Projet d'aide à la décision

Paul Adenot, Etienne Brodu, Maxime Gaudin, Monica Golumbeanu, Yoann Rodière

11 octobre 2010

Table des matières

Résumé

Première partie

Programmation Linéaire monocritère

1 Données

Soient :
- **T** la matrice des temps unitaires d'usinage d'un produit sur une machine (minutes) (*C.f. Table 1*).
- **Q** la matrice de quantité de matières premières par produit (*C.f. Table 2*).
- **S** la matrice des quantité maximum de matières premières (*C.f. Table 3*).
- **V** la matrice des prix de vente des produits finis (*C.f. Table 4*)
- **A** la matrice des prix d'achat des matières premières.
- **C** la matrice des coûts horaires des machines (*C.f. Table 5*).

1.1 Contraintes

Considérons :
- 6 produits identifiés chacun par une lettre $i \in A, B, C, D, E, F$
- 7 machines identifiée chacune par un nombre $j \in 1, 2, 3, 4, 5, 6, 7$
- 3 matières premières identifiée chacune par un nombre $k \in 1, 2, 3$
- n, vecteur colonne du nombre d'unités fabriquées pour chaque produit

L'ensemble de la chaine de production est régie par les contraintes suivantes :

- **Le nombre de produits usinés de type i :** Il doit être non nul

$$n_i \geq 0, \forall i \in A, B, C, D, E, F \tag{1}$$

- **Le temps d'occupation de chaque machine i :** Il doit être inférieur au temps de travail

$$\sum_{j=A}^{F} T_{j,i} \times n_j \leq 2 \times 8 \times 60 \times 5 = 4800, \forall i \in 1, 2, 3, 4, 5, 6, 7 \tag{2}$$

  soit un temps de travail en deux huit, 5 jours par semaine.
- **L'utilisation de chaque matière première i :** Elle doit être inférieure au stock

$$\sum_{j=A}^{F} Q_{i,j} \times n_j \leq S_i, \forall i \in 1, 2, 3 \tag{3}$$

1.1.1 Modélisation sous forme matricielle

Pour donner au problème une forme standard, nous allons le modéliser par des inéquations et des produits matriciels. Les contraintes C0, C1 et C2 se traduisent trivialement de la manière suivante :

$$A.n \leq b \tag{4}$$

Avec :

$$A = \left(\begin{array}{c} \\ -I \\ \\ \hline \\ T^t \\ \\ \hline Q \end{array}\right), b = \left(\begin{array}{c} 0\\0\\0\\0\\0\\0 \\ \hline 4800\\4800\\4800\\4800\\4800\\4800\\4800 \\ \hline S^t \end{array}\right)$$

Ce qui nous donne plus concrètement les matrices suivantes :

$$A = \left(\begin{array}{cccccc} -1&0&0&0&0&0\\ 0&-1&0&0&0&0\\ 0&0&-1&0&0&0\\ 0&0&0&-1&0&0\\ 0&0&0&0&-1&0\\ 0&0&0&0&0&-1\\ \hline 8&15&0&5&0&10\\ 0&1&2&15&7&12\\ 8&1&11&0&10&25\\ 2&10&5&4&13&7\\ 5&0&0&0&10&25\\ 5&5&3&12&8&0\\ 5&3&5&8&0&0\\ \hline 1&2&1&5&0&2\\ 2&2&1&0&2&1\\ 1&0&3&2&2&0 \end{array}\right), b = \left(\begin{array}{c} 0\\0\\0\\0\\0\\0\\ \hline 4800\\4800\\4800\\4800\\4800\\4800\\4800\\ \hline 350\\620\\485 \end{array}\right)$$

2 Objectif : Comptable

Le comptable cherche à maximiser les bénefices sous les contraintes définies précedemment.

2.1 Modélisation

Soit n_i le nombre de produit i fabriqué. Le coup fixe de production n'influant pas sur notre décision, nous ne considérerons que le coût variable de production. Il est défini par la formule suivante :

$$CV(i) = n_i \times \left(\sum_{j=1}^{7} T_{i,j} \times \frac{C_{i,j}}{60} + \sum_{k=1}^{3} A_k \times Q_{k,i} \right)$$

Le chiffre d'affaire par produit est :

$$CA(i) = n_i \times V_i$$

Par conséquent le bénefice par produit se calcule de la manière suivante :

$$\begin{aligned} B(i) &= CA(i) - CV(i) \\ &= n_i \times \left(V_i - \sum_{j=1}^{7} T_{i,j} \times \frac{C_{i,j}}{60} + \sum_{k=1}^{3} A_k \times Q_{k,i} \right) \end{aligned}$$

La matrice permettant de calculer, à partir du vecteur colonne n du nombre de produits sortis de l'usine, le bénéfice d'*Optim* est donc la suivante :

$$\begin{aligned} M_B &= \left(V - \left(\left(T \times C^t \times \frac{1}{60} \right)^t + (A \times Q) \right) \right) \\ &\simeq \begin{pmatrix} 6.0667 & 11.9833 & 12.4333 & 9.5333 & 31.6500 & 27.9000 \end{pmatrix} \end{aligned}$$

Remarquons qu'elle nous donne explicitement le bénéfice unitaire pour chaque produit. Le produit E est *a priori* le plus intéressant.

Nous chercherons à maximiser la fonction linéaire correspondant à cette matrice, donc (pour prendre une forme plus standard) à minimiser son opposé.

2.2 Décisions

En utilisant les outils Matlab (`linprog` en l'occurence, cf. annexes pour le code exact), on obtient le résulat suivant :

$$n_{optimal} = \begin{pmatrix} 0.0000 \\ 20.4082 \\ 0.0000 \\ 0.0000 \\ 242.5000 \\ 94.1837 \end{pmatrix}$$

Lecture : le produit 1 (A) doit être abandonné, le produit 5 (E) doit être produit en 242,5 exemplaires (242 en entier et un demi exemplaire, terminé la semaine suivante), etc.

Ce résultat était (en partie) prévisible à partir de la matrice M_B, puisque le produit E est le plus rentable. On devra donc, pour optimiser le bénéfice, en produire le plus possible, tout en utilisant intelligemment les matières premières et ressources humaines restantes pour maximiser le reste du bénéfice. Ainsi, d'après le résultat on préfère fabriquer le produit B au lieu du produit C car il est moins couteux en matière première MP3.

Notons tout de même que ce résultat n'était pas si évident, puisqu'il prend également en compte l'utilisation des ressources (les machines), que nous avons ignorée dans le raisonnement ≪ intuitif ≫ ci-dessus.

3 Objectif : Responsable d'atelier

Le responsable d'atelier cherche à maximiser le nombre d'unités (toutes catégories confondues) produites sous les contraintes définies précedemment.

3.1 Modélisation

Soit N le nombre de produits fabriqués.

$$N = \sum_{i=A}^{F} \tag{5}$$

3.2 Décisions

4 Objectif : Responsable des stocks

Le responsable des stocks cherche à minimiser le nombre de de produits dans son stock sous les contraintes définies précedemment.

4.1 Modélisation

Soit $\Delta Stock(n_i)$ la variation du nombre d'unités en stock (l'unité est inconnue) en fonction du nombre de produits frabriqués. Cette fonction est la somme des produits fabriqués à laquelle on soustrait la quantité de matières premières utilisée.

On suppose qu'un produit frabriqué correspond à une unité de stock.

On a ainsi la formule suivante, où n_i est la quantité de produit usiné (pour chaque produit i), et $Q_{i,j}$ est la quantité de matière première par produit pour chaque produit i et chaque matière première j.

$$\Delta Stock(n_i) = \sum_i n_i - n_i \times \sum_j Q_{i,j} \tag{6}$$

$\Delta Stock(n_i)$ sera une fonction négative ou nulle : en effet, pour chaque unité de produit frabriqué, au moins une unité de matière première est consommée. En cherchant à minimiser cette expression, on minimise le stock.

4.2 Décisions