

Projet d'aide à la décision

Paul ADENOT, Etienne BRODU, Maxime GAUDIN,
Monica GOLUMBEANU, Nor EL MALKI, Yoann RODIÈRE

12 octobre 2010

Table des matières

I	Programmation Linéaire monocritère	3
1	Données	4
1.1	Contraintes	4
1.1.1	Modélisation sous forme matricielle	5
2	Objectif : Comptable	6
2.1	Modélisation	6
2.2	Décisions	6
3	Objectif : Responsable d'atelier	8
3.1	Modélisation	8
3.2	Décisions	8
4	Objectif : Responsable commercial	9
4.1	Modélisation	9
4.2	Décisions	9
5	Objectif : Responsable des stocks	10
5.1	Modélisation	10
5.2	Décisions	11
II	Programmation linéaire multicritère	12
6	Recherche du point de départ	13
7	Affinement de la solution	13
8	Métriques utilisées	13
9	Utilisation	14
III	Annexe	15
10	Code source	16

Résumé

L'objet de ce TP est de trouver une stratégie de production, menant à un contentement optimal des différents acteurs :

- Le comptable
- Le responsable des ateliers
- Le responsable des stocks
- Le responsable commercial

La démarche sera quant à elle incrémentale puisque nous fournirons dans un premier temps des optimums très locaux, respectant un nombre limité de contraintes et de critères, puis nous combinerons ces différents résultats afin de converger vers une solution prenant en compte l'intérêt de chacun.

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 identifiées chacune par un nombre $j \in 1, 2, 3, 4, 5, 6, 7$
- 3 matières premières identifiées chacune par un nombre $k \in 1, 2, 3$
- n , vecteur colonne du nombre d'unités fabriquées pour chaque produit

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

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

$$n_i \geq 0, \forall i \in A, B, C, D, E, F \quad (1)$$

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

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

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

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

2 Objectif : Comptable

Le comptable cherche à maximiser les bénéfices sous les contraintes définies précédemment.

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 bénéfice 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, on obtient le résultat suivant :

```
1 N = linprog(Units, InfEqConstraints, InfEqValues);
```

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

iiiiiii HEAD Le bénéfice maximum est donc 357.0919 Unités Monétaires.

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.

===== Le bénéfice maximum est donc 357.0919 ☒.

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. ~~~~~

77f155c849fbbcbf30e915678b3a59b086f78877

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 précédemment.

Autrement dit, seul la quantité de matières premières disponible et le temps maximum de travail limitera la production. Il n'intervient donc pour le moment aucune notion de bénéfice.

3.1 Modélisation

Soit N le nombre de produits fabriqués.

$$N = \sum_{i=A}^F n_i \quad (5)$$

Par conséquent, 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 peut donc penser que E et F seront les produits les plus fabriqués.

En pratique, on obtient les résultats suivants :

```
1 N = linprog(Units, InfEqConstraints, InfEqValues);
```

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

On peut donc, au maximum, fabriquer 376.8072 unités, tous produits confondus.

Ceci confirme bien les résultats attendus, c'est à dire qu'il faut donner la priorité aux produits consommant le moins de ressources.

On remarque cependant qu'il est préférable d'abandonner A et C au profit d'autres.

4 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) près).

En prenant 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 conséquent une autre contrainte - nous allons essayer d'équilibrer les familles de produits tout en conservant un bénéfice maximal.

4.1 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 :

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

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.

4.2 Décisions

5 Objectif : Responsable des stocks

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

5.1 Modélisation

Soit $Stock(n_i)$ le nombre d'unités de stock nécessaires pour stocker les produits fabriqués et la matière première nécessaire. Cette fonction est la somme des produits fabriqués à laquelle on ajoute la quantité de matières premières nécessaire à la fabrication.

On suppose qu'un produit fabriqué 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 $Q_{j,i}$ est la quantité de matière première par produit pour chaque produit i et chaque matière première j .

$$Stock(n) = \sum_i (n_i + n_i \times \sum_j Q_{j,i}) \quad (6)$$

La représentation matricielle de cette fonction sera alors :

$$M_S = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} + ((1 \ 1 \ 1) \times Q) \quad (7)$$

Le résultat trivial est :

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

En effet, en l'absence de production, aucun stock n'est nécessaire. Ce résultat n'est pas satisfaisant. Nous allons donc nous intéresser à un second critère : les bénéfices de l'entreprise. Pour ce faire, nous allons nous intéresser aux valeurs obtenues en imposant un minimum de bénéfices. En réalisant les calculs sur plusieurs échelons, on obtient un résultat linéaire par morceaux. Le raisonnement adopté est similaire à celui développé dans la section ??.

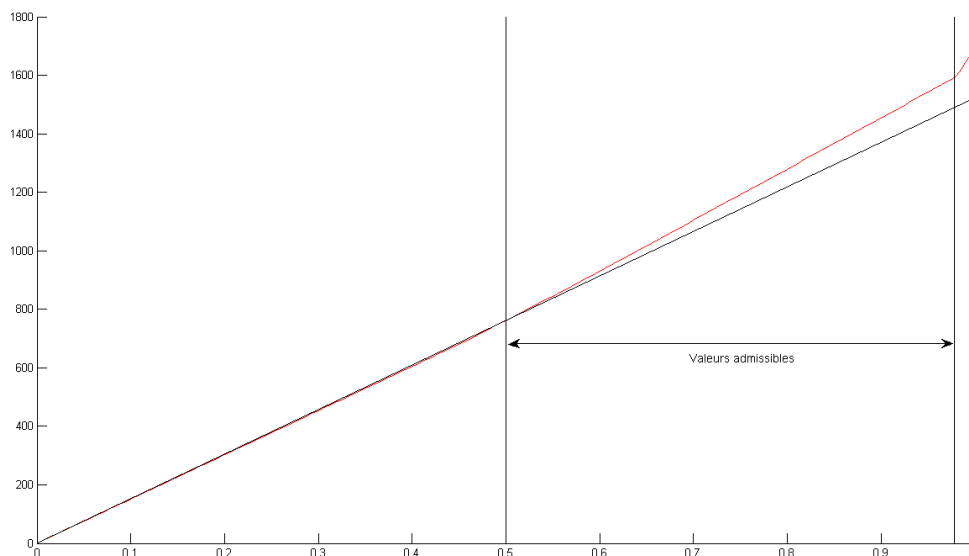


FIGURE 1 – Graphe de l'évolution des stocks en fonction du bénéfice

5.2 Décisions

De nombreuses valeurs sont équivalentes et ne peuvent être départagées selon des critères mathématiques. Les choix possibles se situent dans le deuxième morceau de courbe : en dessous, les bénéfices sont trop bas, au dessus, les besoins de stockage augmentent beaucoup plus que les bénéfices.

On a donc une valeur comprise entre 50% et 98% de bénéfices. Pour 75% du bénéfice maximum, on obtient le nombre de produits suivants :

$$\begin{pmatrix} 1,91903382074088 \times 10^{-10} \\ 2,63753463514149 \times 10^{-10} \\ 1,89174897968769 \times 10^{-10} \\ 1,23691279441118 \times 10^{-10} \\ 124,634235411818 \\ 142,146305832495 \end{pmatrix} \text{ pour une quantité d'unités en stock de } 1191,75640039347.$$

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 devons 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.

$$\begin{cases} Q_{MPi} : \text{quantité de matière première } i \text{ dans le stock à un instant } t. \\ Pu_{MPi} : \text{prix de la matière première } i. \\ QNec_{i_{MPj}} : \text{quantité nécessaire de matière première } j \text{ pour le produit } i. \end{cases}$$

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

$$M_{Stock} = \left(1 - \frac{Q_{produit} + Q_{MP}}{Q_{MP_{Max}}}\right) \times 100$$

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.

	Comptable				Atelier				Stock				Commercial			
Comptable	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Atelier	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Stock	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Commercial	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4

Troisième partie
Annexe

10 Code source

```
1 function [ OptProduction, MaxEarnings ] = Comptable( )
2 %COMPTABLE Summary of this function goes here
3 % Detailed explanation goes here
4
5 FetchData;
6
7 % This is the matrix corresponding to the earnings function
8 Earnings = (V - ( T * C' ./ 60)' + (A * Q) ))
9
10 % We'll try to maximize the function, so we minimize the opposite
11 Earnings = -Earnings;
12
13 % Optimisation
14 OptProduction = zeros(size(Earnings,2),1)
15 OptProduction = linprog(Earnings, InfEqConstraints, InfEqValues);
16
17 % Effective earnings
18 MaxEarnings = -Earnings * OptProduction;
```



```
1 function [ N ] = Atelier( )
2
3 FetchData;
4
5 % This is the matrix corresponding to the earnings function
6 Units = [1; 1; 1; 1; 1; 1];
7
8 % We'll try to maximize the function, so we minimize the opposite
9 Units = -Units;
10
11 % Optimisation
12 N = linprog(Units, InfEqConstraints, InfEqValues);
```