Projet d'aide à la décision

Paul Adenot, Etienne Brodu, Maxime Gaudin,
Monica Golumbeanu, Nor El Malki, Yoann Rodière

17 octobre 2010

Table des matières

Résumé

L'objet de ce TP est de trouver une stratégie de production, menant à un contentement optimal des différent 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 optimum très locaux, respectant un nombre limités de contraintes et de critères, puis nous combinerons ces différents résultat 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 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 :

$$A = \begin{pmatrix} -I \\ \hline T^t \\ \hline Q \end{pmatrix}, b = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \hline 4800 \\ 4800 \\ 4800 \\ 4800 \\ 4800 \\ 4800 \\ 4800 \\ \hline S^t \end{pmatrix}$$

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, on obtient le résulat suivant :

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

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

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.

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. Il n'intervient donc pour le moment aucune notion de bénéfice.

3.1 Modélisation

Soit N le nombre de produits fabriqués.

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

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 :

```
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).
Autrement dit, l'écart entre le nombre d'unités produite pour la famille A et la famille B doit être inférieur à un seuil ϵ.

4.1 Modélisation

Soient :
- N_1 le nombre de produits de la famille 1 fabriqués.
- N_2 le nombre de produits de la famille 2 fabriqués.

$$\begin{aligned} |N_1 - N_2| &\leq \epsilon \\ \Leftrightarrow -\epsilon \leq N_1 - N_2 &\leq \epsilon \\ \Leftrightarrow -\epsilon \leq \sum_{i=A}^{C} n_i - \sum_{j=D}^{F} n_j &\leq \epsilon \end{aligned}$$

Par concéquent, c'est cette nouvelle contrainte qui, venant s'ajouter aux contraintes précédentes, va permettre de calculer le nombre d'unités A, B, C, D, E, et F à fabriquer afin d'équilibrer les deux familles.

Nous obtenons la matrice suivante :

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

La matrice, très simple, représente la somme des différents produits.
La matrice des contraintes devient quant à elle : INSÉRER MATRICE A MODIFIEE

4.2 Décisions

4.3 Interprétation

Evidemment, toutes les solutions triviales du type :

$$M_p = \begin{pmatrix} N \pm \epsilon & N \pm \epsilon & N \pm \epsilon & N & N & N \end{pmatrix}$$

ou encore

$$M_p = \begin{pmatrix} N & N & N & N \pm \epsilon & N \pm \epsilon & N \pm \epsilon \end{pmatrix}$$

Où M_p est la matrice du nombre de produit, et $N \in \mathbb{N}$

sont des solutions *valables*.

Ceci met en évidence qu'avec les critères définis plus haut, il n'y a pas de solution plus ≪ valable ≫ qu'une autre. Par concéquent nous pouvons :
- Choisir une solution au hasard
- Augmenter le nombre de critère et notamment ceux en rapport avec les stocks disponibles, le prix des matières premières, ou encore le temps d'usinage nécessaire.

5 Objectif : Responsable des stocks

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

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 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 $Q_{j,i}$ est la quantité de matière première par produit pour chaque produit i et chaque matière première j.

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

La représentation matricielle de cette fonction sera alors :

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

Le résultat trivial est :

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{8}$$

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 raisonement adopté est similaire à celui développé dans la section **??**.

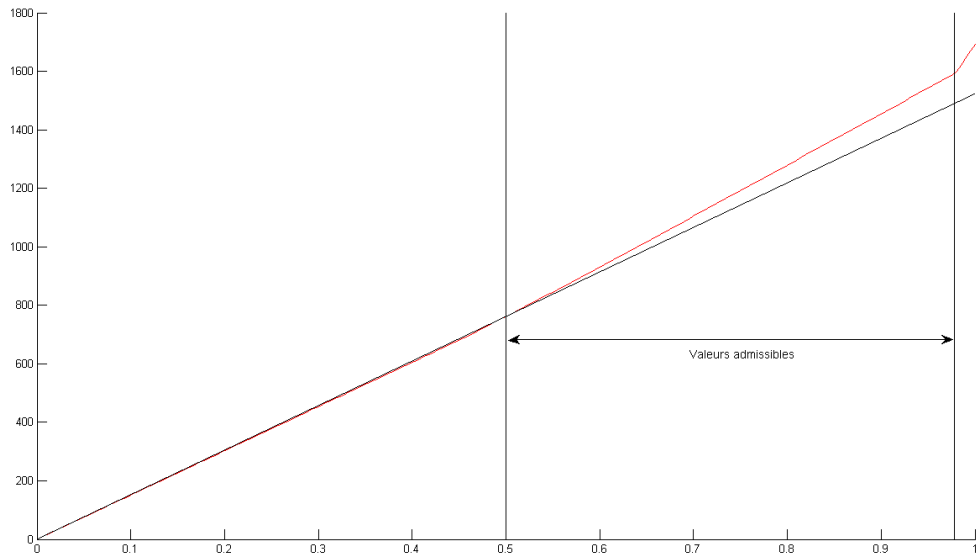

FIGURE 1 – Graphe de l'évolution des stocks en fonction du bénéfice

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 :

$$\begin{pmatrix} 1,91903382074088 \times 10^{-10} \\ 2,63753463514149 \times 10^{-10} \\ 1,89174897968769 \times 10^{-10} \\ 1,23691279441118 \times 10^{-10} \\ 124,634235411818 \\ 142,146305832495 \end{pmatrix} \tag{9}$$

pour une quantité d'unités en stock de 1191,75640039347.

6 Représentation graphique

Les graphiques ci-dessous représentent les résultats obtenus dans cette partie.

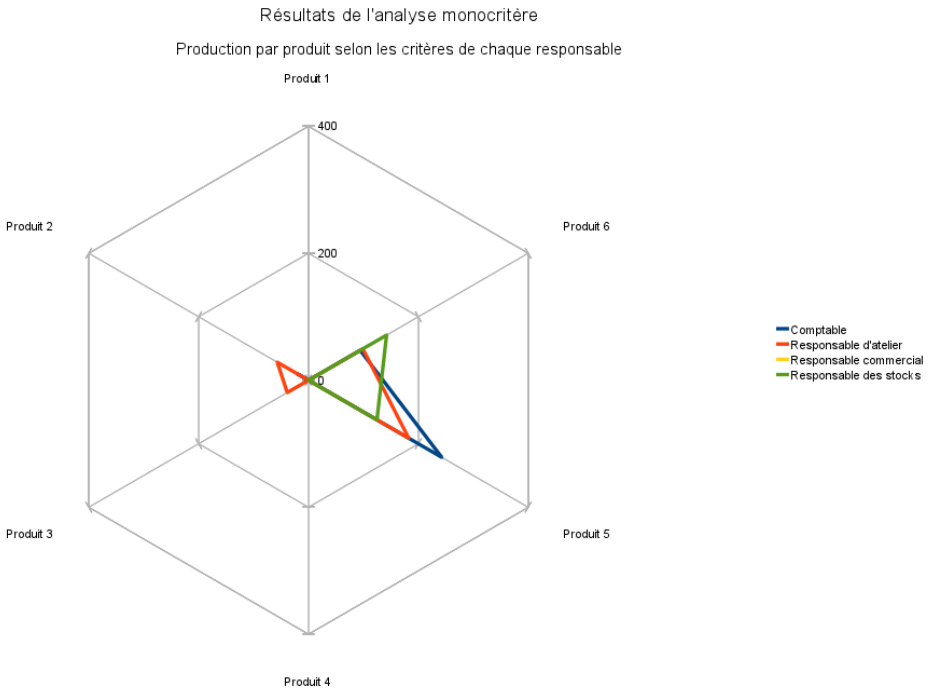

FIGURE 2 – Graphe radar représentant la quantité à produire par produit et par critère

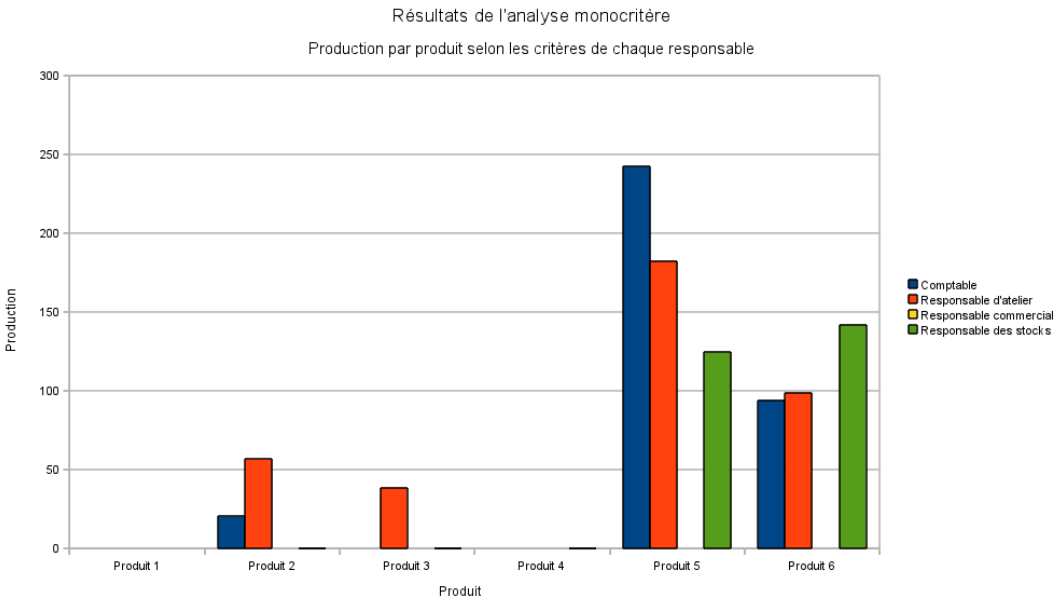

FIGURE 3 – Diagramme représentant la quantité à produire par produit et par critère

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 tel est le cas.

7 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.

8 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.

9 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 Pour élaborer la métrique de satisfaction pour le responsable des stocks nous opterons pour la fonction suivante, qui correspond à ce qui était précédemment annoncé dans la partie 1 (page 11).

Cette fonction est décrite par l'expression suivante :

$$M_{Stocks} = \begin{cases} \frac{x}{1192} \text{ si } x \in [0\,;\,1192] \\ x = 1 \text{ si } x \in [1192; 1559] \\ \frac{-x}{1192} + 1 + \frac{1559}{1192} \text{ si } x \in [1559\,;\,1691] \end{cases}$$

Cette fonction pourrait être justifiée par le fait que plus on s'éloigne des valeurs admise moins le responsable des stocks est satisfait.

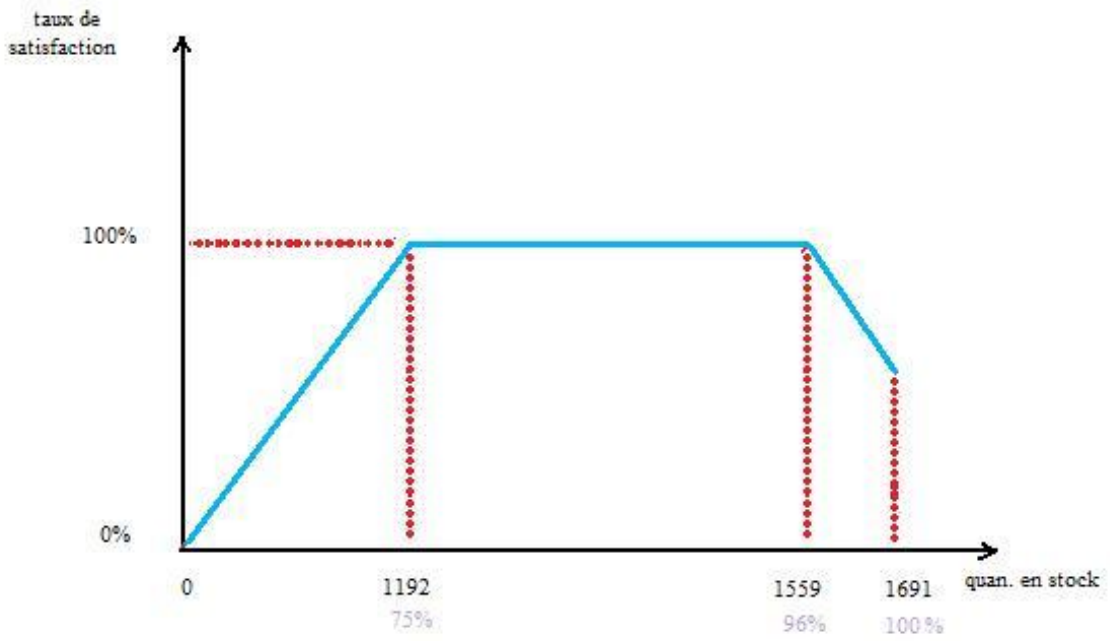

FIGURE 4 – Représentation graphique de la métrique pour le responsable des stocks.

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 :

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

10 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 96.5498% »

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
Comptable	100%	94.7678%	29.9574%	11.4302%
Atelier	96.5498%	100%	31.6113%	47.9244%
Stock	74.1546%	70.8003%	100%	25.2908%
Commercial	81.8654%	93.4330%	29.5354%	100%

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

11 Analyse multicritère

Des deux parties précedentes, emergent 8 propositions de gestion de l'atelier. Le but de cette partie sera de selectionner la meilleure solution en fonction de 4 critères.

11.1 Méthode choisie

La méthode de résolution choisie sera Electre III car elle englobe les 2 précedentes. On pourras ainsi fournir au client la méthode sélectionné comme la plus optimale ainsi qu'une ou plusieurs méthodes alternatives.

Pour réduire au maximum l'échéance, nous avons parallelisé au maximum les flux de travail. Nous avons donc dès le début du projet commencé par coder sous MatLab un algorithme de résolution de Electre 3.

11.2 Mise en œuvre de la méthode

L'algorithme très simple est identique à la méthode proposé en cours :

à partir de la matrice des jugements, il remet à l'echelle les notes des critères en fonction des poids.

for i = 1 :n J(:,i) = ((J(:,i) - (e/2)) * (w(i)/max(w))) + (e/2) ; end ;

Puis il calcule les matrices de concordance et de discordance.

for i = 1 :n for j = 1 :n if (i = j) C(i,j) = sum((J(i, :)¿=J(j, :)).*w)/sum(w) ; D(i,j) = max(max(J(j, :)-J(i, :)),0)/e ; end ; end ; end ;

Pour finir, il établit la matrice des surclassement en fonction de ces deux dernières matrices.

S = ((C ¿ s).*(D ¡= v)) ;

Un graph est ensuite généré à partir de cette matrice. On obtient ainsi rapidement la meilleurs solution en tête de graphe, et grace à la méthode du classement et du classement inverse, on peut obtenir l'ordre des solutions.

11.3 Solution proposé

Dans un premier temps, sans prendre en compte l'étude des 2 première parties, la meilleur solution, serait la solution A.

Dans un second temps, en prenant en compte les poids apporté par l'étude des 2 première parties.

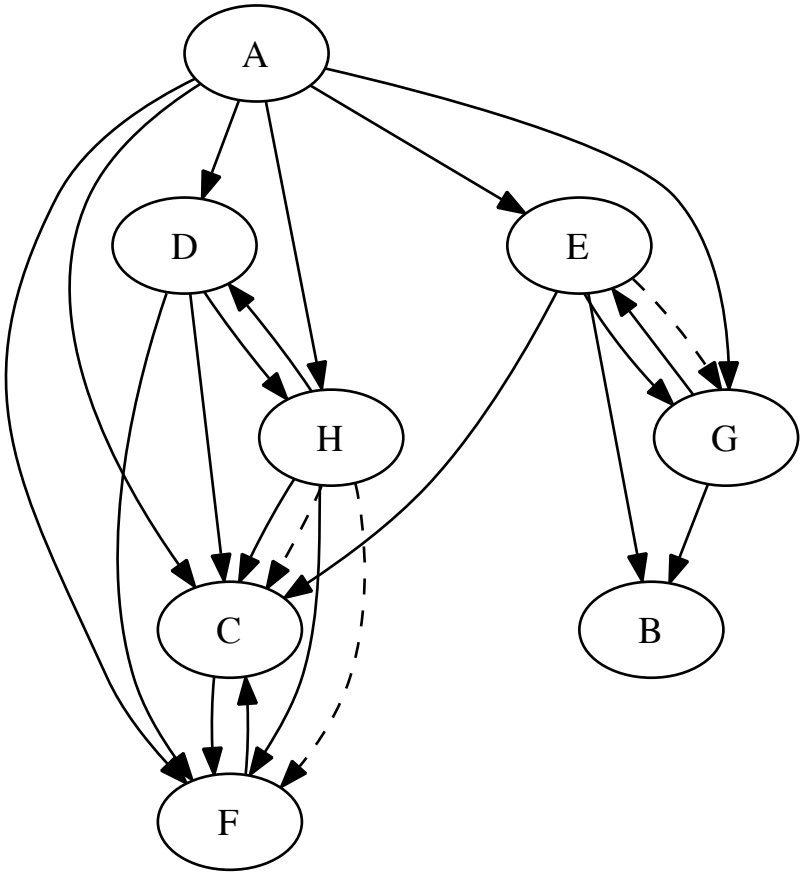

FIGURE 5 – Graphe des surclassement sans prise en compte des poids de chaque critère

Troisième partie

Annexe

12 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 ] = Atelier( )

FetchData;

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

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

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