera la matrice  $-A$ ). Le banche possono, altrettanto ovviamente, decidere di fare credito anche a più di un cliente, ma senza eccedere il loro budget d'investimento (che si assume uguale a 1 fantastilione di dollari). Risulta evidente perfino ai manager delle banche che la strategia ottimale di AA è di massimizzare il minor guadagno atteso, e quella di BB di minimizzare la massima perdita attesa. Si scrivano 2 modelli di PL, uno per i guadagni della banca AA e uno per le perdite della banca BB, e si commenti sulle relazioni tra i due modelli.

## Capitolo 4

# Applicazioni in ambito telecomunicazioni

### 4.1 Instradamento di pacchetti

Ci sono  $n$  flussi di dati che devono essere instradati dal nodo  $s$  al nodo  $t$  seguendo uno di due link possibili, con capacità rispettivamente  $u_1 = 1\text{Mbps}$  e  $u_2 = 2\text{Mbps}$ .



La società che gestisce il link 2 è del 30% più costosa di quella che gestisce il link 1. La tabella seguente riporta la quantità di risorse consumate dall' $i$ -esimo flusso, e il costo di instradare l' $i$ -esimo flusso sul link 1.

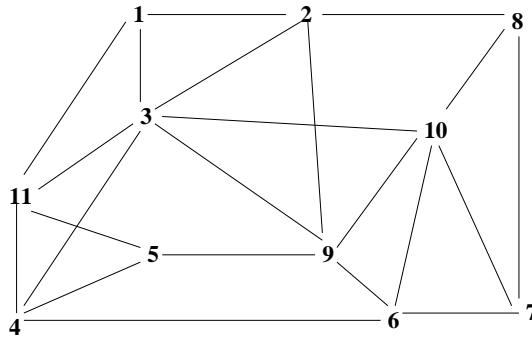
| Flusso | Risorse richieste (Mbps) | Costo sul link 1 |
|--------|--------------------------|------------------|
| 1      | 0.3                      | 200              |
| 2      | 0.2                      | 200              |
| 3      | 0.4                      | 250              |
| 4      | 0.1                      | 150              |
| 5      | 0.2                      | 200              |
| 6      | 0.2                      | 200              |
| 7      | 0.5                      | 700              |
| 8      | 0.1                      | 150              |
| 9      | 0.1                      | 150              |
| 10     | 0.6                      | 900              |

Si formuli il problema di minimizzare il costo di instradamento totale dei flussi rispettando le capacità dei link.

Come va cambiata la formulazione del problema se c'è un numero  $m$  di link possibili?

### 4.2 Affidabilità di una rete

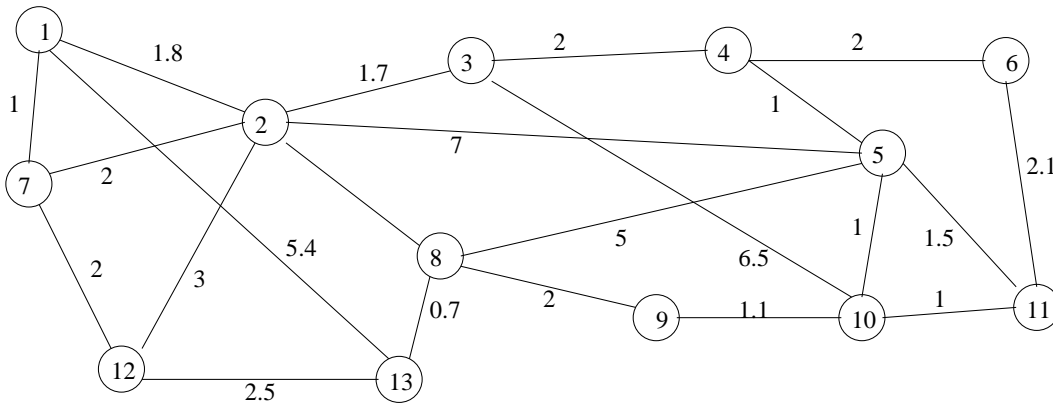
Si consideri la rete che consiste di undici siti connessi da linee bidirezionali per la trasmissione dei dati, mostrata sotto.



Per ragioni di affidabilità, le specifiche di rete richiedono che i due siti 10 e 11 rimangano comunicanti anche se altri tre nodi siano malfunzionanti. La rete mostrata sopra soddisfa a queste specifiche? [Tratto da Guéret, Prins e Sevaux, *Applications of Optimization with XPress-MP*, Dash Optimization, 2002]

### 4.3 Network Design

Telecom Italia, gestore unico della rete di telecomunicazioni nella figura sotto (i costi sui lati sono le distanze  $d(i, j)$  tra i nodi, in decine di Km), ha deciso di suddividere la rete in tre sotto-reti più piccole, di almeno  $m = 2$  nodi l'una, per darle in gestione a Wind, Tre e Vodafone. Si rende perciò necessaria la progettazione di una rete backbone (di gestione Telecom Italia) che assicuri la connettività tra le tre sottoreti. Potenziare un collegamento nella rete esistente in modo che diventi un "collegamento backbone" ha un costo  $c = 25$  Keuro al chilometro. Si tratta di individuare le tre sottoreti in modo che il costo (in euro) di realizzazione della backbone sia minimo. Come cambia la soluzione se Telecom decide di suddividere la rete in quattro sottoreti anziché tre?



### 4.4 Network Routing

Si vuole rinnovare la rete dorsale telefonica tra le principali sedi del Politecnico di Milano (Milano, Como, Lecco, Piacenza, Cremona). È data la topologia della rete, come nella figura sotto.



Per ogni collegamento è specificata una coppia  $(u, c)$  dove  $u$  è la capacità di traffico in Mb/s e  $c$  la lunghezza in Km.

1. Como, Lecco: (200, 30)
2. Como, Milano: (260, 50)
3. Como, Piacenza: (200, 110)
4. Lecco, Milano: (260, 55)
5. Lecco, Cremona: (200, 150)
6. Milano, Piacenza: (260, 72)
7. Milano, Cremona: (260, 90)
8. Piacenza, Cremona: (200, 100)

È noto un insieme di richieste di traffico (espresse in Mb/s) tra coppie ordinate di sedi:

1. Como, Lecco: 20
2. Como, Piacenza: 30
3. Milano, Como: 50
4. Milano, Lecco: 40
5. Milano, Piacenza: 60
6. Milano, Cremona: 25
7. Cremona, Lecco: 35
8. Cremona, Piacenza: 30

È dato inoltre un insieme iniziale di percorsi:

1. Como  $\rightarrow$  Lecco
2. Como  $\rightarrow$  Milano  $\rightarrow$  Piacenza

3. Milano  $\rightarrow$  Lecco  $\rightarrow$  Como
4. Milano  $\rightarrow$  Como  $\rightarrow$  Lecco
5. Milano  $\rightarrow$  Piacenza
6. Milano  $\rightarrow$  Piacenza  $\rightarrow$  Cremona
7. Cremona  $\rightarrow$  Milano  $\rightarrow$  Piacenza  $\rightarrow$  Como  $\rightarrow$  Lecco
8. Cremona  $\rightarrow$  Milano  $\rightarrow$  Como  $\rightarrow$  Lecco  $\rightarrow$  Milano  $\rightarrow$  Piacenza

Si richiede di reinstradare tutte le richieste di traffico in modo che la somma delle lunghezze dei cammini tra ogni coppia di sedi sia minimo. [*Belotti e Liberti*]



# Capitolo 5

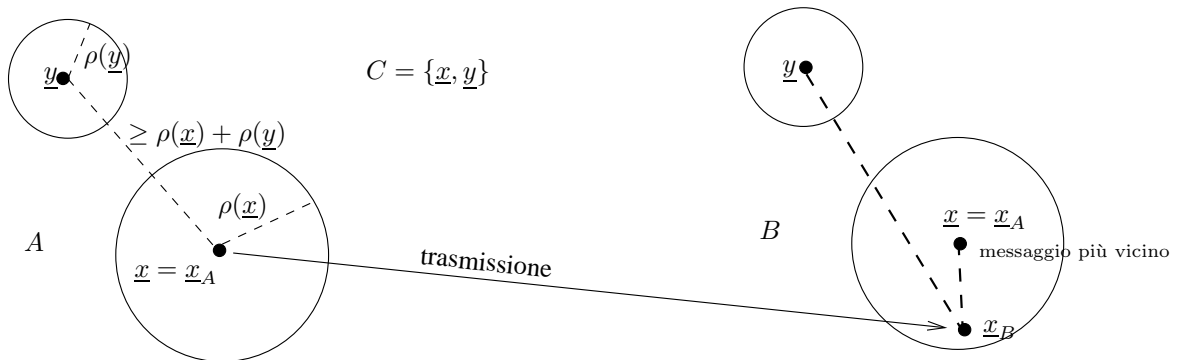
## Problemi non lineari

### 5.1 Codici correttori di errore

Un messaggio spedito da  $A$  a  $B$  viene rappresentato da un vettore  $\underline{z} = (z_1, \dots, z_m) \in \mathbb{R}^m$ . Un *codice correttore di errori*, nella sua accezione più generale, è un insieme  $C$  di messaggi (cioè di vettori in  $\mathbb{R}^m$ ) con una funzione associata  $\rho : C \rightarrow \mathbb{R}$ , tale che per ogni coppia di messaggi distinti  $\underline{x}, \underline{y} \in C$ , si abbia  $\|\underline{x} - \underline{y}\| \geq \rho(\underline{x}) + \rho(\underline{y})$ . Si assume che  $C$  abbia cardinalità finita  $n$ , sicché  $\rho$  è nota quando sono noti i valori  $\rho_i = \rho(\underline{x}^i)$ , dove  $C = \{\underline{x}^i \mid i \leq n\}$ . Il *raggio di correzione* del codice  $C$  è dato da

$$R_C = \min_{\underline{x} \in C} \rho(\underline{x}),$$

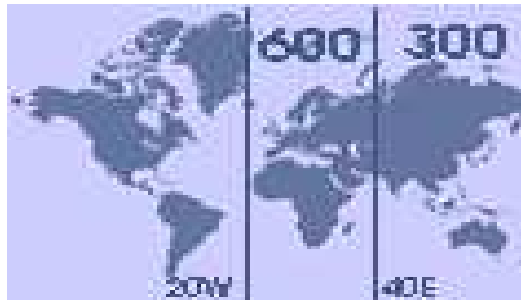
e rappresenta il massimo errore che si riesce a correggere su ogni messaggio del codice. Supponiamo che  $A$  e  $B$  conoscano entrambi il codice  $C$ , e che la linea di comunicazione sia disturbata. Un messaggio  $\underline{x}_A \in C$  che viene spedito da  $A$ , per via degli errori di comunicazione, viene ricevuto in  $B$  come  $\underline{x}_B \notin C$ . Supponendo che l'errore di  $\underline{x}_B$  sia strettamente minore di  $R_C$ , il ricevente in  $B$  è in grado di risalire al messaggio originale  $\underline{x}_A$  cercando il messaggio  $\underline{x} \in C$  più vicino a  $\underline{x}_B$ , come nella figura sotto.



Si proponga un modello di PNL per costruire un codice correttore di errori  $C$  di 10 messaggi in  $\mathbb{R}^{12}$ , dove tutte le coordinate dei messaggi devono essere comprese tra 0 e 1, in modo da massimizzare il raggio di correzione.

## 5.2 Manutenzione aerea

La Boeing deve costruire 5 centri di manutenzione aerea che servano l'area euro-asiatica. Il costo di costruzione di ogni centro è di 300 milioni di euro nell'area europea (tra 20°W e 40°E) e di 150 nell'area asiatica (tra 40°E e 160°E), come nella figura sotto. Ogni centro può servire 60 aviogetti/anno.



I centri dovranno servire gli aeroporti dove si concentrano i maggiori clienti Boeing, come dettagliato nella tabella sotto (nome dell'aeroporto, coordinate geografiche, numero atteso di aviogetti/anno che avranno bisogno di manutenzione).

| Aeroporto           | Coordinate |       | N. aviogetti |
|---------------------|------------|-------|--------------|
| London Heathrow     | 51°N       | 0°W   | 30           |
| Frankfurt           | 51°N       | 8°E   | 35           |
| Lisboa              | 38°N       | 9°W   | 12           |
| Zürich              | 47°N       | 8°E   | 18           |
| Roma Fiumicino      | 41°N       | 12°E  | 13           |
| Abu Dhabi           | 24°N       | 54°E  | 8            |
| Moskva Sheremetyevo | 55°N       | 37°E  | 15           |
| Vladivostok         | 43°N       | 132°E | 7            |
| Sydney              | 33°S       | 151°E | 32           |
| Tokyo               | 35°N       | 139°E | 40           |
| Johannesburg        | 26°S       | 28°E  | 11           |
| New Dehli           | 28°N       | 77°E  | 20           |

Il costo totale di un centro di manutenzione è dato dal costo di costruzione sommato al costo atteso di servizio. Il costo di servizio di ogni aviogetto dipende linearmente dalla distanza che questo deve coprire per raggiungere il centro di manutenzione, con una costante di proporzionalità di 50 euro/Km. Si assume che la terra sia una sfera perfetta e che la distanza più breve tra due punti di coordinate geografiche  $(\delta_1, \varphi_1)$  e  $(\delta_2, \varphi_2)$  sia data da:

$$d(\delta_1, \varphi_1, \delta_2, \varphi_2) = 2r \operatorname{asin} \sqrt{\sin^2 \left( \frac{\delta_1 - \delta_2}{2} \right) + \cos \delta_1 \cos \delta_2 \sin^2 \left( \frac{\varphi_1 - \varphi_2}{2} \right)},$$

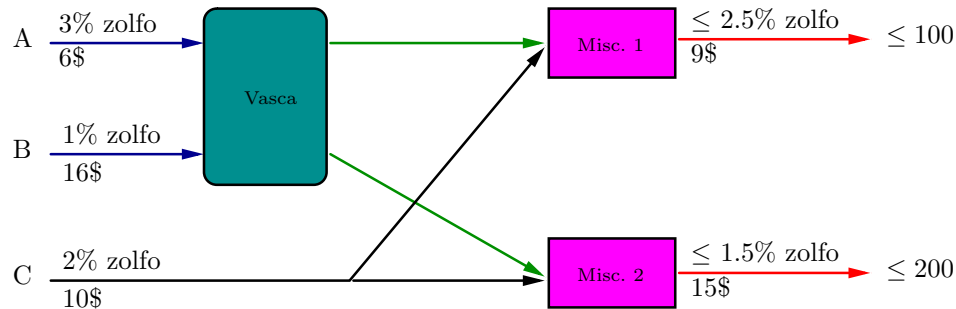
dove  $r$ , il raggio terrestre, è di 6371Km.

Si formuli un modello di PNL per decidere dove localizzare i centri di manutenzione minimizzando i costi dell'operazione.

## 5.3 Miscelazione di greggio

In una raffineria l'impianto di miscelazione è composto da una vasca e due miscelatori, come nella figura sotto. La vasca ha due input A,B di greggio i cui costi unitari e percentuale di zolfo sono rispettiva-

mente 6\$, 16\$ e 3%, 1%. Il contenuto della vasca è portato ai miscelatori, che hanno anche un input supplementare C di greggio di costo unitario 10\$ e una percentuale di zolfo del 2%. Il miscelatore 1 deve produrre un petrolio di costo unitario 9\$, garantito al 2.5% di zolfo, mentre il miscelatore 2 deve produrre un petrolio più puro, di costo unitario 15\$, garantito al 1.5% di zolfo. La domanda massima di acquisto del mercato è di 100 unità di petrolio raffinato di tipo 1 e 200 unità di petrolio raffinato di tipo 2. Si scriva un modello di PNL per risolvere il problema di determinare le quantità di greggio di tipo A, B, C richieste dall'impianto al fine di massimizzare il guadagno. Il problema è convesso?



[Tratto da Haverly, *Studies of the behaviour of recursion for the pooling problem*, ACM SIGMAP Bulletin 25:19-28, 1978]

## 5.4 Controllo ottimale di un razzo

Un razzo di massa  $m$  deve essere portato dal livello del mare a un'altezza  $H$  in tempo  $t$ . Sia  $y(t)$  l'altezza del razzo al tempo  $t$  e  $u(t)$  la forza agente in direzione verticale al tempo  $t$ , che non può eccedere un determinato valore  $b$ . Il razzo, bruciando il combustibile, perde massa in ragione di  $\alpha u(t)$  kg s<sup>-1</sup>, e la massa di combustibile iniziale è  $c$  (si assuma che la massa iniziale  $m$  del razzo sia  $m_0 + c$ , dove  $m_0$  è la massa del razzo senza il combustibile). Si supponga che l'accelerazione di gravità  $g$  sia costante nell'intervallo  $[0, H]$ .

- Discretizzando il tempo  $t$  in un'orizzonte  $[0, T]$  in  $n$  intervalli, si scriva un modello di PNL che determini (per ogni  $k \leq n$ ) la forza  $u(t_k)$  agente sul razzo in modo che la quantità di carburante consumata sia minima.
- Si dica come si può approssimare la formulazione ottenuta a un problema non vincolato, si commenti sulle proprietà di convessità del problema approssimato, e lo si risolva con MATLAB utilizzando un metodo opportuno.

[Suggerimento. Si introduca un termine di penalità per ogni vincolo di equazione e una barriera logaritmica per ogni vincolo di disuguaglianza.]

## 5.5 Lotti ottimi

Consideriamo un problema industriale affrontato dalla Pfizer Pharmaceuticals. La produzione di farmaci si basa su una successione di reazioni chimiche organiche effettuate in grandi recipienti da migliaia di litri. Ogni lotto di prodotto è suddiviso in un certo numero di sotto-lotti. Il problema è quello di determinare il numero di sotto-lotti in ogni lotto in modo da minimizzare i costi. I costi sono di due tipi: costi di "changeover" tra un lotto e l'altro (elevati perché bisogna pulire i recipienti e riconfigurare l'equipaggiamento) e costi di magazzino. In particolare, via via che un sotto-lotto viene processato, le quantità già pronte devono essere messe in magazzino fino a che l'intero sotto-lotto non sia disponibile per

la prossima fase di produzione. Si considerino  $j = 1, \dots, n$  tipi di farmaci con produzione in  $k = 0, \dots, m$  fasi (la fase  $k = 0$  si riferisce alle componenti del farmaco allo stato grezzo, prima dell'inizio della produzione vera e propria), con i seguenti parametri:

- $d_j$  = numero di sotto-lotti di prodotto  $j$  richiesti annualmente dal mercato
- $t_{jk}$  = tempo (in frazioni di settimane) di "changeover" per lotto richiesto dal prodotto  $j$  alla fase  $k$
- $p_{jk}$  = tempo (in frazioni di settimane) di processamento per sotto-lotto di prodotto  $j$  alla fase  $k$
- $v_{jk}$  = valore monetario per sotto-lotto di prodotto  $j$  alla fine della fase  $k$ .

Si assume inoltre che siano disponibili  $T$  settimane per la produzione degli  $n$  farmaci, che il magazzino costi settimanalmente una frazione  $a$  del valore del prodotto, e che i costi di "changeover" siano  $c$  alla settimana.

Si proponga un modello di PNL per determinare la divisione in sotto-lotti che minimizza i costi. [Tratto da Rardin, *Optimization in Operations Research*, Prentice Hall, Upper Saddle River (NJ), 1998]

[*Suggerimento.* Il costo dato da un singolo prodotto è di (numero lotti / anno)  $\times$  ((costi di "changeover") + (costi di magazzino)). L'unico vincolo è quello sul tempo totale di produzione.]