Projet d'aide à la décision

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

12 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.
Autrement dit, seul la quantité de matières premières disponible et le temps maximum de travail limitera la production.

3.1 Modélisation

Soit N le nombre de produits fabriqués.

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

Par concéquent, nous obtenons la matrice suivante, qui représente la somme des différents produits :

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

La matrice des contraintes quant à elle, n'est pas modifiée.

3.2 Décisions

Instinctivement, on sait qu'il faudra fabriquer l'unité consommant le moins de ≪ temps machine ≫ et le moins de matières premières. On peux donc penser que E et F seront les produits les plus fabriqués.
En pratique, on obtient les résultats suivants :

$$M = \begin{pmatrix} 0 \\ 56.732 \\ 38.6928 \\ 0 \\ 182.4608 \\ 98.9216 \end{pmatrix}$$

Ceci confirme bien les résultats attendus, c'est à dire qu'il faut donner la priorité aux produit consommant le moins de ressources.
On remarque aussi qu'il est préférable d'abandonner A et C au profit d'autres produits.

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.

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

$\Delta Stock(n_i)$ 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.

On obtient la représentation matricielle suivante :

$$M_S = (1, \ldots, 1) \times (Id - Q) \qquad (7)$$

(Application numérique...)

4.2 Décisions

$\Delta Stock(n_i)$ est minimum quand le maximum d'unités de matières premières est écoulé. Ainsi, la combinaison optimale favorise la production des produits consommant le plus de matières premières de manière à en éliminer le plus (il vaut mieux par exemple fabriquer deux produits consommant 6 unités de matières premières qu'un seul en consommant 4).

5 Objectif : Responsable commercial

Le responsable commercial cherche à équilibrer le nombre d'unités de A, B, C (famille 1) et D, E, F (famille 2) afin que ces deux familles contiennent le même nombre d'unités (à ϵ unité(s) prés).
En prennant en compte les contraintes définies avant et en ajoutant une nouvelle contrainte d'équilibre entre les deux familles de produits on se rend compte que la meilleure solution sera de ne rien produire. C'est solution n'est pas avantageuse. On introduit par conéquent une autre contrainte - nous allons essayer d'équilibrer les familles de produits tout en conservant un bénéfice maximal.

5.1 Modélisation

Soient :
- n_A le nombre de produits A usinés.
- n_B le nombre de produits B usinés.
- n_C le nombre de produits C usinés.
- n_D le nombre de produits D usinés.
- n_E le nombre de produits E usinés.
- n_F le nombre de produits F usinés.

Pour étudier l'équilibre entre les deux familles des produits on définira une fonction qui sera égale à la différence entre les quantités des produits provenant des deux familles, soit :

$$F = (n_A + n_B + n_C) - (n_D + n_E + n_F)$$

La contrainte sur l'équilibre va se traduire par l'essai de minimiser la fonction F, c'est à dire :

$$\begin{aligned} |F| &\leq \epsilon \\ \Leftrightarrow -\epsilon \leq F &\leq \epsilon \\ \Leftrightarrow -\epsilon \leq (n_A + n_B + n_C) - (n_D + n_E + n_F) &\leq \epsilon \\ Avec \quad \epsilon \to 0. \end{aligned}$$

Ensuite on va essayer de maximiser le bénéfice obtenu. On va procéder de la manière suivante :
- Nous fixons un certain bénéfice à atteindre, par exemple 60% du bénéfice maximal.
- Pour le bénéfice choisi, nous essayons de minimiser F.

En répétant cette démarche un certain nombre de fois, nous pouvons tracer une courbe représentant la valeur de F en fonction du bénéfice atteint. L'interprétation de cette courbe nous permettra de trouver le meilleur compromis entre nos deux objectifs.

5.2 Décisions

Deuxième partie

Programmation linéaire multicritère

L'objectif est de trouver une solution de compromis entre les différents responsables. Pour trouver une telle solution nous serons amenés à utiliser la programmation multicritère (PLM). Auparavant dans la partie 1, nous avons trouvé un optimum pour chaque responsable indépendamment, ce qui nous conduit à un point de mire. Dans un monde parfait ce point de mire respecterait les contraintes de chaque responsable. Nous devons donc voir si telle est le cas.

6 Recherche du point de départ

Si le point de mire est assez proche de l'ensemble des solutions acceptables, nous choisirons une solution proche de celle d'un responsable.

Sinon, nous allons calculer la satisfaction de chaque objectif, sachant qu'une solution a été retenue. Nous devrons alors définir des métriques, correspondant à cette satisfaction. Par exemple, pour le comptable, cette satisfaction sera exprimée par le ratio du bénéfice obtenue dans un solution par rapport au bénéfice maximal. Ensuite, nous choisirons comme point de départ la solution qui offre le plus de satisfaction à tout le monde, par exemple en utilisant une moyenne pondérée, dont la pondération sera basée sur *l'importance* de chaque critère.

7 Affinement de la solution

La solution trouvée précédemment peut sûrement être optimisée. Il peut être intéressant de perdre dans un critère, si cela nous fait gagner beaucoup dans un autre critère, d'autant plus si ce second critère est jugé plus *intéressant* que le premier.

8 Métriques utilisées

Cette section décrit les métriques utilisées pour caractériser une solution, du point de vue d'un cadre de l'entreprise. Les solutions pourront ainsi être comparées entre elles.

Comptable : La métrique utilisée sera le pourcentage du bénéfice par rapport au bénéfice maximum :

$$M_{Comptable} = \frac{B_S}{B_{max}} \times 100$$

Responsable d'atelier La métrique utilisée sera le pourcentage du nombre de produits fabriqués par rapport au nombre maximum :

$$M_{Atelier} = \frac{N_S}{N_{max}} \times 100$$

Responsable des stocks La métrique utilisée sera la somme de la valeur du stock (le prix unitaire de chaque matière première avec la quantité de cette matière première en stock), plus le nombre de produit fabriqués (et donc en stock), multiplié par la quantité nécessaire de matière première pour fabriquer chaque produit. On peut imaginer que plus un produit nécessite de matière première, et plus cette matière première est chère, plus il sera cher à stocker.

Il pourra être nécessaire de mettre cette métrique sous forme de pourcentage, pour en faciliter la lecture.

$$\begin{cases} Q_M Pi : \text{quantité de matière première } i. \\ Pu_{Mpi} : \text{prix de la matière première } i. \\ QNec_{i_{MPj}} : \text{quantité nécessaire de matière première } j \text{ pour le produit } i. \end{cases}$$

$$M_{Stock} = \left(\sum_{i=1}^{3} Q_{MPi} \times Pu_{Mpi} + \sum_{i=A}^{F} \sum_{j=1}^{3} QNec_{i_{MP1j}} \times Pu_{Mpj} \right)^{-1}$$

Responsable commercial La métrique utilisée sera l'écart par rapport à un équilibre parfait. Si autant de produit de la famille de produit 1 (comprenant les produits A, B et C) que de la famille 2 (comprenant les produits D, E et F), la métrique sera à 100%. Si une seule famille de produit est fabriquée, la métrique devra valoir zéro.

Si F_1 (respectivement F_2 est le nombre de produit de la famille 1 (respectivement famille 2), la métrique sera :

$$M_{Commercial} = \left(1 - \frac{|F_1 - F_2|}{F_1 + F_2}\right) \times 100$$

9 Utilisation

Les résultats seront placés dans un tableau de ce type, qui qui permettra d'un seul coup d'œil de voir la meilleur des solutions. La colonne en rouge se lira par exemple :

« En suivant la volonté du responsable d'atelier, le comptable aura une satisfaction de n% »

Un calcul de moyenne pourra être un premier indicateur de qualité d'une solution. Cette moyenne pourra être pondérée dans le future.

	Comptable	Atelier	Stock	Commercial	Moyenne
Comptable	100%	2	3	4	
Atelier	1	100%	3	4	
Stock	1	2	100%	4	
Commercial	1	2	3	100%	

Table 1 – Tableau de satisfaction des différents cadres de l'entreprise en fonction de la solution retenue.

Troisième partie

Annexe

10 Code source

```
function [ OptProduction, MaxEarnings ] = Comptable( )
%COMPTABLE Summary of this function goes here
%   Detailed explanation goes here

FetchData;

% This is the matrix corresponding to the earnings function
Earnings = (V - ( (T * C' ./ 60)' + (A * Q) ))

% We'll try to maximize the function, so we minimize the opposite
Earnings = -Earnings;

% Optimisation
OptProduction = zeros(size(Earnings,2),1)
OptProduction = linprog(Earnings, InfEqConstraints, InfEqValues);

% Effective earnings
MaxEarnings = -Earnings * OptProduction;
```

```
function [ N ] = Comptable( )

FetchData();

% This is the matrix corresponding to the earnings function
Products = [1; 1; 1; 1; 1; 1];

% We'll try to maximize the function, so we minimize the opposite
Products = -Products;

% Optimisation
N = linprog(Products, InfEqConstraints, InfEqValues);
```