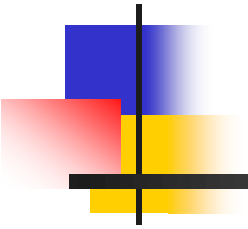

# Mathematical Programming: Modelling and Applications

September 2009

Sonia Cafieri

*LIX, École Polytechnique*

cafieri@lix.polytechnique.fr

# AMPL model – production planning

Starting from the mathematical formulation, try to code the AMPL model

- We have 2 indices for many variables/constraints:

i (products): A1, A2, A3

j (months): 1,2,3,4

It is useful to define 2 sets:

```
set PRODUCTS;
param Months;
set MONTHS := 1..Months
```

We also have $z_{i0}$ -- index 0

```
set MONTHS0 := MONTHS union {0}
```

# AMPL model – production planning

- Now we can define the parameters:

  all parameters are non negative

```
param days{MONTHS} >= 0;
param demand { PRODUCTS, MONTHS } >= 0;
param price { PRODUCTS } >= 0;
param cost { PRODUCTS } >= 0;
param quota { PRODUCTS } >= 0;
param activation { PRODUCTS } >= 0;
param batch { PRODUCTS } >= 0;
param storage { PRODUCTS } >= 0;
param capacity >= 0;
```

- Variables:

```
var x { PRODUCTS, MONTHS } >= 0;
var w { PRODUCTS, MONTHS } >= 0;
var z { PRODUCTS, MONTHS0 } >= 0;
var y { PRODUCTS, MONTHS } >= 0, binary;
```

# AMPL model – production planning

- Objective function:

```
maximize revenue:
sum {i in PRODUCTS}
(price[i] * sum {j in MONTHS} w[i,j] - cost[i] * sum {j in MONTHS} x[i,j]
- storage[i] * sum {j in MONTHS} z[i,j]
- activation[i] * sum {j in MONTHS} y[i,j]) ;
```

- Constraints:

```
subject to requirement {i in PRODUCTS, j in MONTHS}:
        w[i,j] <= demand[i,j];

subject to production {j in MONTHS}:
        sum {i in PRODUCTS} (x[i,j] / quota[i]) <= days[j];

subject to balance {i in PRODUCTS, j in MONTHS}:
        z[i,j-1] + x[i,j] = z[i,j] + w[i,j];

subject to capacitymag {j in MONTHS}:
        sum {i in PRODUCTS} z[i,j] <= capacity;

subject to active {i in PRODUCTS, j in MONTHS}:
        x[i,j] <= days[j]*quota[i]*y[i,j];

subject to minbatch {i in PRODUCTS, j in MONTHS}:
        x[i,j] >= batch[i]*y[i,j];
```

# AMPL dat – production planning

```
set PRODUCTS := A1 A2 A3 ;

param Months := 4 ;

param days :=
     1 23
     2 20
     3 23
     4 22 ;

param demand:   1       2         3          4    :=
        A1     5300    1200      7400        5300
        A2     4500    5400      6500        7200
        A3     4400    6700     12500       13200 ;

param : price    cost quota activation batch storage :=
     A1     124  73.30 500   150000      20    3.5
     A2     109  52.90 450   150000      20    4
     A3     115  65.40 550   100000      16    3   ;

param capacity := 800 ;
```

```
let {i in PRODUCTS} z[i,0] := 0;
fix {i in PRODUCTS} z[i,0];
```

# AMPL run – production planning

```
model productionplanning.mod;
data productionplanning.dat;

option solver cplex;
solve;

option display_round 4;
display revenue;
display x;
display y;
```

# Solution – production planning

```
ILOG AMPL 10.100, licensed to "ecolepolytechnique-palaiseau".
AMPL Version 20060626 (Linux 2.6.9-5.ELsmp)
ILOG CPLEX 10.100, licensed to "ecolepolytechnique-palaiseau", options: e
m b q use=8
CPLEX 10.1.0: optimal integer solution; objective 1581550
33 MIP simplex iterations
0 branch-and-bound nodes
revenue = 1581550.0000

x :=                          y :=
A1 1    6100.0000             A1 1   1.0000
A1 2       0.0000             A1 2   0.0000
A1 3       0.0000             A1 3   0.0000
A1 4       0.0000             A1 4   0.0000
A2 1       0.0000             A2 1   0.0000
A2 2    3518.1818             A2 2   1.0000
A2 3       0.0000             A2 3   0.0000
A2 4       0.0000             A2 4   0.0000
A3 1    4400.0000             A3 1   1.0000
A3 2    6700.0000             A3 2   1.0000
A3 3   12650.0000             A3 3   1.0000
A3 4   12100.0000             A3 4   1.0000
;                             ;
```