Continuous Optimization

Laboratory Assignment on Constrained Continuous Optimization

Optimal Supply Chain strategy with Pricing (SCP)

(v3.0 - 11/18)

F.-Javier Heredia

http://gnom.upc.edu/heredia

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Departament d'Estadística
i Investigació Operativa

Continuous Optimization - Laboratory Assignment

Optimal supply chain strategy with pricing.

F.-Javier Heredia (f.javier.heredia@upc.edu, http://gnom.upc.edu/heredia)

Group on Numerical Optimization and Modeling

Departament d'Estadística i Investigació Operativa, UPC

V3.0, November 2018

Contenido

Description of the (SCP) problem

Let's consider the problem of a company owning a set of factories $i \in \mathcal{F}$ that produce a set of different products $p \in \mathcal{P}$. Each one of these products $p \in \mathcal{P}$ can be sold to any of the markets $j \in \mathcal{M}$ at a price γ_j^p. The company would like to know the price at which these products should be sold so that the total profits earned by the company along a given set of time periods $t \in \mathcal{T}$ is maximized. In order to do so, a linearly constrained nonlinear optimization problem named (SCP) is to be formulated that includes the following assumption on the *manufacturing process*, the *inventory dynamics*, *transportation* between factories and markets, *demand elasticity* and *pricing policy*.

Manufacturing.

The manufacturing process is such that for each factory $i \in \mathcal{F}$ and time period $t \in \mathcal{T}$ there is a limited available amount of each one of the manufacturing resources $r \in \mathcal{R}$. The amount of resource r spent in the production of a unit of product p (a_r^p) is assumed known. There is a linear production unit cost c_i^{pt}[€] that depends on the product p, factory i and time period t.

Inventory.

The unsold production at the end of each time period remains available for the subsequent periods. There is enough room at each factory to store any amount of products but every unit held in the inventory incurs a known holding unit cost h_i^p[€] that does not depend on the time period. Each factory stores an initial inventory s_j^p at the beginning of the first time period.

Transportation.

At any time period $t \in \mathcal{T}$, the company can sell any available quantity of product $p \in \mathcal{P}$ produced in any factory $i \in \mathcal{F}$ to any market $j \in \mathcal{M}$. There is a known linear transportation unit cost f_{ij}^p[€] from factory i to market j that does not depend on the time period.

Demand elasticity.

We assume that the demand of each market is a linear function of the price with known (negative) slope. Let $\tilde{\gamma}^p$ be the current price for each product (the same for all markets and time periods), and $\tilde{d}_j^{pt}$ the current demand for each product, time period and market. Then, the actual demand in terms of the price γ_j^p can be expressed as:

$$d_j^{pt} = \tilde{d}_j^{pt} + \alpha_j^p\left(\gamma_j^p - \tilde{\gamma}^p\right), \quad j \in \mathcal{M},\ p \in \mathcal{P},\ t \in \mathcal{T}.$$

where the slope $\alpha_j^p < 0$ is the *demand response* to the price γ_j^p.

Price spread policy.

The company considers that the price difference of the same product in different markets, γ_j^p and γ_k^p, $j \neq k$, should be bounded. In order to impose that condition, the optimal solution must satisfy that every price γ_j^p must be greater than a given fraction β^{min} and less that a given fraction β^{max} $(0 < \beta^{min} < 1 < \beta^{max})$ of the average price for product p among all the markets j, $\bar{\gamma}_p = {\sum_{j\in\mathcal{M}} \gamma_j^p}\big/{|\mathcal{M}|}$.

Test case

The (SCP) problem formulated in the previous section must be used to solve an extension of the problems **`steelP`** and **`steelT`** described in the lecture notes of the course. In these problems we have:

- Four time periods, $\mathcal{T} = \{$**`1,2,3,4`**$\}$
- Three products $\mathcal{P} = \{$**`bands, coils, plate`**$\}$.
- Three factories $\mathcal{F} = \{$**`GARY, CLEV, PITT`**$\}$.
- Seven markets $\mathcal{M} = \{$**`FRA, DET, LAN, WIN, STL, FRE, LAF`**$\}$.

Each student shall solve the (SCP) problem with a unique data set than can be found in the file **`steelSCP.dat`**. The following table shows the correspondence between datasets and students:

DNI	Dataset
00077526	1
43216930	2
38100847	3
49188055	4
41650265	5
04580010	6

DNI	Dataset
01700125	7
47924628	8
01477699	9
47334294	10
00080318	11

Each dataset is composed of the following information:

`param`	
`FACT, MARK, PROD, TIME`	$i \in \mathcal{F}, j \in \mathcal{M}, p \in \mathcal{P}, t \in \mathcal{T}$
`avail`	b_i: available hours.
`inv0`	z_0^p: initial stock.
`rate`	$a_r^p\, [Tm/h]$
`demand_current`	$\tilde{d}_j^{pt}$
`demand_response`	α_j^p

`param`	
`price_current`	$\tilde{\gamma}^p$
`price_min`	β^{min}
`price_max`	β^{max}
`make_cost`	c_i^{pt}: production cost.
`trans_cost`	f_{ij}^p: transportation cost.
`inv_cost`	h_i^p: holding costs.

In addition, to help you verifying the correctness of your code, the value of the objective function at the optimal solution is included:

```
MINOS 5.51: optimal solution found.
257 iterations, objective 2719395.264
Nonlin evals: obj = 364, grad = 363.
```

Report

You must upload to Atenea a unique file **`surname.zip`** or **`surname.rar`** with the surname of the student and the following contents:

1. A report (**`.pdf`**) with the following sections:
    a. A cover with the title of the assignment and the name of the student
    b. A description of the problem (SCP).
    c. The mathematical formulation of the problem (SCP).
    d. The implementation in AMPL, including a table with two columns listing, in the first column, the different parts of the mathematical model (sets, parameters, variables, objective function and constraints) and, in the second column, the associated element in the AMPL implementation.
    e. The optimal solution found. In this section **you must create a series of tables** using the **`display`**, **`print`** and **`printf`** at your best convenience (see Chapter 12 of the AMPL Book) to show the following information:
        i. The incomes (sales) and costs (manufacturing, inventory, transportation) at each time period.
        ii. The optimal pricing policy and its effect on demand, showing the change w.r.t. the original values.
        iii. The average variation, in %, of price and demand for every product.
        iv. The optimal inventory policy for each product and factory.
        v. The optimal transportation policy.

    The annex of this document shows an example of these tables. Each table must include your comments discussing the results. Any graphical representation facilitating the understanding of the results will be welcoming.

2. The **`AMPL`** implementation of the (SCP) code: **`.run`**, **`.mod`** and **`.dat`** files.

We encourage students to consult the AMPL Book as a help in the development of the AMPL implementation.

Annex: example of output tables

```
Costs :::::::::::::::::::::::::::::::::::::::::::::::::::::::::

:  inc_sales cost_manuf   cost_inv  cost_trans   profit     :=
1   10307500   -6545840   -126026     -1998790   1636810
2   11068700   -6486630    -94677.9   -2042660   2444780
3   12176000   -7055130   -102555     -2373860   2644450
4   11910900   -2842900          0    -2253450   6814600
;

Optimal pricing policy and demand elasticity :::::::::::::::::::::::::::::::::::::::::

p = bands

# $3 = sum{t in TIME} demand_current[j,p,t]
# $4 = sum{t in TIME} demand_actual[j,p,t]
:   price_current[p] Price[j,p].val   $3      $4        :=
FRA       210.4         231.362      6392    1948
DET       210.4         213.465      7732    6604
LAN       210.4         216.133      6353    4289
WIN       210.4         223.843      8289    4525
STL       210.4         217.287      4724    3016
FRE       210.4         237.037      9855    2076.88
LAF       210.4         235.688      9937    3261
;
```

(the same for the remaining products)

```
Optimal inventory management:::::::::::::::::::::::::::::::::::::::::::::::::::::::::

# $4 = Inv[i,p,prev(t, TIME0)]
# $5 = sum{j in MARK} (Trans[i,j,p,t].val)
:     p       i    Make[i,p,t].val    $4    $5  Inv[i,p,t].val    :=
1   bands   GARY          0         2057   410        1647
2   bands   GARY          0         1647     0        1647
3   bands   GARY          0         1647   715         932
4   bands   GARY          0          932   932           0
;
```

(the same for the remaining combination of products and factories)

```
Optimal transportation policy :::::::::::::::::::::::::::::::::::::::::::::::::::::::

Trans [CLEV,*,bands,*]
:         1          2          3          4        :=
FRA       0          0          0          0
DET    1406.51       0          0          0
LAN     476          0        423       3390
WIN     681          0          0          0
STL       0          0          0          0
FRE     711.471    710.471    614.471     40.4708
LAF       0          0          0          0
```

(the same for the remaining combination of factories and products)