

4

Il linguaggio di modellizzazione algebrica AMPL

In questo capitolo sono riportate le implementazioni in AMPL di alcuni esercizi dei quali è già stata fornita la formulazione algebrica nel capitolo precedente.

4.1 ESERCIZI DI FORMULAZIONI MEDIANTE IL LINGUAGGIO AMPL

Esercizio 4.1.1 *Si traduca in AMPL (producendo un file .mod e uno .dat) il modello di Programmazione Lineare definito nell'Esercizio 3.1.1.*

Soluzione.

```
industria.mod

set PROD;
set MACCH;

param Profitto{PROD};
param Tempo_macch{PROD,MACCH};
param Tempo_oper;
param Numero_macch{MACCH};
param Numero_oper;
param Ore_max_macch;
param Ore_max_oper;

var Quantita{PROD} >= 0;

maximize profitto_tot: sum{p in PROD} Profitto[p]*Quantita[p];
```

```

subject to limite_macchine {m in MACCH}:
    sum{p in PROD} Tempo_macch[p,m]*Quantita[p] <=
        Numero_macch[m]*Ore_max_macch;

subject to limite_oper:
    Tempo_oper*sum{p in PROD} Quantita[p] <=
        Numero_oper*Ore_max_oper;

```

industria.dat

```

set PROD := P1 P2 P3 P4 P5;
set MACCH := M1 M2;

param: Profitto :=
P1      250
P2      300
P3      500
P4      450
P5      180;

param Tempo_macch:  M1  M2 :=
P1                10  9
P2                15 13
P3                 7  0
P4                18  0
P5                 0 20;

param Tempo_oper := 18;

param: Numero_macch :=
M1      4
M2      3;

param Numero_oper := 10;
param Ore_max_macch := 80; # giorni*turni*ore
param Ore_max_oper := 40; # giorni*turni*ore

```

Esercizio 4.1.2 Si traduca in AMPL (producendo un file *.mod* e uno *.dat*) il modello di Programmazione Lineare definito nell'Esercizio 3.1.2.

Soluzione.

```
elettrodomestici.mod
set ELET;
set REP;

param NumOper{REP};
param OreOper;
param OreProd{ELET,REP};
param Prezzo{ELET};
param Domanda{ELET};

var Qta{ELET,REP} >= 0, integer;

maximize profitto: sum{e in ELET, r in REP} Prezzo[e]*Qta[e,r];

subject to produzione {r in REP}:
    sum{e in ELET} OreProd[e,r]*Qta[e,r] <= NumOper[r]*OreOper;

subject to domanda_min {e in ELET}:
    sum{r in REP} Qta[e,r] >= Domanda[e];
```

```
elettrodomestici.dat
set ELET := E1 E2 E3 E4;
set REP := R1 R2 R3;

param: NumOper :=
R1      40
R2      35
R3      25;

param OreOper := 40;

param OreProd:  R1  R2  R3 :=
E1              1   1.2 0.8
E2              1.5 1.3 1.7
E3              0.5 0.6 0.7
```

```
E4          1.6 1    1.3;
```

```
param: Prezzo Domanda :=
```

```
E1      800    1000
```

```
E2     1200     600
```

```
E3      950     300
```

```
E4     1100     200;
```

Esercizio 4.1.3 Si traduca in AMPL (producendo un file *.mod* e uno *.dat*) il modello di Programmazione Lineare definito nell'Esercizio 3.2.1.

Soluzione.

raffineria.mod

```
set BENZ;
```

```
set CARB;
```

```
param Ottani{BENZ};
```

```
param Ettol_disp{BENZ};
```

```
param Costo{BENZ};
```

```
param Prezzo_benz{BENZ};
```

```
param Min_ottani{CARB};
```

```
param Prezzo_carb{CARB};
```

```
param Dom_min{CARB};
```

```
param Dom_max{CARB};
```

```
var Qta{BENZ,CARB} >= 0;
```

```
var Qta_grez{BENZ} >= 0;
```

```
maximize ricavo:
```

```
(sum{b in BENZ, c in CARB} (Prezzo_carb[c]-Costo[b])*Qta[b,c]) +
(sum{b in BENZ} (Prezzo_benz[b]-Costo[b])*Qta_grez[b]);
```

```
subject to qualita {c in CARB}:
```

```
sum{b in BENZ} (Ottani[b]-Min_ottani[c])*Qta[b,c] >= 0;
```

```
subject to disponibilita {b in BENZ}:
```

```
sum{c in CARB} Qta[b,c] + Qta_grez[b] <= Ettol_disp[b];
```

```
subject to domanda {c in CARB}:
```

```
Dom_min[c] <= sum{b in BENZ} Qta[b,c] <= Dom_max[c];
```

```
raffineria.dat
```

```
set BENZ := B1 B2 B3 B4;
```

```
set CARB := C1 C2;
```

```
param: Ottani Ettol_disp Costo Prezzo_benz :=
```

```
B1      90      3500      260      280
```

```
B2      73      6000      210      250
```

```
B3      79      4500      190      250
```

```
B4      86      5200      220      280;
```

```
param: Min_ottani Prezzo_carb Dom_min Dom_max :=
```

```
C1      80      350      0      25000
```

```
C2      85      520      10000      Infinity;
```

Esercizio 4.1.4 Si traduca in AMPL (producendo un file *.mod* e uno *.dat*) il modello di Programmazione Lineare definito nell'Esercizio 3.3.2.

Soluzione.

```
distribuzione.mod
```

```
set IMP;
```

```
set MAG;
```

```
set DIS;
```

```
param Offerta{IMP};
```

```
param Costo_trasp1{IMP,MAG};
```

```
param Domanda_mag{MAG};
```

```
param Prezzo_mag{MAG};
```

```
param Costo_trasp2{MAG,DIS};
```

```
param Domanda_dis{DIS};
```

```
param Prezzo_dis{DIS};
```

```
var x{IMP,MAG} >= 0;
```

```
var y{MAG,DIS} >= 0;
```

```
var z{m in MAG} >= Domanda_mag[m];
```

```

maximize profitto:
    sum{m in MAG} (Prezzo_mag[m]*z[m] + sum{d in DIS} Prezzo_dis[d]*y[m,d]) -
    sum{i in IMP,m in MAG} Costo_trasp1[i,m]*x[i,m] -
    sum{m in MAG,d in DIS} Costo_trasp2[m,d]*y[m,d];

subject to offerta_imp {i in IMP}: sum{m in MAG} x[i,m] <= Offerta[i];

subject to transito {m in MAG}:
    sum{i in IMP} x[i,m] = sum{d in DIS} y[m,d] + z[m];

subject to domanda_dis {d in DIS}:
    sum{m in MAG} y[m,d] >= Domanda_dis[d];

```

distribuzione.dat

```

set IMP := I1 I2;
set MAG := M1 M2;
set DIS := D1 D2 D3 D4;

param: Offerta :=
I1      3000
I2      2000;

param Costo_trasp1: M1 M2 :=
I1          21  25
I2          27  22;

param: Domanda_mag Prezzo_mag :=
M1      500      150
M2      500      170;

param Costo_trasp2: D1 D2 D3 D4 :=
M1          33  31  36  30
M2          27  30  28  31;

param: Domanda_dis Prezzo_dis :=
D1      150      350
D2      190      280
D3      220      200
D4      170      270;

```
