1. Si traduca nel linguaggio **AMPL** (file .mod) il seguente modello di programmazione lineare intera, relativo a un problema di produzione:

$$\max \sum_{i \in I} P_i\, x_i$$

$$\text{s.t.} \quad \sum_{i \in I} A_{ij}\, x_i \le B_j \quad , \quad \forall\, j \in J$$

$$\sum_{i \in I} (C_i\, x_i + F_i\, y_i) \le W$$

$$x_i \le M y_i \quad , \quad \forall\, i \in I$$

$$x_i \in \mathbb{Z}_+ \, , \qquad y_i \in \{0,1\}, \qquad \forall\, i \in I$$

dove I è l'insieme dei prodotti, J è l'insieme delle materie prime. Le variabili sono: x_i la quantità di prodotti (si possono realizzare solo quantità intere), y_i binaria con valore 1 se viene realizzata almeno un'unità del prodotto corrispondente, 0 altrimenti. I dati sono: P_i il profitto unitario per il prodotto $i \in I$, B_j la quantità disponibile di materia prima $j \in J$, A_{ij} la quantità di materia prima j utilizzata per unità di prodotto i, C_i il costo di produzione per unità di prodotto $i \in I$, F_i il costo fisso per la produzione di prodotti $i \in I$, W il budget disponibile, M una costante sufficientemente grande (ad esempio pari a W).

```
set I; #insieme dei prodotti
set J; #insieme delle materie prime

param C{I}; #costo di produzione
param P{I}; #profitto unitario
param F{I}; #costo fisso
param B{J}; #materia prima disponibile
param A{I, J};
param W; #budget
param M default W; #costante

var x{I} >=0 integer;
var y{I} binary;

maximize costo: sum{i in I} P[i]*x[i];
s.t. materiaprima{j in J}: sum{i in I} A[i,j]*x[i] <= B[j];
s.t. costo_fisso: sum{i in I} (C[i]*x[i] + F[i]*y[i]) <= W;
s.t. bigM {i in I}: x[i] <= M*y[i];
```

2. Per l'esercizio precedente, si consideri $I = \{auto, moto, bicicletta, monopattino\}$ e $J = \{ruote, tubi, bulloni\}$ e si dia un esempio di file .dat di **AMPL** che definisca gli insiemi dati e una valorizzazione per i parametri P, A, B, C, F, W e M (si scelgano dei valori a piacere).

```
set I := auto, moto, bicicletta, monopattino;
set J := ruote, tubi, bulloni;

param C := auto 5 moto 10 bicicletta 15 monopattino 20;
param P := auto 10 moto 15 bicicletta 20 monopattino 25;
param F := auto 7 moto 8 bicicletta 9 monopattino 10;
param B := ruote 4 tubi 6 bulloni 8;
param W := 100;
param A :           ruote        tubi         bulloni :=
auto                1            2            3
moto                4            5            6
bicicletta          7            8            9
monopattino         10           11           12
;
```

Si vuole risolvere il seguente problema con **AMPL**. Un'azienda produce aranciata e concentrato in polvere e ha a disposizione 1 tonnellata di arance e 10 000 litri di acqua minerale. Per ogni confezione da un litro di aranciata si consumano 500 grammi di arance e 2 litri di acqua (parte in bottiglia, parte usata dal processo produttivo); per ogni confezione da un kg di concentrato si consumano 5 kg di arance e si ottengono 0,5 litri di acqua minerale disponibile per la produzione di aranciata. Il ricavo da ciascuna confezione di aranciata e di concentrato è, rispettivamente, di 80 eurocent e 2 euro, rispettivamente. Il generico modello di programmazione lineare intera per la massimizzazione dei ricavi è il seguente:

$$\max \sum_{i \in I} r_i \, x_i$$

$$\text{s.t.} \quad \sum_{i \in I} a_{ij} \, x_i \le b_j \quad , \quad \forall\, j \in J$$

$$x_i \in \mathbb{Z}_+ \ , \quad \forall\, i \in I$$

dove I è l'insieme dei prodotti (aranciata e concentrato nel caso specifico), J è l'insieme delle risorse (acqua minerale e arance), x_i è la variabile che indica la quantità di prodotto i da produrre, r_i è il ricavo unitario dal prodotto i, b_j è la quantitià di risorsa j disponibile, a_{ij} è la quantità di risorsa j consumata per ogni unità di prodotto i (si noti che, per conentrato e acqua il valore di questo parametro è negativo).

Si traduca **il modello generico dato** in linguaggio **AMPL** (file .mod) e si scriva il corrispondente **file dei dati** (.dat) per la soluzione del problema specifico.

Nell'ordine riportiamo:

- File .mod

```
set I; #prodotti (aranciata e concentrato)
set J; #risorse (acqua minerale e arance)

param B{J}; #risorsa disponibile
param R{I}; #ricavo unitario prodotto
param A{I,J}; #quantità "j" consumata per unità di prodotto "i"

var x{I} >= 0 integer; #quantità di prodotto da produrre

maximize ricavo: sum{i in I} R[i]*x[i];
s.t. soddisfazione{j in J}: sum {i in I} A[i,j]*x[i] <= B[j];
```

- File .dat

```
#convertendo tutti pesi in Kg (grammi/tonnellate)

set I := aranciata concentrato; #prodotti
set J := acqua arance; #risorse

param B := acqua 10000 arance 1000; #disponibilità risorse
param R := aranciata 0.80 concentrato 2; #ricavi

param A :   acqua arance :=
aranciata   2           0.5
concentrato 0.5         5
;
```

4. Si traduca nel linguaggio **AMPL** (file .mod) il seguente modello di programmazione lineare intera (riferibile, ad esempio, a un problema di produzione di prodotti j su più linee i, con costi fissi f di attivazione delle linee, costi orari c per linea e prodotto, produttività oraria a per linea e prodotto, richiesta minima b per prodotto, capacità d per linea). Si dia inoltre una possibile definizione della costante M in funzione dei parametri d del problema.

$$
\begin{aligned}
\min \ & \sum_{i \in I, j \in J} c_{ij}\, x_{ij} + \sum_{i \in I} f_i\, y_i \\
\text{s.t.} \ & \sum_{i \in I} a_{ij}\, x_{ij} \geq b_j \quad , \quad \forall j \in J \\
& \sum_{j \in J} x_{ij} \leq d_i \quad , \quad \forall i \in I \\
& \sum_{j \in J} x_{ij} \leq M y_i \quad , \quad \forall i \in I \\
& x_{ij} \in \mathbb{Z}_+ \ , \quad \forall i \in I, j \in J \\
& y_i \in \{0,1\}, \quad \forall i \in I
\end{aligned}
$$

- File .mod

```
set I; #linee
set J; #prodotti

param F{I}; #costi fissi linea
param C{I,J}; #costi orari linea/prodotto
param A{I,J}; #produttività oraria linea/prodotto
param B{J}; #richiesta minima linea
param D{I}; #capacità per linea

param M default 10000; #big-M constraint rispetto ai parametri D

var x{I, J} >=0 integer;
var y{I} binary;

minimize costo: sum{i in I, j in J}
      C[i,j]*x[i,j] + sum{i in I} F[i]*y[i];
s.t. produttivita{j in J}: sum{i in I} A[i,j]*x[i,j] >= B[j];
s.t. capacita{i in I}: sum{j in J} x[i,j] <= D[i];
s.t. viaggi{i in I}: sum{j in J} x[i,j] <= M*y[i];
```

5. Si consideri il modello dell'esercizio precedente, riferito alla produzione di *auto*, *moto* e *biciclette* (insieme J) su due linee chiamate *linea1* e *linea2* (insieme I): c_{ij} è il costo per ogni ora di lavorazione del prodotto j sulla linea i; f_i è il costo fisso per l'attivazione della produzione sulla linea i; a_{ij} è il numero di prodotti j prodotti per ogni ora di lavorazione sulla linea i; b_j è il numero di prodotti j richiesti; d_i è il numero di ore complessive disponibili sulla linea i; M è una costante sufficientemente grande. Si scriva il file .dat di **AMPL** che definisca i valori dei parametri riassunti nella seguente tabella e un opportuno valore per M.

	Linea 1		Linea 2		Richiesta b
	Costo orario c	Produttività a	Costo orario c	Produttività a	
Auto	6	2	5	3	100
Moto	4	4	3	5	200
Bici	2	6	1	7	300
Costo fisso f	10		15		
Capacità d	800		900		

```
set I := linea1 linea2;
set J := auto moto biciclette;

param F := linea1 10 linea2 15;
param D := linea1 800 linea2 900;

param B := auto 100 moto 200 biciclette 300;

param C (tr):     linea1      linea2 :=
auto              6           5
moto              4           3
biciclette        2           1
;

param A (tr):     linea1      linea2 :=
auto              2           3
moto              4           5
biciclette        6           7
;
```

Esercizio: distribuzione PC (scenari)

Per bilanciare la produzione, l'azienda richiede che nello stabilimento italiano si assemblino almeno il 25% dei PC.

1. Inoltre, l'azienda richiede che nello stabilimento italiano si assemblino almeno il 30% dei PC (**ipotesi 1**) [o il 40% dei PC (**ipotesi 2**)] prodotti in ciascuno degli altri stabilimenti
2. Produrre un elenco che permetta di individuare i casi in cui una banca riceve forniture da un solo paese.
3. Visualizzare l'utilizzo delle capacità produttive per paese.
4. Conviene, nell'ipotesi 2, potenziare di 5000 unità la produzione in Cina, al costo di 4.000 euro?
5. Tornare alla situazione senza bilanciamenti e studiare gli effetti della diminuzione (a intervalli di 6 euro) del costo di produzione in Italia (diminuzione massima di 40 euro), indicando in quali casi in Italia la produzione complessiva supera quella della Francia.

Insiemi: S (stabilimenti) e B (banche)

Parametri: w_i (costi prod.), c_{ij} (costi trasp.), a_i (capacità prod.), b_j (richieste), α (bilanciamento generale), β (bilanciamento singolo)

$$\min \sum_{i\in S, j\in B} (w_i + c_{ij})\, x_{ij}$$

$$\text{s.t.} \sum_{j\in B} x_{ij} \le a_i \qquad \forall\, i \in S$$

$$\sum_{i\in S} x_{ij} = b_j \qquad \forall\, j \in B \qquad \text{(no eccessi di produzione} \Rightarrow \text{uguaglianza)}$$

$$\sum_{j\in B} x_{\text{Italia}\ j} \ge \alpha \sum_{i\in S, j\in B} x_{ij}$$

$$\sum_{j\in B} x_{\text{Italia}\ j} \ge \beta \sum_{j\in B} x_{ij} \qquad \forall\, i \in S \setminus \{\text{Italia}\}$$

$$x_{ij} \in Z_+ \qquad \forall\, i \in S, j \in B$$

[Risorse] PC.mod
- PC.dat -
PC.run
(PC_plus.run)

PC.mod:

```
set I; #stabilimenti – S
set J; #banche – B
param A{I}; #capacità produttive
param B{J}; #richieste
param W{I}; #costi produttivi
param C{I, J}; #costi
var x{I, J} integer >= 0;
param stab_bil symbolic in S; #Per mettere 'it' all'interno delle
variabili → expected number
param bil1;
param bil2;
minimize costo: sum{i in I, j in J}
(W[i] + C[i,j])*x[i,j];
s.t. produzione{i in I}: sum{j in J} x[i,j] <= A[i];
s.t. domanda{j in J}: sum{i in I} x[i,j] = B[j];
#Introduciamo i due vincoli di bilanciamento aggiuntivi
#Inseriamo 'it' perché è parte di un insieme, ma è una stringa
#Per fare in modo che AMPL non dipenda dal tipo stringa,
#inseriamo un parametro simbolico "stab_bil", tale che
#non venga visto solo come numero
s.t. bilan1: sum{j in J} x[stab_bil, j] >= 0.25 *
     sum{i in I, j in J} x[i, j];

#Attenzione che, a livello di insieme, deve essere escluso 'it'
s.t. bilan2{i in I: i != 'it'}: sum{j in J} x['it', j] >= 0.30 *
     sum{j in J} x[i, j];
```

La modifica del file *pc.dat* avviene come segue, introducendo anche i parametri:

```
set I := it ch fr;
set J := bi uc av cs bc;
param A := it 10000 ch 20000 fr 10000;
param B := bi 7100 uc 3400 av 9700 cs 5200 bc 3050;
param W := it 220 ch 180 fr 200;
#param :        w     a :=
#it             220   10000
#ch             100   20000
#fr             200   10000
```

```
param C:        bi      uc      av      cs      bc :=
it              5.5     7.5     6.9     8.0     10.3
ch              15      14.3    13.0    16.4    5.0
fr              6.0     7.8     6.3     6.8     11.0
;
param stab_bil := 'it';
param bil1 := 0.25;
param bil2 := 0.30;
```

Dualità in AMPL: un esempio (testo)

- *Modellare il seguente problema, trovare la soluzione ottima e analizzarla alla luce della teoria della dualità*

Un'industria produce due tipi di creme: fondente e gianduia. Per avere un kg di ciascuna crema sono necessari, tra gli altri, due ingredienti grezzi (zucchero e cacao) e la lavorazione su una macchina, come riportato in tabella:

	Fondente	Gianduia
Zucchero (kg)	3	2
Cacao (kg)	4	1
Lavorazione (ore)	2	1

Settimanalmente, si hanno a disposizione al più 1200 Kg di zucchero e al più 1000 Kg di ~~gianduia~~ cacao, mentre la disponibilità massima settimanale di ore lavorative della macchina è pari a 700. Un kg di fondente è venduto a 24 Euro e un kg di gianduia è venduto a 14 Euro. Si vuole pianificare la produzione settimanale in modo da massimizzare il ricavo complessivo.

Dualità in AMPL: un esempio (modelli)

PROBLEMA PRIMALE	PROBLEMA DUALE
$\max 24x_1 + 14x_2$	$\min 1200u_1 + 1000u_2 + 700u_3$
$3x_1 + 2x_2 \le 1200$	$3u_1 + 4u_2 + 2u_3 \ge 24$
$4x_1 + x_2 \le 1000$	$2u_1 + u_2 + u_3 \ge 14$
$2x_1 + x_2 \le 700$	$u_1 \ge 0, u_2 \ge 0, u_3 \ge 0$
$x_1 \ge 0, x_2 \ge 0$	

Implementare il problema primale e il problema duale dell'esempio e:
1. verificare il valore ottimo delle variabili primali e duali
2. verificare il teorema della dualità forte
3. verificare le condizioni di complementarietà primale-duale
4. vedere come cambiano i valori ottimi delle funzioni obiettivo primale e duale variando i termini noti dei vincoli primali, uno per volta
5. dire se esiste una relazione tra il valore ottimo delle variabili duali e le variazioni osservate nel valore ottimo della funzione obiettivo

File .mod

```
set PRODOTTI; #I
set RISORSE; #J
param prezzo{PRODOTTI}; #p{I}
param consumo_risorse{RISORSE, PRODOTTI}; #A{J,I}
param disp{RISORSE}; #d{J}
var x{PRODOTTI}; #x{I}
maximize ricavo: sum{j in PRODOTTI} prezzo[j] * x[j];
s.t. v_disp{i in RISORSE}:
      sum{j in PRODOTTI} consumo_risorse[i,j] * x[j]
      <= disp[i];
```

File .dat

```
set PRODOTTI := fondente gianduia;
set RISORSE := zucchero cacao lavoro;
param prezzo := fondente 24 gianduia 14;
param consumo_risorse :    fondente     gianduia :=
zucchero                   3            2
cacao                      4            1
lavoro                     2            1
;
param disp := zucchero 1200 cacao 1000 lavoro 700;
```

File .run

```
reset;
option solver cplex;
model mixopt2.mod;
data mixopt_creme.dat;
```

```
solve;
display ricavo, x;
```

6. Si consideri il seguente modello di programmazione lineare relativo a un problema di produzione di un insieme di prodotti K, da realizzare con materie prime nell'insieme J, fornite da fornitori nell'insieme I. Sono definiti i parametri: P_k (prezzo di vendita del prodotto k), C_{ij} (costo unitario della materia prima j presso il fornitore i), F_i (costo fisso per fornirsi dal fornitore i), Q_{jk} (quantità di materia j consumata da un'unità di prodotto k), M_i (limite massimo agli acquisti dal fornitore i), e B (budget disponibile).

$$\max \sum_{k \in K} P_k x_k$$

$$\text{s.t.} \quad \sum_{i \in I, j \in J} C_{ij} y_{ij} + \sum_{i \in I} F_i z_i \leq B$$

$$\sum_{i \in I} y_{ij} \geq \sum_{k \in K} Q_{jk} x_k \quad , \forall j \in J$$

$$\sum_{j \in J} y_{ij} \leq M_i z_i \quad , \quad \forall i \in I$$

$$x_k \in \mathbb{Z}_+, y_{ij} \in \mathbb{R}_+, \; z_i \in \{0,1\},$$

$$\forall i \in I, \; j \in J, \; k \in k$$

a. Si traduca nel linguaggio **AMPL** il modello proposto (**file .mod**).

b. Si produca un **file .dat** di esempio per 3 fornitori, 2 materie prime e 3 prodotti.

c. Si scriva uno script di AMPL (**file .run**) che risolve l'istanza specificata e visualizza il valore della funzione obiettivo e delle variabili per una soluzione ottima.

Nell'ordine: file .mod, .run e .dat

```
set I; #fornitori
set J; #materie prime
set K; #prodotti
param P{K}; #prezzo di vendita
param C{I,J}; #costo unitario materia prima
param F{I}; #costo fisso
param Q{J,K}; #quantità materia consumata
param M{I}; #limite massimo acquisti fornitore
param B; #budget
var x{K} integer >=0;
var y{I,J} integer >=0;
var z{I} binary;

maximize ricavo: sum{k in K} P[k]*x[k];
s.t. limite{i in I, j in J}: C[i,j]*y[i,j] + F[i]*z[i] <= B;
s.t. quantita{j in J}: sum{i in I} y[i,j] >= sum{k in K} Q[j,k] * x[k];
s.t. acquisti{i in I}: sum{j in J} y[i,j] <= M[i] * z[i];

#3 fornitori, 2 materie prime, 3 prodotti
set I := forn1 forn2 forn3;
set J := mat1 mat2;
set K := prod1 prod2 prod3;

param P := prod1 10 prod2 20 prod3 30;
param F := forn1 20 forn2 30 forn3 40;
param M := forn1 100 forn2 200 forn3 300;
param B := 5000;

param C (tr):     forn1       forn2       forn3 :=
mat1              10          20          30
mat2              40          50          60
;
```

```
param Q (tr):     mat1  mat2 :=
prod1             5         10
prod2             15        20
prod3             25        30
;


reset;
option solver cplex;
model test.mod;
data file.dat;
solve;
display ricavo, x;
```

Esercizio (Max-Flow)

Si determini, in base alle capacità degli archi, il massimo numero *maxf* di unità che possono essere trasferite da A a F.

Suggerimento: *il problema può essere modellato come un flusso, introducendo un arco fittizio da F ad A cui corrisponde una variabile x_{FA} che indichiamo con y e rappresenta la quantità (incognita) da trasferire da A (con bilanciamento $-y$) a F (con bilanciamento $+y$), e considerando come funzione obiettivo la massimizzazione della stessa variabile.*

- `maxf.mod`, `maxf.dat`

Osservazione: la soluzione rimane intera se le unità sono frazionabili e risolviamo con il simplesso (la matrice dei vincoli è una matrice di incidenza del grafo con l'arco fittizio e quindi resta totalmente unimodulare)

Osservazione: l'interezza del rilassamento continuo si perde con l'introduzione di un vincolo di budget (si osservano soluzioni frazionarie)

Modello PL per Max Flow

Variabili

- quantità x_{ij} da far fluire sull'arco $(i,j) \in A$
- quantità y di flusso che esce da A e arriva in F

$$\max \quad y$$

$$s.t. \quad \sum_{(i,v)\in A} x_{iv} - \sum_{(v,j)\in A} x_{vj} = \begin{cases} -y \text{ se } v = A \\ +y \text{ se } v = F \\ 0 \quad \text{altrimenti} \end{cases} \qquad \forall \ v \in N$$

$$x_{ij} \leq u_{ij} \qquad \forall \ (i,j) \in A$$

$$\left[\sum_{(i,j)\in A} c_{ij} x_{ij} \leq \text{budget} \right]$$

$$x_{ij} \in \mathbb{Z}_+ (\equiv \mathbb{R}_+) \quad [\neq \mathbb{R}_+]$$

File .mod:

```
set N; #nodi
set A within N cross N; #archi definiti su prodotto cartesiano

param c{A}; #i costi sono definiti sugli archi,
#ma sono definiti come coppie di nodi
param b{N}; #parametro di bilanciamento
param W; #budget
param u{A}; #unità del problema

check: sum{i in N} b[i] = 0;
var x{A} integer >=0;
var y;

maximize flow: y; #variabile fittizia
s.t. balance{v in N}: sum{(i,v) in A} x[i,v] - sum{(v,j)in A} x[v,j] = b[v] * y;
#nel vincolo di bilanciamento, si dipende
#dalla var. fittizia che modella
#la quantità da trasferire da A ad F
s.t. capacity {(i,j) in A}: x[i,j] <= u[i,j];
s.t. budget: sum{(i,j) in A} c[i,j]*x[i,j] <= W;
```

File .run:

```
set N := A B C D E F;

set A := A B  A C  B C  B E  B F  B D
            C F  D F  E C  E D  F E;

param:      c     u :=
```

```
  A B        7      20
  A C        8      50
  B C        -1     40
  B D        4      30
  B E        1      20
  B F        5      14
  C F        3      48
  D F        1      33
  E C        -2     21
  E D        0      14
  F E        3      18
;

param b default 0 :=
A -1
F 1
;

param W :=500;
```

Esercizio 1

Risolvere con il metodo del Branch-and-bound:

$$\begin{array}{ll} \min & 1.97x_1 + 3x_2 + 5x_3 + 2.14x_4 + 2x_5 \\ s.t. & -x_1 + 3x_2 + 1x_3 + 2x_4 + x_5 \geq 4 \\ & 2x_1 + 1.5x_2 + 2x_3 + 3x_4 + x_5 \geq 7 \\ & x_1, \ldots, x_5 \in Z^+ \end{array}$$

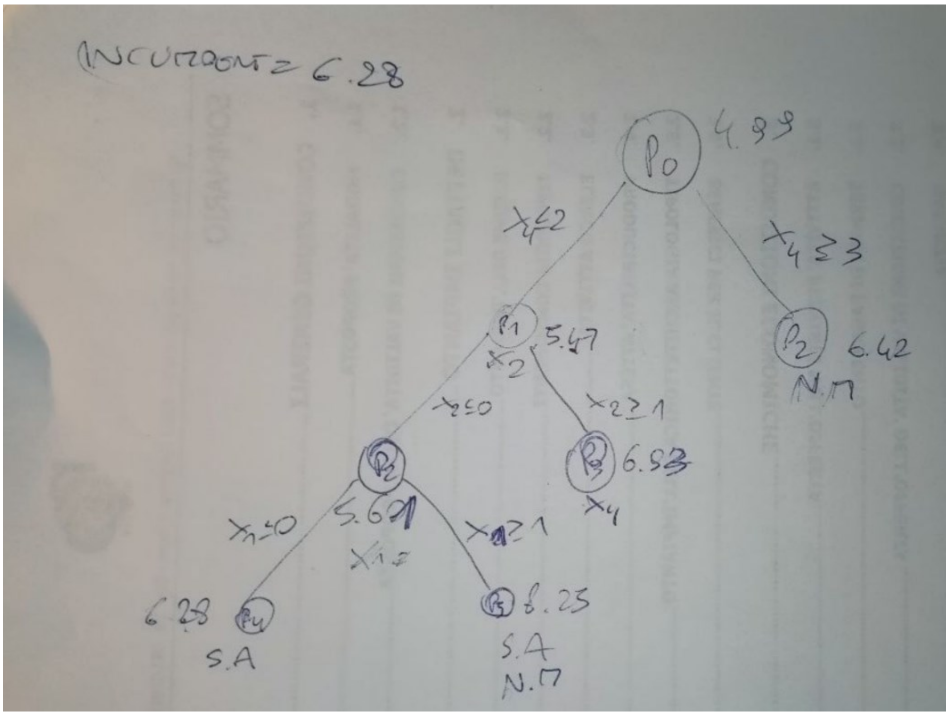
- Branching: binario
- Bound: rilassamento continuo (usare AMPL!)
- Fathoming: standard
- Esplorazione: a piacere (Best Bound First)
- Valutazione soluzioni ammissibili: nessuno (da rilassamento intero)
- Stop: lista nodi aperti vuota

File *.mod*:

```
var x1 >=0;
var x2 >=0;
var x3 >=0;
var x4 >=0;
var x5 >=0;
minimize f: 1.97*x1 + 3*x2 + 5*x3 + 2.14*x4 + 2*x5;
s.t. v1: -x1 + 3*x2 + x3 + 2*x4 + x5 >= 4;
s.t. v2: 2*x1 + 1.5*x2 + 2*x3 + 3*x4 + x5 >= 7;
```

File *.run*:

```
reset;
model es1.mod;
solve;
display f;
display x1,x2,x3,x4,x5;
```



Esercizio 2

Si consideri il problema "Assunzione multiperiodale di personale" e il modello formulato nelle note "Modelli di Programmazione Lineare". Si implementi il modello in AMPL e lo si risolva, per il caso descritto nel testo, con il metodo del Branch-and-Bound, assumento di avere a disposizione soltanto un solver per programmazione lineare a variabili continue.

`assunzionemulti.mod, assunzionemulti.dat`

Per non fare riferimenti a vuoto, inseriamo il relativo modello (pagg. 30/31 dispense "Note di programmazione lineare"):

Variabili decisionali

x_i: neoassunti nel mese i, $\forall\, i \in \{1,2,3,4,5\}$;

y_i: esperti disponibili nel mese i, $\forall\, i \in \{1,2,3,4,5\}$;

w_i: variabile logica legata alla scelta di assumere nel mese i, $\forall\, i \in \{1,2,3,4,5\}$:

$$w_i = \begin{cases} 1 & \text{se si decide di assumere nel mese } i \\ 0 & \text{altrimenti;} \end{cases}$$

z: variabile logica legata alla scelta di ottenere o meno il contributo statale:

$$z = \begin{cases} 1 & \text{se si decide di ottenere il contributo statale} \\ 0 & \text{altrimenti.} \end{cases}$$

Parametri

M: costante sufficientemente elevata (maggiore del numero massimo di apprendist assumibili nei mesi 3, 4 o 5).

Modello PLI

$$\min \quad 500(x_1+x_2+x_3+x_4+x_5)+1000(y_1+y_2+y_3+y_4+y_5)-10000z$$

s.t.

(Mese 1) $y_1 = 20$
$x_1 \leq y_1$
$150(y_1 - x_1) + 70x_1 \geq 2000$

(Mese 2) $y_2 = y_1 + x_1$
$x_2 \leq y_2$
$150(y_2 - x_2) + 70x_2 \geq 4000$

(Mese 3) $y_3 = y_2 + x_2$
$x_3 \leq y_3$
$150(y_3 - x_3) + 70x_3 \geq 7000$

(Mese 4) $y_4 = y_3 + x_3$
$x_4 \leq y_4$
$150(y_4 - x_4) + 70x_4 \geq 3000$

(Mese 5) $y_5 = y_4 + x_4$
$x_5 \leq y_5$
$150(y_5 - x_5) + 70x_5 \geq 3500$

(attiva z) $x_1 + x_2 \geq 10z$
(attiva w) $x_i \leq Mw_i, \forall\, i = 3,4,5$
(limiti) $w_3 + w_4 + w_5 \leq 1$

$x_i, y_i \in \mathbb{Z}_+, \forall\, i \in \{1,2,3,4,5\}$

$z \in \{0,1\}$
$w_i \in \{0,1\}, \forall\, i \in \{3,4,5\}$

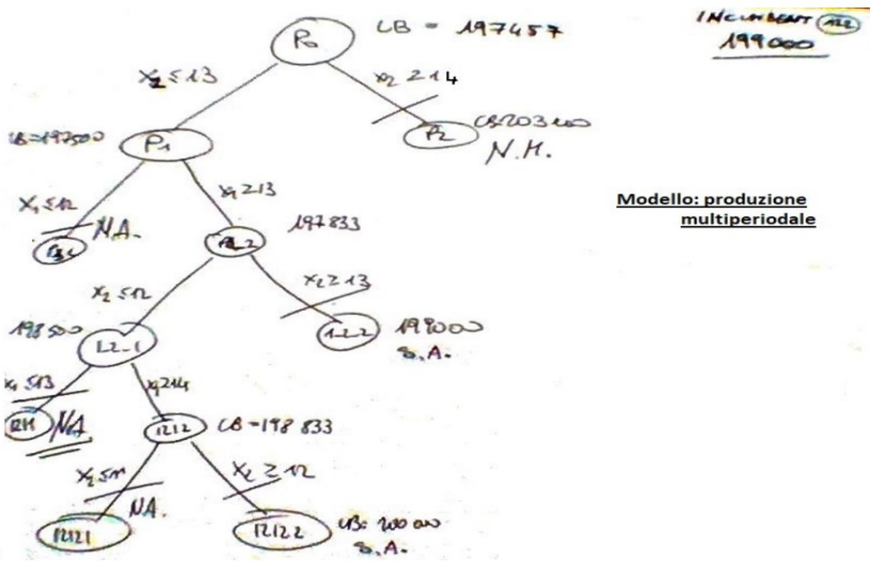
Per il file .mod:

```
### ASSUNZIONE MULTIPERIODALE ###
#################################
#INSIEMI
set mesi;
set mesi_limitati in mesi;
set mesi_iniziali = mesi diff mesi_limitati;
set mSet;
#PARAMETRI
param num_operai_init;
param num;
param den;
param costo_neoassunto := num/den;
param costo_esperto;
param incentivo;
param base_incentivo;
param capacita_operaio;
param capacita_istruttore;
param richiesta{mesi};
param bigM := (sum{i in mesi} richiesta[i]) / capacita_istruttore;
#VARIABILI
var X{mesi} integer >= 0;      #numero neoassunti
var Y{mesi} integer >= 0;      #numero eperti
var Z binary;                  #raccolgo incentivo
var W{mesi_limitati} binary;   #assumo nel mese
#FUNZIONE OBIETTIVO
minimize    costo_totale:
            costo_neoassunto * sum {i in mesi} X[i]
            + costo_esperto  * sum {i in mesi} Y[i]
            - incentivo      * Z;
#VINCOLI
s.t. operai_iniziali: Y[1] = num_operai_init;
s.t. bilancio_mensile {i in mesi: i > 1}: Y[i] = X[i-1] + Y[i-1];
s.t. sostieni_mensile {i in mesi}: X[i] <= Y[i];
s.t. domanda_mensile  {i in mesi}:
            capacita_operaio * (Y[i]-X[i]) + capacita_istruttore * X[i] >=
richiesta[i];
s.t. attiva_incentivo: sum {i in mesi_iniziali } X[i] >= base_incentivo * Z;
s.t. limite_assunzioni: sum {i in mesi_limitati} W[i] <= 1;
```

```
s.t. attiva_W {i in mesi_limitati}: X[i] <= bigM * W[i];
```

Per il file .dat:

```
### ASSUNZIONE MULTIPERIODALE ###
#################################
#INSIEMI
set mesi := 1 2 3 4 5;
set mesi_limitati := 3 4 5;
#PARAMETRI
param num_operai_init := 21;
param num := 1;
param den := 3;
#param costo_neoassunto := 500;
param costo_esperto := 1000;
param incentivo := 10000;
param base_incentivo := 10;
param capacita_operaio := 150;
param capacita_istruttore := 70;
param richiesta :=
1       2000
2       4000
3       7000
4       3000
5       3500
;
```



6. Si vuole risolvere con **AMPL** un problema di trasporto di alberi da un insieme di origini I a un insieme di destinazioni J. Ciascuna origine i mette a disposizione O_i alberi e ciascuna destinazione richiede D_j alberi. Il costo unitario di trasporto da i a j è C_{ij} e si ha un costo fisso F_i per l'organizzazione dei trasporti da ciascuna origine i. Non è inoltre possibile organizzare il trasporto in più di N origini. Il modello per la minimizzazione dei costi è riportato affianco e utilizza le variabili x_{ij} per indicare il numero di alberi trasportati da i a j, e y_i che vale 1 se si organizza il trasporto da i, 0 altrimenti.

$$
\begin{aligned}
\min \sum_{i \in I, j \in J} C_{ij}\, x_{ij} - \sum_{i \in I} F_i\, y_i \\
\text{s.t.} \sum_{i \in I} x_{ij} \geq D_j \quad , \quad \forall j \in J \\
\sum_{j \in J} x_{ij} \leq O_i\, y_i \quad , \quad \forall i \in I \\
\sum_{i \in I} y_i \leq N \\
x_{ij} \in \mathbb{Z}_+, \;\; y_i \in \{0,1\}, \;\; \forall\, i \in I, \;\; j \in J
\end{aligned}
$$

a. Si traduca nel linguaggio **AMPL** il modello proposto (**file .mod**).

b. Si produca il **file .dat** per l'istanza con origini Croazia, Svezia, Gran Bretagna e Canada (disponibilità di 1000, 2000, 3000 e 4000 alberi rispettivamente), destinazioni Italia, Francia e Germania (con richieste di 5000, 3000 e 2000 rispettivamente), $N = 3$, costi fissi F_i di 1000 euro per tutte le origini, e costi di trasporto verso Italia, Francia e Germania (nell'ordine) pari a: dalla Croazia 10, 20 e 30 euro; dalla Svezia 40, 50 e 60 euro; dalla Gran Bretagna 70, 80 e 90 euro; dal Canada 100, 110 e 120 euro.

c. Si scriva uno script di AMPL (**file .run**) che risolve l'istanza specificata e visualizza il valore della funzione obiettivo e delle variabili per una soluzione ottima.

Punto a)

```
set I;              set J;
param O{I};         param D{J};
param C{I,J};       param F{I};
param N;
var x{I,J} >=0 integer;
var y{I} binary;
minimize fo: sum{i in I, j in J} C[i,j]*x[i,j] - sum{i in I} F[i]*y[i];
s.t. d{j in J}: sum{i in I} x[i,j] >= D[j];
s.t. o{i in I}: sum{j in J} x[i,j] <= O[i] * y[i];
s.t. n: sum{i in I} y[i] <= N;
```

Punto b)

```
set I := Croazia Svezia GranBretagna Canada;
set J := Italia Francia Germania;
param :        F     O :=
Croazia        1000  1000
Svezia         1000  2000
GranBretagna   1000  3000
Canada         1000  4000;
param D := Italia 5000 Francia 3000 Germania 2000;
param N := 4;
param C :      Italia     Francia     Germania :=
Croazia        10         20          30
Svezia         40         50          60
GranBretagna   70         80          90
Canada         100        110         120;
```

Punto c)

```
reset;
model ampl.mod;
data ampl.dat;
option solver cplexamp;
solve;
display fo, x, y;
```

6. Si vuole risolvere con **AMPL** un problema di trasporto di alberi da un insieme di origini I a un insieme di destinazioni J. Ciascuna origine i mette a disposizione O_i alberi e ciascuna destinazione richiede D_j alberi. Il costo unitario di trasporto da i a j è C_{ij} e si ha un costo fisso F_i per l'organizzazione dei trasporti da ciascuna origine i. Non è inoltre possibile organizzare il trasporto in più di N origini. Il modello per la minimizzazione dei costi è riportato affianco e utilizza le variabili x_{ij} per indicare il numero di alberi trasportati da i a j, e y_i che vale 1 se si organizza il trasporto da i, 0 altrimenti.

$$\min \sum_{i\in I, j\in J} C_{ij}\, x_{ij} - \sum_{i\in I} F_i\, y_i$$

$$\text{s.t.} \sum_{i\in I} x_{ij} \geq D_j \quad , \quad \forall j \in J$$

$$\sum_{j\in J} x_{ij} \leq O_i\, y_i \quad , \quad \forall i \in I$$

$$\sum_{i\in I} y_i \leq N$$

$$x_{ij} \in \mathbb{Z}_+,\ \ y_i \in \{0,1\},\ \ \forall i \in I,\ \ j \in J$$

a. Si traduca nel linguaggio **AMPL** il modello proposto (**file .mod**).

b. Si produca il **file .dat** per l'istanza con origini Croazia, Svezia, Gran Bretagna e Canada (disponibilità di 1000, 2000, 3000 e 4000 alberi rispettivamente), destinazioni Italia, Francia e Germania (con richieste di 5000, 3000 e 2000 rispettivamente), $N = 3$, costi fissi F_i di 1000 euro per tutte le origini, e costi di trasporto verso Italia, Francia e Germania (nell'ordine) pari a: dalla Croazia 10, 20 e 30 euro; dalla Svezia 40, 50 e 60 euro; dalla Gran Bretagna 70, 80 e 90 euro; dal Canada 100, 110 e 120 euro.

c. Si scriva uno script di AMPL (**file .run**) che risolve l'istanza specificata e visualizza il valore della funzione obiettivo e delle variabili per una soluzione ottima.

Punto a)

```
set I;                  set J;
param O{I};             param D{J};
param C{I,J};           param F{I};
param N;
var x{I,J} >=0 integer;
var y{I} binary;
minimize fo: sum{i in I, j in J} C[i,j]*x[i,j] - sum{i in I} F[i]*y[i];
s.t. d{j in J}: sum{i in I} x[i,j] >= D[j];
s.t. o{i in I}: sum{j in J} x[i,j] <= O[i] * y[i];
s.t. n: sum{i in I} y[i] <= N;
```

Punto b)

```
set I := Croazia Svezia GranBretagna Canada;
set J := Italia Francia Germania;
param :         F     O :=
Croazia         1000  1000
Svezia          1000  2000
GranBretagna    1000  3000
Canada          1000  4000;
param D := Italia 5000 Francia 3000 Germania 2000;
param N := 4;
param C :       Italia      Francia     Germania :=
Croazia         10          20          30
Svezia          40          50          60
GranBretagna    70          80          90
Canada          100         110         120;
```

Punto c)

```
reset;
model ampl.mod;
data ampl.dat;
option solver cplexamp;
solve;
display fo, x, y;
```

5. Si traduca nel linguaggio AMPL il seguente modello di programmazione lineare intera, relativo a un problema di produzione:

$$\min \sum_{i \in I} c_i x_i$$

$$\text{s.t.} \quad \sum_{i \in I} a_{ij} x_i \geq b_j \quad , \quad \forall j \in J$$

$$x_i \in \mathbb{Z}_+ \, , \quad \forall j \in J$$

dove I è l'insieme delle materie prime, J è l'insieme dei prodotti. I dati sono: c_i il costo unitario della materia prima $i \in I$, b_j la richiesta minima del prodotto $j \in J$, a_{ij} la quantità di prodotto j estraibile da un'unità di materia prima i. Le variabili sono: x_i la quantità di materia prima i impiegata (si possono impiegare solo quantità intere).

Viene quindi richiesto di scrivere un modello come segue:

```
set I; #materie prime
set J; #prodotti
param C{I}; #costo unitario materia prima
param B{J}; #richiesta minima prodotto
param A{I, J}; #quantità di prodotto estraibile da un'unità di materia prima

var x{I} >=0 integer;

minimize f: sum{i in I} c[i] * x[i];
s.t. v1{j in J}: sum{i in I} a[i,j]*x[i] >= b[j];
```

Varie:

- **Variabili decisionali:**
  - x_L : quantità in ettari da destinare a lattuga
  - x_P : quantità in ettari da destinare a patate
- **Funzione obiettivo:**
  - $max\ 3000\, x_L + 5000\, x_P$ (ricavo totale da massimizzare)
- **Sistema dei vincoli:**
  - $x_L + x_P \leq 11$ (ettari disponibili)
  - $7\, x_L \leq 70$ (semi disponibili)
  - $3\, x_P \leq 18$ (tuberi disponibili)
  - $10\, x_L + 20\, x_P \leq 145$ (fertilizzante disponibile)
  - $x_L \geq 0, x_P \geq 0$ (dominio)

```
var xL; #ettari da coltivare a lattuga
var xP; #ettari patate

maximize resa: 3000*xL+5000*xP; #f.o.

subject to ettari: xL+xP <= 11;
subject to semi: 7*xL <= 70;
s.t. tuberi: 3*xP<=18;
s.t. concime: 10*xL+20*xP <= 145;
s.t. domL: xL >= 0;
s.t. domP: xP >= 0;
```

Modelli di mix ottimo di produzione

$$\max \sum_{i \in I} P_i x_i$$

$$s.t. \quad \sum_{i \in I} A_{ij} x_i \leq Q_j \qquad \forall j \in J$$

$$x_i \in \mathbb{R}_+ \,[\, \mathbb{Z}_+ \mid \{0,1\} \,] \quad \forall i \in I$$

I insieme dei beni che possono essere prodotti;

J insieme delle risorse disponibili;

P_i profitto (unitario) per il bene $i \in I$;

Q_j quantità disponibile della risorsa $j \in J$;

A_{ij} quantità di risorsa j necessaria per la produzione di un'unità del bene i.

Modello in AMPL: sintassi base

Espressioni indicizzanti

```
#DICHIARAZIONE INSIEMI
set Prodotti;
set Risorse;
#DICHIARAZIONE PARAMETRI
param maxNumProd;      # massimo numero prodotti
param P {Prodotti};    # profitto unitario
param Q {Risorse};     # disponibilità risorsa
param A {Prodotti,Risorse};
                       # risorsa per unità di pr.
var x {Prodotti} >=0 , <= maxNumProd;
maximize profitto: sum {i in Prodotti} P[i]*x[i];
subject to disponib {j in Risorse}:
        sum {i in Prodotti} A[i,j]*x[i] <= Q[j];
```

1. **Un gioco di assemblaggio**

Per l'assemblaggio di telecomandi, si hanno a disposizione 10 moduli display, 18 moduli di logica di controllo, 12 trasmettitori, 21 tastierini, 9 moduli di navigazione e 10 led. I telecomandi sono di due tipi. Il tipo A richiede un display, un modulo di navigazione, 2 tastierini, 2 moduli di logica, un trasmettitore e un led. Il tipo B richiede 2 display, 3 tastierini, 2 moduli di logica e 3 trasmettitori. Considerando che il tipo A permette un guadagno netto di 3 euro e il tipo B di 8 euro, determinare la produzione che massimizza il guadagno.

$$
\begin{array}{llllll}
\max & 3x_A & + & 8x_B & & & \text{(guadagno complessivo)} \\
s.t. & & & & & & \\
& x_A & + & 2x_B & \leq & 10 & \text{(display)} \\
& x_A & & & \leq & 9 & \text{(navigazione)} \\
& 2x_A & + & 3x_B & \leq & 21 & \text{(tastierini)} \\
& 2x_A & + & 2x_B & \leq & 18 & \text{(logica)} \\
& x_A & + & 3x_B & \leq & 12 & \text{(trasmissione)} \\
& x_A & & & \leq & 10 & \text{(led)} \\
& & & & & & \\
& x_A & , & x_B & \in & \mathbb{Z}_+ & \text{(dominio)}
\end{array}
$$

```
set I; #prodotti
set J; #risorse

param maxNumProd;
param P{I};
param Q[j];
param A{I,J};

var x {i in I} >= 0 <= maxNumProd[i] integer;

maximize profitto: sum{i in I} P[i]*x[i];
s.t. dispon{j in J}: sum{i in I} A[i, j]*x[i]
```

```
#insiemi

set I := telA telB;
set J := modisplay navigazione tastierini logica tx led;

#parametri

param P := telA 3 telB 6;

param maxNumProd := telA 5 telB 8;

param        Q  :=
modisplay    10
navigazione   9
tastierini   21
logica       18
tx           12
led          10
;

param A:    modisplay  navigazione  tastierini  logica    tx  led :=
telA        1          1            2           2         1   1
telB        2          0            3           2         3   0
```

Un dietologo deve preparare una dieta che garantisca un apporto giornaliero di proteine, ferro e calcio di almeno 20 mg, 30 mg e 10 mg, rispettivamente. Il dietologo è orientato su cibi a base di verdura (5 mg/kg di proteine, 6 mg/Kg di ferro e 5 mg/Kg di calcio, al costo di 4 €/Kg), carne (15 mg/kg di proteine, 10 mg/Kg di ferro e 3 mg/Kg di calcio, al costo di 10 €/Kg) e frutta (4 mg/kg di proteine, 5 mg/Kg di ferro e 12 mg/Kg di calcio, al costo di 7 €/Kg). Determinare la dieta di costo minimo.

Risolvere il problema con AMPL (file .mod e .dat separati)

Modello generale: dieta

I insieme delle risorse disponibili;

J insieme delle domande da coprire;

C_i costo (unitario) per l'utilizzo della risorsa $i \in I$;

D_j ammontare della domanda di $j \in J$;

A_{ij} capacità (unitaria) della risorsa i di soddisfare la domanda j.

$$
\begin{array}{ll}
\min & \sum_{i \in I} C_i x_i \\
s.t. & \\
& \sum_{i \in I} A_{ij} x_i \geq D_j \qquad \forall\, j \in J \\
& x_i \in \mathbb{R}_+ \ [\, \mathbb{Z}_+ \mid \{0,1\}\,] \quad \forall\, i \in I
\end{array}
$$

Il file *dieta.mod* contiene come codice:

```
set I; #risorse
set J; #domande

param C{I}; #costo risorse
param D[J]; #domande
param A{I,J}; #capacità risorsa "i" per
soddisfare la domanda "j"

var x{I} >= 0 integer;

minimize costo: sum{i in I} C[i]*x[i];

s.t. soddisfazione{j in J}: sum{i in I}
A[i,j] * x[i] >= D[j];
```

Modello PL

- Siano x_1, x_2 e x_3 le quantità di cibi a base di verdura, carne e frutta, rispettivamente

min $4\,x_1+10\,x_2+7\,x_3$ (costo giornaliero dieta)

s.t.

$5x_1+15x_2+4x_3 \geq 20$ (proteine)

$6x_1+10x_2+5x_3 \geq 30$ (ferro)

$5x_1+3x_2+12x_3 \geq 10$ (calcio)

$x_i \in \mathbb{R}_+, \ \forall i \in \{1, 2, 3\}$

Il file *dieta.dat* contiene come codice:

```
#insiemi
set I := 1ver 2car 3fru; #insieme delle risorse
set J := pro fer cal; #insieme delle domande

#parametri

#costi indicizzati dall'indice "i"
param C :=  1ver 4      2car 10     3fru 7;

#domande indicizzate dall'indice "j"
param D :=  pro 20      fer 30      cal 10;

#tabella con colonne indicizzate da "i"
#e righe indicizzate da "j"
#e quindi si scrive trasposta con (tr)

param A (tr) :          1ver        2car        3fru :=
pro                     5           15          4
fer                     6           10          5
cal                     5           3           12
;
```

In merito al file *diet.run* contiene:

```
reset;
model dieta.mod;
data dieta.dat;

option solver cplex;
solve;

display costo, x;
```

Esercizio 5.

Il dietologo vuole inserire alimenti a base di pesce azzurro
(10 mg/kg di proteine, 15 mg/kg di ferro e 2 mg/kg di calcio,
al costo di 3 euro/kg) nella dieta.
Modificare opportunamente i file relativi al problema.

Quindi, listiamo tutti i file, prendendo il file *diet2.mod*:

```
set I; #risorse
set J; #domande

param C{I}; #costo risorse
param D{J}; #domande
param A{I,J};
param MinRisorsa{I} default 0; #parametro indicizzato nell'insieme I
#messo a default a 0 (perché una, la risorsa minima, ha valore 3
#mentre tutte le altre a 0)

var x{i in I} integer >= MinRisorsa[i];

minimize costo: sum{i in I} C[i]*x[i];
s.t. soddisfazione{j in J}: sum{i in I} A[i,j] * x[i] >= D[j];
```

Segue il file *diet2.dat*:

```
#insiemi
set I := 1ver 2car 3fru 4azz; #insieme delle risorse
set J := pro fer cal; #insieme delle domande

#parametri

#costi indicizzati dall'indice “i”
param C :=  1ver 4      2car 10     3fru 7      4azz 3;

#domande indicizzate dall'indice “j”
param D :=  pro 20      fer 30      cal 10;

param MinRisorsa := #solo per il pesce azzurro ho un .bound di almeno 3
4azz 3;

#tabella con colonne indicizzate da "i"
#e righe indicizzate da "j"
#e quindi si scrive trasposta con (tr)

param A (tr) :          1ver        2car        3fru        4azz :=
pro                     5           15          4           10
fer                     6           10          5           15
cal                     5           3           12          2
;
```

Esercizio 6. Indagine di mercato

Un'azienda pubblicitaria deve svolgere un'indagine di mercato per lanciare un nuovo prodotto. Si deve contattare telefonicamente un campione significativo di persone: almeno 150 donne sposate, almeno 110 donne non sposate, almeno 120 uomini sposati e almeno 100 uomini non sposati. Le telefonate possono essere effettuate al mattino (al costo operativo di 1.1 euro) o alla sera (al costo di 1.6 euro). Le percentuali di persone mediamente raggiunte sono riportate in tabella.

	Mattino	Sera
Donne sposate	30%	30%
Donne non sposate	10%	20%
Uomini sposati	10%	30%
Uomini non sposati	10%	15%
Nessuno	40%	5%

Si noti come le telefonate serali sono più costose, ma permettono di raggiungere un maggior numero di persone: solo il 5% va a vuoto. Si vuole minimizzare il costo complessivo delle telefonate da effettuare (mattina/sera) in modo da raggiungere un campione significativo di persone

Risolvere il problema con AMPL (usare soluzione Es. 4)

Modello PLI

- Siano x_1 e x_2 il numero di telefonate da fare al mattino e alla sera, rispettivamente

min $\quad 1.1\,x_1 + 1.6\,x_2$ (costo totale telefonate)

s.t.

$0.3x_1 + 0.3x_2 \geq 150$ (donne sposate)

$0.1x_1 + 0.2x_2 \geq 110$ (donne non sposate)

$0.1x_1 + 0.3x_2 \geq 120$ (uomini sposati)

$0.1x_1 + 0.15x_2 \geq 100$ (uomini non sposati)

$x_i \in \mathbb{Z}_+, \ \forall i \in \{1, 2\}$

[Risorse] `diet.indagine.dat`

Modello generale: dieta

I insieme delle risorse disponibili;

J insieme delle domande da coprire;

C_i costo (unitario) per l'utilizzo della risorsa $i \in I$;

D_j ammontare della domanda di $j \in J$;

A_{ij} capacità (unitaria) della risorsa i di soddisfare la domanda j.

$$\begin{aligned} \min \quad & \sum_{i \in I} C_i x_i \\ s.t. \quad & \\ & \sum_{i \in I} A_{ij} x_i \geq D_j \qquad \forall j \in J \\ & x_i \in \mathbb{R}_+ \,[\, \mathbb{Z}_+ \mid \{0,1\} \,] \quad \forall i \in I \end{aligned}$$

Nel file *indagine.dat* si dettaglia la seguente struttura per i singoli dati:

```
#insiemi
set I := 1mattino 2sera; #risorse: telefonate al mattino e alla sera
set J := ds dn us uc; #domande: uomini/donne sposati-e/non sposati-e

#parametri

param C := 1mattino 1.1 2sera 1.6; #costo mattino-sera
param D := ds 150 dn 110 us 120 uc 100; #domande uomini/donne sp/nsp

param A (tr):         1mattino          2sera := #tab. come da slide
ds                    0.3               0.3
dn                    0.1               0.2
us                    0.1               0.3
uc                    0.1               0.15
```

Esempio: Localizzazione di servizi

Una città è divisa in sei quartieri, dove si vogliono attivare dei centri unificati di prenotazione (CUP) per servizi sanitari. In ciascun quartiere è stata individuata una possibile località di apertura. Le distanze medie in minuti da ciascun quartiere a ciascuna delle possibili località è indicata in tabella. Si desidera che nessun utente abbia un tempo medio di spostamento superiore a 15 minuti per arrivare al CUP più vicino e si vuole minimizzare il numero di CUP attivati.

	Loc. 1	Loc. 2	Loc 3	Loc. 4	Loc. 5	Loc. 6
Q.re 1	5	10	20	30	30	20
Q.re 2	10	5	25	35	20	10
Q.re 3	20	25	5	15	30	20
Q.re 4	30	35	15	5	15	25
Q.re 5	30	20	30	15	5	14
Q.re 6	20	10	20	25	14	5

Sia $x_i = 1$, se viene aperto il CUP nel quartiere *i*, 0 altrimenti

$$
\begin{aligned}
\min \quad & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\
s.t. \quad & \\
& x_1 + x_2 \geq 1 \text{ (esigenze q.re 1)} \\
& x_1 + x_2 + x_6 \geq 1 \text{ (esigenze q.re 2)} \\
& x_3 + x_4 \geq 1 \text{ (esigenze q.re 3)} \\
& x_3 + x_4 + x_5 \geq 1 \text{ (esigenze q.re 4)} \\
& x_4 + x_5 + x_6 \geq 1 \text{ (esigenze q.re 5)} \\
& x_2 + x_5 + x_6 \geq 1 \text{ (esigenze q.re 6)} \\
& x_i \in \{0,1\}
\end{aligned}
$$

- Come modello di copertura («dieta»): richiede data preprocessing ([Risorse]: `diet.mod` – `diet.CUP.dat`)
- Modello specifico a partire dai dati «grezzi»:

I : insieme dei quartieri

c_{ij} : distanza da $i \in I$ a $j \in I$

T : distanza soglia

$$\min \sum_{i \in I} x_i$$

$$\text{s.t.} \sum_{j \in I : c_{ij} \leq T} x_j \geq 1, \quad \forall i \in I$$

$$x_i \in \{0,1\}, \forall i \in I$$

```
#insiemi
set I := loc1 loc2 loc3 loc4 loc5 loc6;
set J := qre1 qre2 qre3 qre4 qre5 qre6;

#parametri
param C := loc1 1 loc2 1 loc3 1 loc4 1 loc5 1 loc6 1;

param D := qre1 1 qre2 1 qre3 1 qre4 1 qre5 1 qre6 1;

param A (tr):   loc1    loc2    loc3    loc4    loc5    loc6 :=
qre1            1       1       0       0       0       0
qre2            1       1       0       0       0       1
qre3            0       0       1       1       0       0
qre4            0       0       1       1       0       0
qre5            0       0       0       1       1       0
qre6            0       1       0       0       1       1
;
```

```
set I; #quartieri
param C{I,I}; #costi
param T; #soglia
var x{I} binary;
minimize costo: sum {i in I} x{i};
subject to soddisfa{i in I}:
      sum{j in I : C[i, j] <= T} x[j] >= 1;
```

Andiamo a creare i dati dentro *cuptempi.dat*:

```
set I := q1 q2 q3 q4 q5 q6; #quartieri
param C :           q1    q2    q3    q4    q5    q6  :=
q1                  5     10    20    30    30    20
q2                  10    5     25    35    20    10
q3                  20    25    5     15    30    20
q4                  30    35    15    5     15    25
q5                  30    20    30    15    5     14
q6                  20    10    20    25    14    5
;

param T := 15; #soglia
```

Qui poi il relativo *cuptempi.run*:

```
set I := q1 q2 q3 q4 q5 q6; #quartieri
param C :           q1    q2    q3    q4    q5    q6  :=
q1                  5     10    20    30    30    20
q2                  10    5     25    35    20    10
q3                  20    25    5     15    30    20
q4                  30    35    15    5     15    25
q5                  30    20    30    15    5     14
q6                  20    10    20    25    14    5
;

param T := 15; #soglia
```

Una ditta di produzione di elettrodomestici produce dei frigoriferi in tre stabilimenti e li smista in quattro magazzini intermedi di vendita. La produzione settimanale nei tre stabilimenti A, B e C è rispettivamente di 50, 70 e 20 unità. La quantità richiesta dai 4 magazzini è rispettivamente di 10, 60, 30 e 40 unità. I costi per il trasporto di un frigorifero tra gli stabilimenti e i magazzini 1, 2, 3 e 4 sono i seguenti:

- dallo stabilimento A: 6, 8, 3, 4 euro;
- dallo stabilimento B: 2, 3, 1, 3 euro;
- dallo stabilimento C: 2, 4, 6, 5 euro.

Utilizzare AMPL per determinare il piano di trasporti di costo minimo, considerando che non sono ammesse rimanenze alla fine della settimana e che lo stesso modello dovrà essere utilizzato per diverse settimane.

Modello generale: trasporti

I insieme dei centri di offerta; O_i ammontare dell'offerta in $i \in I$;

J insieme dei centri di domanda; D_j ammontare della domanda in $j \in J$.

C_{ij} costo (unitario) per il trasporto da $i \in I$ a $j \in J$;

$$\begin{aligned}
\min \quad & \sum_{i\in I}\sum_{j\in J} C_{ij}x_{ij} \\
s.t. \quad & \\
& \sum_{j\in J} x_{ij} \le O_i && \forall\, i \in I \\
& \sum_{i\in I} x_{ij} \ge D_j && \forall\, j \in J \\
& x_{ij} \in \mathbb{R}_+ \ [\ \mathbb{Z}_+ \mid \{0,1\}\ ] && \forall\, i \in I, j \in J
\end{aligned}$$

- Sia x_{ij} il numero di frigoriferi prodotti nello stabilimento i e smistati nel magazzino j

$$\begin{aligned}
\min \quad & 6\,x_{A1}+8\,x_{A2}+3\,x_{A3}+4\,x_{A4}+ \\
+ \quad & 2\,x_{B1}+3\,x_{B2}+1\,x_{B3}+3\,x_{B4}+ \\
+ \quad & 2\,x_{C1}+4\,x_{C2}+6\,x_{C3}+5\,x_{C4} \\
s.t. \quad & \\
& x_{A1}+x_{A2}+x_{A3}+x_{A4} \le 50 && \text{(capacità produttiva stabilimento A)} \\
& x_{B1}+x_{B2}+x_{B3}+x_{B4} \le 70 && \text{(capacità produttiva stabilimento B)} \\
& x_{C1}+x_{C2}+x_{C3}+x_{C4} \le 20 && \text{(capacità produttiva stabilimento C)} \\
& x_{A1}+x_{B1}+x_{C1} \ge 10 && \text{(domanda magazzino 1)} \\
& x_{A2}+x_{B2}+x_{C2} \ge 60 && \text{(domanda magazzino 2)} \\
& x_{A3}+x_{B3}+x_{C3} \ge 30 && \text{(domanda magazzino 3)} \\
& x_{A4}+x_{B4}+x_{C4} \ge 40 && \text{(domanda magazzino 4)} \\
& x_{ij} \in \mathbb{Z}_+ \ \forall i \in \{A, B, C\}, j \in \{1, 2, 3, 4\}
\end{aligned}$$

Il modello generale dei trasporti viene implementato come segue su *trasporto.mod*:

```
set I; #insieme offerta
set J; #insieme domanda

param O{I}; #offerta
param D{J}; #domanda
param C{I, J}; #costo trasporto

var x{I, J} >=0 integer;

minimize costo_totale:
      sum{i in I, j in J} C[i,j] * x [i, j];

s.t. origine{i in I}: sum{j in J} x[i, j]<=O[i];

s.t. destinazione{j in J}: sum{i in I} x[i,j] >= D[j];
```

I dati del modello dei frigoriferi si trovano nel file come segue *traspFrigo.dat*:

```
set I := A B C;
set J := 1 2 3 4;

param O := A 50 B 70 C 20;

param D := 1 10   2 60 3 30 4 40;

param C :         1     2     3     4   :=
A                 6     8     3     4
B                 2     3     1     3
C                 2     4     6     5
;
```

Poi, il successivo file per farlo eseguire, *traspFrigo.run*:

```
reset;
model trasporto.mod;
data traspFrigo.dat;
option solver cplex;
solve;
display costo_totale, x;
```

Un'azienda assembla dei PC in tre diversi stabilimenti con diverso costo unitario di produzione. I PC sono venduti a cinque clienti bancari e si sopportano dei costi di trasporto (inclusi gli oneri di importazione) per spedire un PC da ciascuno stabilimento a ciascun cliente. Sono definite le richieste di PC di ogni cliente e la produzione di ciascuno stabilimento è limitata. Non sono ammessi eccessi di produzione. I dati sono riassunti nella tabella seguente.

Scrivere in AMPL un modello del problema e fornire la soluzione, in termini di costo complessivo di trasporto e di quantità trasportate tra stabilimenti e sedi bancarie.

Produzione			Costi di trasporto				
Unità	**costo unit.**	**Capa-cità**	**Banca Intesa**	**Uni Credit**	**Anton Veneta**	**Credit Suisse**	**Banca Cina**
Italia	220	10000	5,5	7,5	6,9	8,0	10,3
Cina	180	20000	15,0	14,3	13,0	16,4	5,0
Francia	200	10000	6,0	7,8	6,3	6,8	11,0
		Domanda	7100	3400	9700	5200	3050

Insiemi: *S* (stabilimenti) e *B* (banche)
Parametri: w_i (costi prod.), c_{ij} (costi trasp.), a_i (capacità prod.), b_j (richieste)

$$\min \sum_{i \in S, j \in B} (w_i + c_{ij})\, x_{ij}$$

$$\text{s.t.} \sum_{j \in B} x_{ij} \leq a_i\,, \qquad \forall\, i \in S$$

$$\sum_{i \in S} x_{ij} = b_j\,, \qquad \forall\, j \in B \qquad \text{(no eccessi di produzione} \Rightarrow \text{uguaglianza)}$$

$$x_{ij} \in Z_+ \qquad \forall\, i \in S\,, j \in B$$

Il modello viene considerato nel file *pc.mod*:

```
set I; #stabilimenti - S
set J; #banche - B

param A{I}; #capacità produttive
param B{J}; #richieste
param W{I}; #costi produttivi
param C{I, J}; #costi
var x{I, J} integer >= 0;

minimize costo: sum{i in I, j in J}
(W[i] + C[i,j])*x[i,j];
s.t. produzione{i in I}: sum{j in J} x[i,j] <= A[i];
s.t. domanda{j in J}: sum{i in I} x[i,j] = B[j];
```

I dati invece seguono nel file *pc.dat*:

```
set I := it ch fr;
set J := bi uc av cs bc;

param A := it 10000 ch 20000 fr 10000;
param B := bi 7100 uc 3400 av 9700 cs 5200 bc 3050;

param W := it 220 ch 180 fr 200;
```

```
#param :    w      a :=
#it                220   10000
#ch                100   20000
#fr                200   10000

param C:           bi           uc           av           cs           bc :=
it                 5.5          7.5          6.9          8.0          10.3
ch                 15           14.3         13.0         16.4         5.0
fr                 6.0          7.8          6.3          6.8          11.0
;
```

Il file *pc.run* segue qui:

```
reset;
model pc.mod;
data pc.dat;
option solver cplex;
solve;
display costo, x;
```