Mathematical Programming: Modelling and Applications

September 2009

Sonia Cafieri

LIX, École Polytechnique

cafieri@lix.polytechnique.fr

Outline

- An easy modelling problem
- Formulation of the mathematical model
- The AMPL model
- Files .mod, .dat, .run
- Solution

Mixed production problem

A firm is planning the production of 3 products A1, A2, A3 .

In a month production can be active for 22 days.

The following are given:

- maximum demands (units=100Kg)
- selling price ($/100Kg)
- production costs (per 100Kg of product)
- production quotas (maximum amount of 100Kg units of product that would be produced in a day if all production lines were dedicated to the product).

Mixed production problem

Product	A1	A2	A3
Maximum demand	5300	4500	5400
Selling price	$124	$109	$115
Production cost	$73.30	$52.90	$65.40
Production quota	500	450	550

Formulate an AMPL model to determine the

production plan to maximize the total income

Mathematical model

What is to be identified to write the mathematical formulation?

- Decision variables
- Objective function
- Constraints
- Parameters

- What are the decision variables?

$x_i \quad i \in \{1,2,3\}$: quantity of product i to produce

any bound?

$\forall i \in \{1,2,3\} \quad x_i \geq 0$

Mathematical model

- What is the objective function?

determine the production plan to **maximize** *the* **total income**

$$\max \sum_{i=1}^{3} (v_i - c_i) x_i$$

each x_i *has a* **selling price** *and a* **production cost**

Mathematical model

- What are the constraints?

demand: $\forall i \in \{1,2,3\} \quad x_i \leq d_i$

production: $\sum_{i=1}^{3} \frac{x_i}{q_i} \leq P$

P = number of production days in a month

Mathematical model

- What are the parameters?

P = number of production days in a month

d_i = maximum market demand for product i

v_i = selling price for product i

c_i = production cost for product i

q_i =maximum production quota for product i

Modelling and solving the problem using AMPL

Remember that it is necessary to write:

1. a **model** file (extension `.mod`)

   contains the mathematical formulation of the problem

   - logical structure of the problem -

2. a **data** file (extension `.dat`)

   contains the numerical values of the problem parameters

   - more data files may correspond to the same model -

3. (possibly) a **run** file (extension `.run`)

   specifies the solution algorithm

AMPL model / data

Model file

Logical structure:

1. Parameters ____________ `param name_parameter;`
2. Variables ______________ `var name_variable;`
3. Objective function _ `maximize(minimize) name_objective:…`
4. Constraint(s)_______ `subject to name_constraint: …`

Data file

```
param name_parameter1 := …;
param name_parameter2 := …;
……
```

AMPL model – mixed production

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

- You already know the parameters:

```
days
demand
price
cost
quota
```

All of them are non negative: `>= 0`

`demand,price,cost,quota` are indexed on a set containing the products:

```
set PRODUCTS;

param days >= 0;
param demand { PRODUCTS } >= 0;
param price { PRODUCTS } >= 0;
param cost { PRODUCTS } >= 0;
param quota { PRODUCTS } >= 0;
```

AMPL model – mixed production

- Decision variables: `x`

All of them are non negative: `>= 0`

Are indexed on a set containing the products (previously declared).

- Objective function:

In order to code in ampl the objective function of the mathematical formulation, you just need to know how to write a sum in ampl:

`sum {i in PRODUCTS}` …

- Constraints:

1. Each x_i must be less than or equal to its demand: `subject to requirement {i in PRODUCTS}` ……
2. The sum of x_i/q_i must be less than or equal to the number of the days of production: `subject to production: sum {i in PRODUCTS}` ……

AMPL file mod

```
# mixedproduction.mod

set PRODUCTS;
param days >= 0;
param demand { PRODUCTS } >= 0;
param price { PRODUCTS } >= 0;
param cost { PRODUCTS } >= 0;
param quota { PRODUCTS } >= 0;

var x { PRODUCTS } >= 0;               # quantity of product

maximize revenue: sum {i in PRODUCTS} (price[i] - cost[i]) * x[i];


subject to requirement {i in PRODUCTS}:
x[i] <= demand[i];

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

AMPL file dat

```
# mixedproduction.dat

set PRODUCTS := A1 A2 A3 ;

param days := 22;

param demand :=
A1 5300
A2 4500
A3 5400
;

param price :=
A1 124
A2 109
A3 115
;

param cost :=
A1 73.30
A2 52.90
A3 65.40
;

param quota :=
A1 500
A2 450
A3 550
;
```

Alternatively:

```
param : demand  price  cost   quota :=
     A1  5300    124   73.30  500
     A2  4500    109   52.90  450
     A3  5400    115   65.40  550 ;
```

↑

the same indices set for each param

AMPL file run

```
# mixedproduction.run

model mixedproduction.mod;
data mixedproduction.dat;

option solver cplex;
solve;

display x;
```

Solving the problem with AMPL

1:

```
ampl: model mixedproduction.mod;
ampl: data mixedproduction.dat;
ampl: option solver cplex;
ampl: solve;
ampl: display x;
```

2:

```
cat mixedproduction.run | ampl
```

3:

```
ampl < mixedproduction.run
```

Mixed production: solution

```
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 solution; objective 576483
0 dual simplex iterations (0 in phase I)
x [*] :=
A1  5300
A2   711.818
A3  5400
;
```

Exercise

One step more:

- Change the mathematical program and the AMPL model to cater for a fixed activation cost on the production line, as follows:

Product	A1	A2	A3
Activation cost	$170000	$150000	$100000

- Change the mathematical program and the AMPL model to cater for both the fixed activation cost and for a minimum production batch:

Product	A1	A2	A3
Minimum batch	20	20	16

Mathematical model updated

The basic model is unchanged. But something has to be added.

Parameters. We have 2 parameters more:

a_i = activation cost for the plant producing i

b_i = minimum batch of product i

Variables. For each product i, the production line can be activated or not

y_i = activation status of the product i

Binary variable:

$$\forall i \in \{1,2,3\} \quad y_i = \begin{cases} 1 & \text{if product } i \text{ is active} \\ 0 & \text{otherwise} \end{cases}$$

Mathematical model updated

Objective function. Takes into account the possible activation for each product:

$$\max \sum_{i=1}^{3} \left((v_i - c_i) x_i - a_i y_i \right)$$

Constraints. Two constraints more:

original constraints +

activation: $\forall i \in \{1,2,3\} \quad x_i \leq P q_i y_i$

minimum batch: $\forall i \in \{1,2,3\} \quad x_i \geq b_i y_i$

AMPL file mod updated

```
# mixedproduction.mod

set PRODUCTS;
param days >= 0;
param demand { PRODUCTS } >= 0;
param price { PRODUCTS } >= 0;
param cost { PRODUCTS } >= 0;
param quota { PRODUCTS } >= 0;
param activ_cost { PRODUCTS } >= 0;    # activation costs
param min_batch { PRODUCTS } >= 0;     # minimum batches

var x { PRODUCTS } >= 0;               # quantity of product
var y { PRODUCTS } >= 0, binary;       # activation of production lines

maximize revenue: sum {i in PRODUCTS}
((price[i] - cost[i]) * x[i] - activ_cost[i] * y[i]);

subject to requirement {i in PRODUCTS}: x[i] <= demand[i];

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

subject to activation {i in PRODUCTS}:  x[i] <= days * quota[i] * y[i];

subject to batch {i in PRODUCTS}: x[i] >= min_batch[i] * y[i];
```

AMPL file dat updated

```
# mixedproduction.dat

set PRODUCTS := A1 A2 A3 ;

param days := 22;

param : demand  price  cost   quota  activ_cost  min_batch :=
     A1  5300    124   73.30   500     170000        20
     A2  4500    109   52.90   450     150000        20
     A3  5400    115   65.40   550     100000        16 ;
```

Mixed production updated: solution

```
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 270290
1 MIP simplex iterations
0 branch-and-bound nodes
x [*] :=
A1     0
A2  4500
A3  5400
;

y [*] :=
A1  0
A2  1
A3  1
;
```