REPUBLIQUE ALGERINNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L’ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVRSITE MOULOUD MAMMERI DE TIZI-OUZOU

FACULTE DES SCIENCES
DÉPARTEMENT DE MATHEMATIQUE

MEMOIRE DE FIN D’ÉTUDES
En vue de l’obtention de
MASTER 2 PROFESSIONNEL
MATHEMATIQUES APPLIQUEES A LA GESTION

Thème

Optimisation d’un Problème de Coupe par la Programmation Linéaire.
Cas : Minimisation des chutes d’acier dans l’unité Froid (ENIEM)

Soutenu publiquement le Samedi 08 octobre 2016 devant le jury composé de :

Président : Mr BELHADJ A.

Examinatrice : Mme NOURI N.

Promoteur :

Dr. B. OUKACHA

Réalisé par :

Melle BEN BELKACEM Narimane

Mme MAMOU Razika Eps ZEMMOUCHE

2015-2016

Remerciements

En **premier lieu,** ***nous*** **remercions Dieu puissant de nous avoir donné la force et la volonté pour arriver à terme de ce travail qui représente le fruit de plusieurs années d'études.**

N**ous exprimons notre plus grande reconnaissance et nos plus vifs remerciements à notre promoteur : Dr OUKACHA Brahim pour son orientation, et son dévouement total à fin de concrétiser ma recherche et achever ce modeste travail.**

N**ous remerciement le personnel de l'ENIEM, en particulier monsieur KABI Zidane qui m'a aidée durant mon stage pratique et sans oublier monsieur BOUMRAH Djamel.**

N**ous remerciement chaleureusement nos professeurs pour la qualité de leurs enseignement.**

E**t en fin, nos salutations très distinguées pour nos membres de jury et nos remerciements pour avoir accepter de juger notre travail.**

DEDICACES

Je dédie ce modeste travail au bon dieu qui ma donné le courage de le réaliser.

A mon fils Mohamed akli (moumouh) ainsi que mon mari Farid Ali ;

A ma mère et mon père

A mes deux belles mères et mon beau père

A mes frères Mohamed, M'hamed et Marzouk

A ma sœur Tassadit et mes belles sœur : Zhor, Hakima et Lina ainsi que Farida, Karima et Karima

A mes neveux : Ali, Mohamed amine, amine et Aymane

A mes nièces Imène, Yasmine - nada et Yasmine

A mon beau frère Mustapha et Hamid

A tous les familles Mamou, Zemmouche.

A ma binôme Narimane et sa famille

A tous le groupe master mathématique appliquée à la gestion.

A tous mes amis.

RAZIKA

DEDICACES

Je dédie ce modeste travail au bon dieu qui ma donné le courage de le réaliser.

A mes très chers parents qui mon beaucoup soutenus

A mes frères Sofiane et Aguellidh

A mes grands parents

A toute la famille

A tout mes amis

NARIMANE

SOMMAIRE

Tables des matières

<u>Chapitre II : Généralités sur la programmation linéaire</u>

INTRODUCTION GENERALE

La recherche opérationnelle (RO), appelée aussi aide à la décision définie comme l'ensemble des méthodes et techniques rationnelles d'analyse et de synthèse des phénomènes des managements du système d'information utilisable pour élaborer de meilleurs décisions.

La RO est un domaine qui a pris son essor au cour de la seconde guerre mondiale, lorsque l'Etat – major Britannique fit l'appel à des équipes de mathématiciens et de physiciens pour analyser divers problèmes de nature militaire (développement d'un réseau de radar, organisation de convois maritimes…). Après la guerre. Les techniques se sont considérablement développées, grâce, notamment, à l'explosion des capacités de calculs des ordinateurs. Les domaines d'application se sont également multipliés.

Une des parties essentielles de la recherche opérationnelle est la programmation linéaire, qui étudie l'optimisation d'une fonction objectif linéaire soumise à des contraintes linéaire, elle est résolue par l'américain G.B. DANTZING en 1957.

La programmation linéaire est l'un des principaux utile de modélisation en recherche opérationnelle. C'est aussi la source des principale méthodes avancées plusieurs méthodes de résolution, la méthode graphique, l'algorithme de simplexe, la dualité ...

Dans notre travail, nous nous intéressons à la modélisation et la résolution des problèmes de programmation linéaire avec l'algorithme de simplexe (problème de découpe).

Les problèmes de découpes occupent une place importante en Recherche Opérationnelle. Ceux sont généralement des problèmes considérés comme difficiles, et souvent de grande dimension. Ils sont par conséquent des sujets de recherche de premier ordre, où l'on retrouve beaucoup de méthodes abordées dans d'autres domaines.

Notre travail est divisé en 4 chapitres.

Le premier présente l'organisme d'accueil.

Le deuxième chapitre est consacré à des généralités sur la programmation linéaire.

Le troisième chapitre sur la l'optimisation à l'aide de l'algorithme de simplexe.

Le dernier chapitre c'est la résolution avec visuel Xpress.

Chapitre I : Etude préalable

Introduction

Dans le cadre de la préparation à l'intégration au milieu professionnel, ce stage à était lancé.

Il s'agit d'un projet de fin d'étude pour l'obtention du Master Professionnel en Mathématique Appliquée à la Gestion. Ce projet s'est déroulé au sein d'entreprise nationale ENIEM (Enterprise Nationale des Industries de Electroménager).

Dans ce chapitre nous mettons notre travail dans son contexte. En première lieu nous présentons une vue globale sur l'organisme d'accueil, puis on explore en détail notre champ d'étude afin de prendre connaissance approfondi de ces fonctionnalités, recenser ses problématiques et de répondre aux besoins des utilisateurs. En fin nous parlerons du travail demandé en proposant des solutions.

Section 1 : Présentation de l'organisme d'accueil

1.1 Situation géographique :

L'Entreprise ENIEM (Enterprise Nationale des Industries de Electroménagers) se trouve au sein de la zone industrielle AISSAT - IDIR, OUED - AISSI à 10 Km de TIZI - OUZOU, elle s'étale sur une surface totale de 55 Hectares, sa direction générale se trouve au Chef-lieu de TIZI - OUZOU à proximité de la gare ferroviaire.

1.2 Historique d'ENIEM :

ENIEM résulte d'un contrat "produit en main" établi dans le cadre du premier plan quadriennal, et signé le 21 Août 1971 avec un groupe d'Entreprises allemandes représentées par le chef de file D.I.A.G (Société allemande) pour une valeur de 400 millions de dinars, les travaux de Génie Civil ont été entamés en 1972 et la réception des bâtiments avec tous les équipements nécessaires a eu lieu en juin 1977.

L'ENIEM est issue de la restructuration organique de la société nationale de fabrication et de montage électrique (SONELEC) fondée en 1977, elle a été créée par le décret N°83/19 et rendu opérationnel depuis le 02 janvier 1983, donc c'est une entreprise au statut de la société nationale.

En octobre 1989, l'ENIEM est passée à l'autonomie et dénommée ENIEM.ENIEM a été chargé de la production et de commercialisation des produits électroménagers.

1.3 Organigramme actuel de l'entreprise

L'ENIEM est organisé en plusieurs directions administré par une direction générale. La direction d'exploitation est subdivisée en plusieurs unités et chaque unité est compartimentée en départements dont finance et comptabilité.

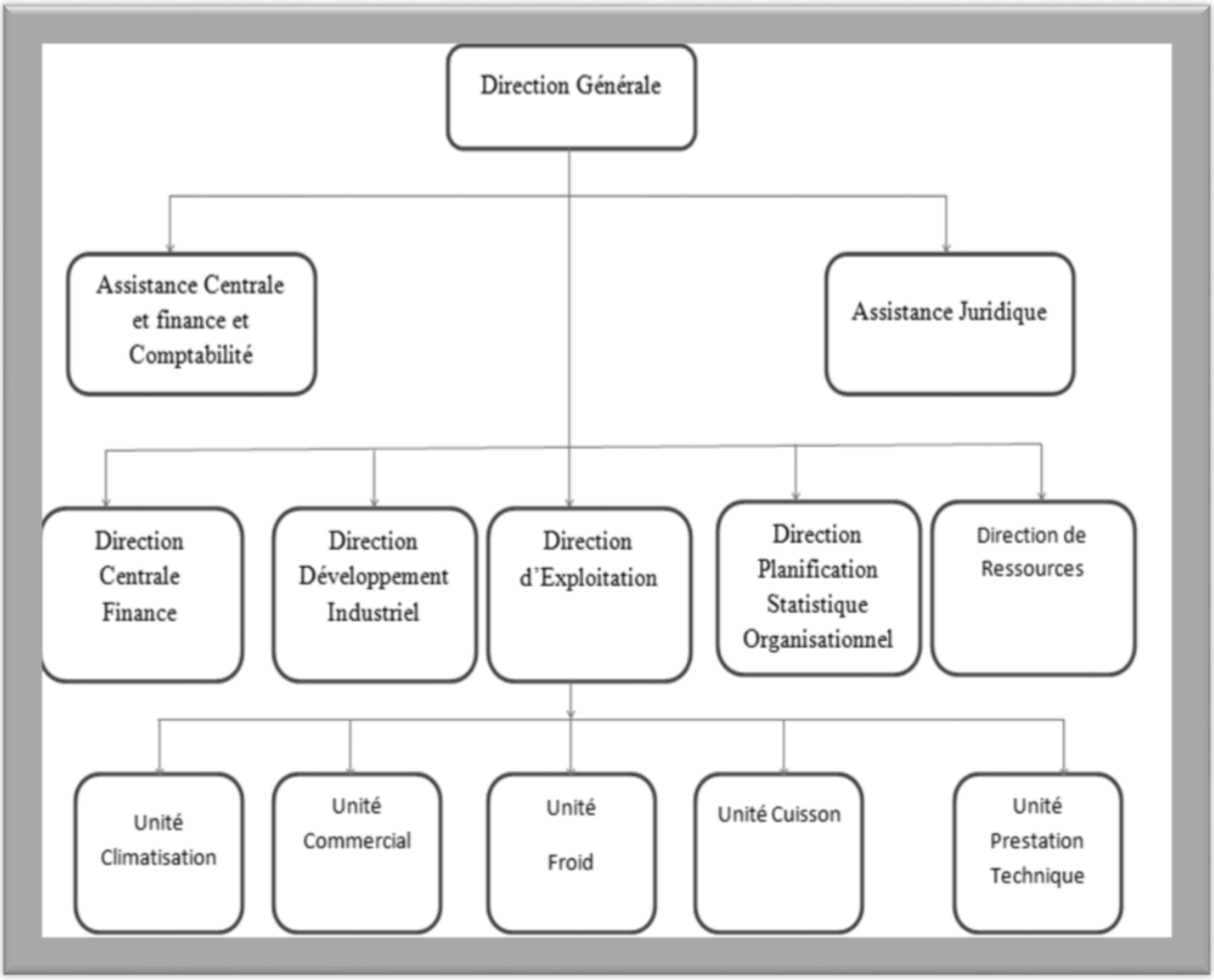


Figure 1: Organigramme général de l'entreprise ENIEM

Section 2 : Les activités de l'entreprise

Les activités de l'ENIEM sont concentrées sur la fabrication de réfrigérateurs, cuisinières, et climatiseurs. Ces activités sont assurées par plusieurs unités de productions :

2.1 Unité Froid :

Elle est composée de 3 lignes de production.

- **Une ligne de réfrigérateurs petits modèles :**

Les capacités installées sont de 110.000 réfrigérateurs / an, dont les modèles fabriqués sous licence BOSCH –Allemagne - 1977. ont :

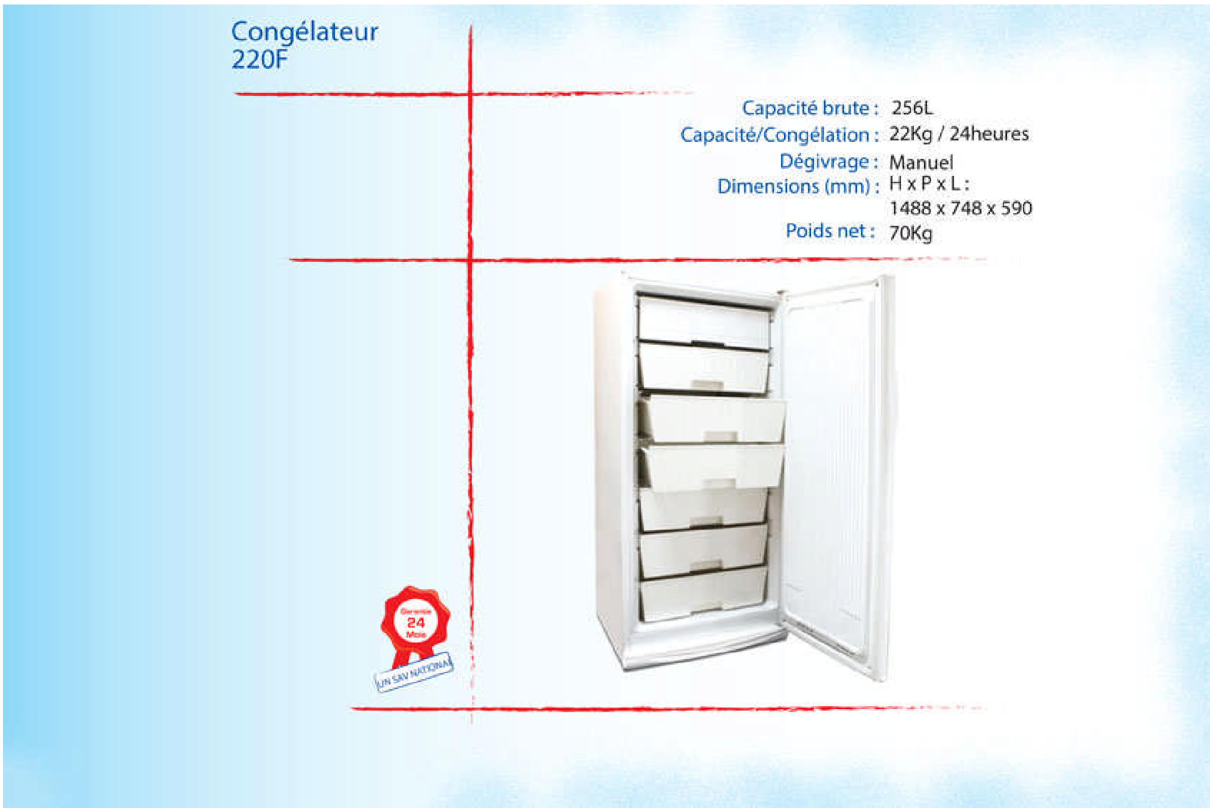
✓ ***160 l - 1 porte***

✓ ***240 l - 1 porte***

- **Une ligne de réfrigérateurs grands modèles :**

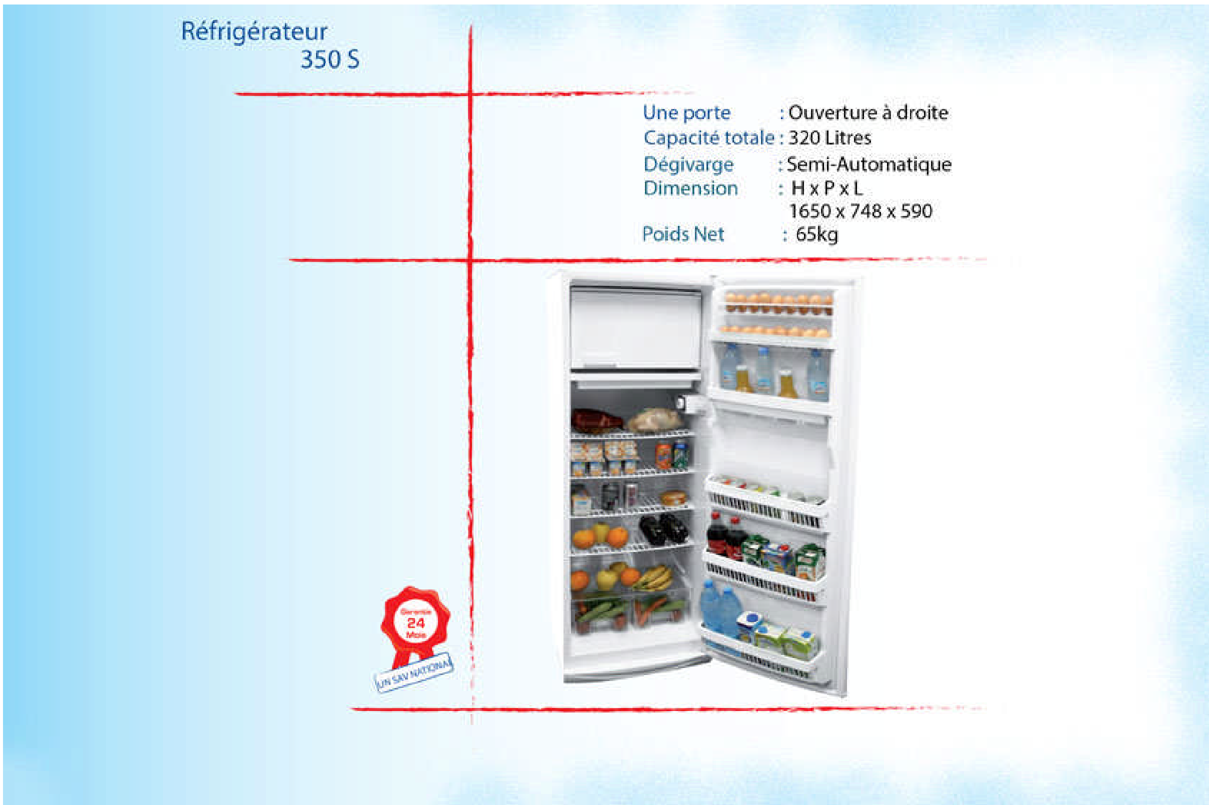
Les capacités installées sont de 390.000 Réfrigérateurs par an dont les modèles fabriqués sous licence TOSHIBA - JAPON – 1987. Sont :

✓ ***Congélateur vertical 220 F- 1 porte***



✓ ***Réfrigérateurs NO FROST HD 520 w***

✓ ***Réfrigérateur vertical 350 S -1 porte***



✓ ***Réfrigérateur SBS FRS U20 GA***

✓ ***Réfrigérateur 20 L ADE***

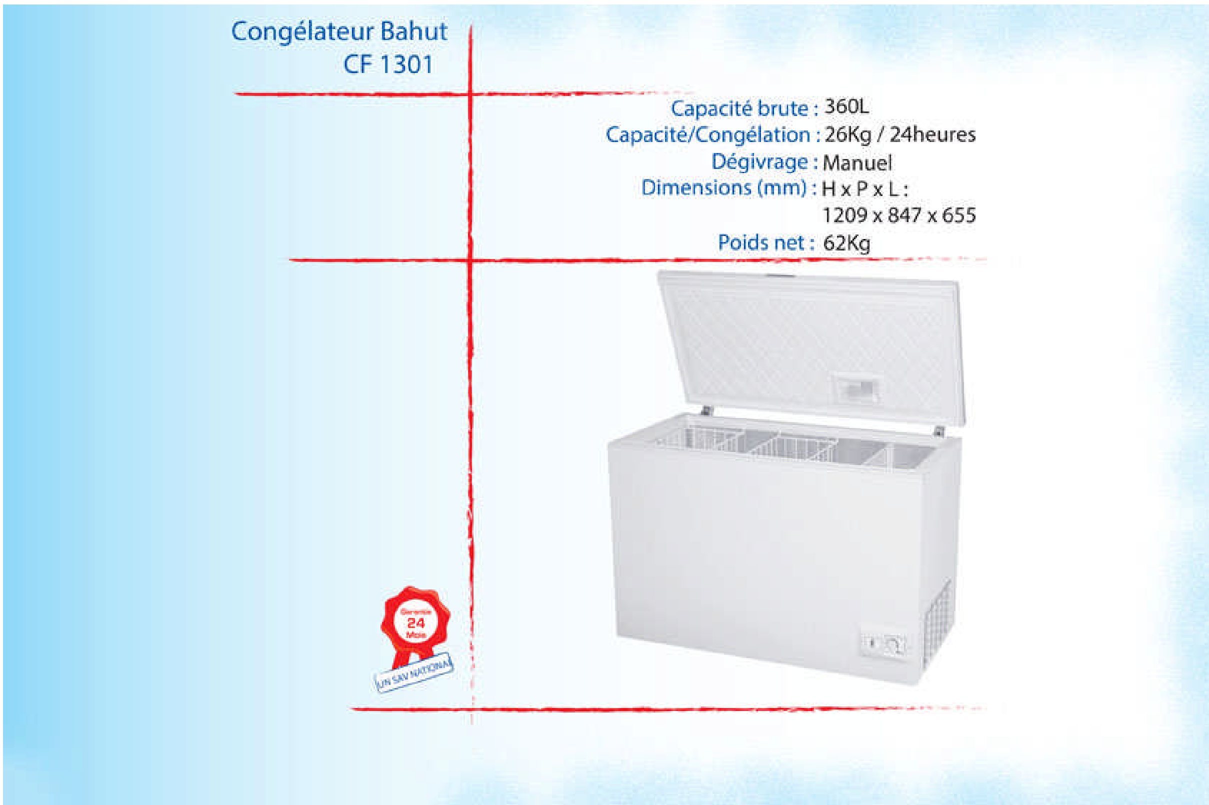
- **Une ligne de congélateurs bahut et réfrigérateurs de 520 L :**

Les capacités installées sont de 60.000 appareils par an. Dont les modèles sous licence LEMATIC-Liban- 1993 sont:

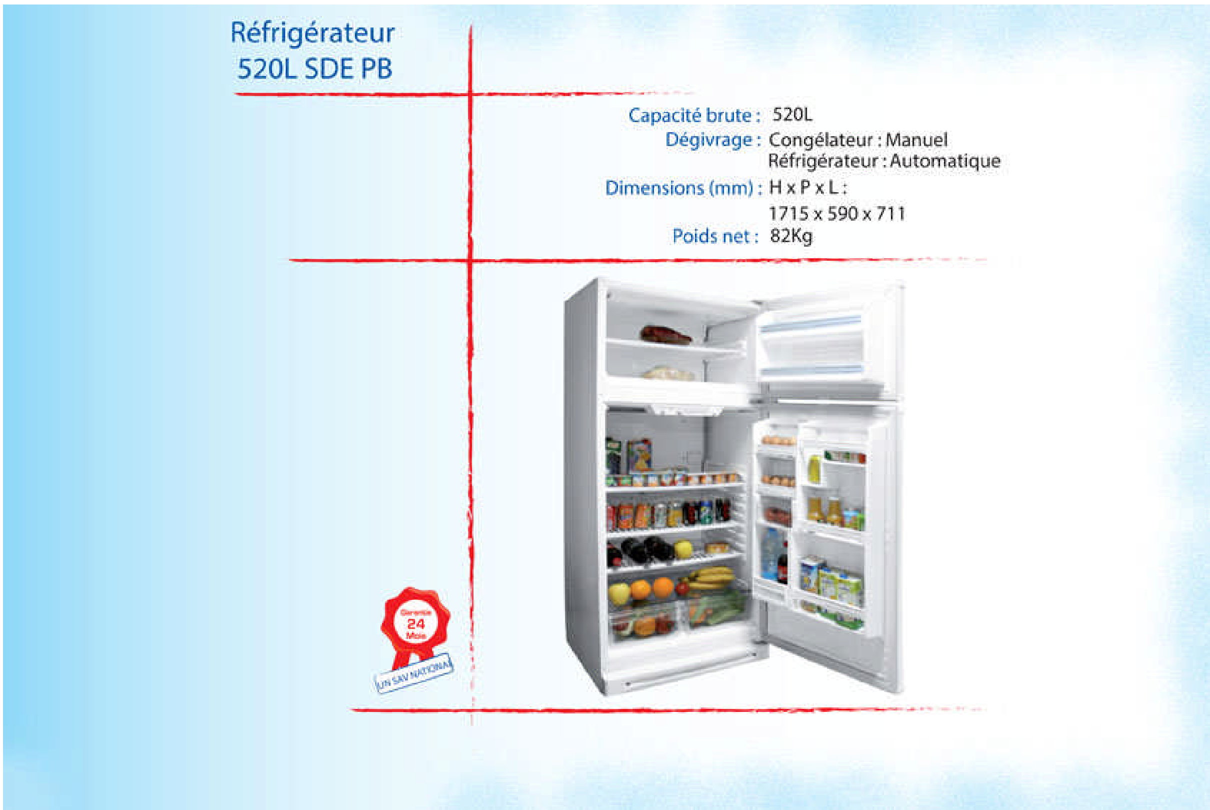
✓ ***Conservateur 420 TV***

✓ ***Bahut 500 D***

✓ ***Congélateurs bahut CF 1301***



✓ *Réfrigérateurs 520L SDE PB - 2 portes*



✓ *Réfrigérateur SBS HC 666 WE*

2.2 Unité Cuisson :

Elle assure la production des cuisinières, et les capacités installées sont de 150000 cuisinières par an, fabriquées sous licence TECHNO GAZ- Italie – 1991, dont les modèles sont :

- ✓ ***Cuisinière tout Gaz 8210 ,5 feux***

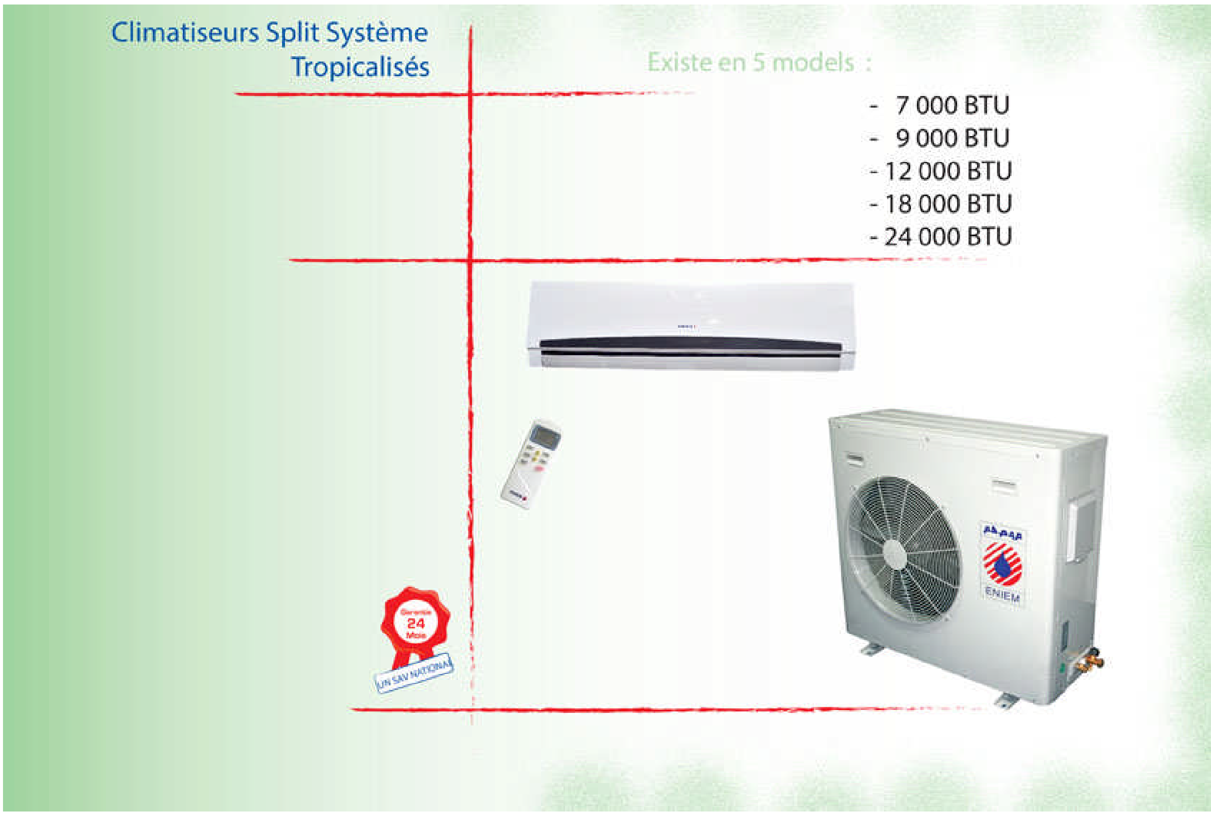
- ✓ ***Cuisinière tout gaz 6540 ,4 feux***

- ✓ ***Cuisinières tout gaz 6520 ,4 feux***

2.3 Unité Climatisation :

Les capacités existantes sont de 60.000 climatiseurs sous Licence AIWELL - France 1977 dont les modèles sont :

- ✓ ***Climatiseurs Type fenêtre - 9000, 12000, et 15000 BTU/h***
- ✓ ***Climatiseurs Split système tropicalisés***



- ***Armoire de climatisation***

- ***Climatiseurs Split système S430 - 14950 BTU/h***
- ***Climatiseurs Split système S530 - 18000 BTU/h***

✓ *Machine à laver*

✓ *Chauffe-eau 10 litre /3P*

2.4 Unité Prestations Techniques :

Cette unité assure les fonctions de soutien aux unités de production dans les domaines de :

- ✓ Réparation des outils et moules.
- ✓ Fabrication de pièces de rechange mécanique.
- ✓ Conception et réalisation d'outillages.
- ✓ Gestion des énergies et fluides.
- ✓ Gardiennage et sécurité.
- ✓ Travaux d'imprimerie.
- ✓ Travaux de menuiserie.
- ✓ Travaux de nettoyage.
- ✓ Prestation informatique: l’informatisation des différents services, exploitations et maintenance des équipements informatiques, formation des personnels administratifs.

Eu plus de ces quatre unité chargées de la production (Unités de CAM) on trouve une cinquième unité qui est :

2.5 Unité Commerciale :

Ses activités sont :

- ✓ La distribution et l exportation des produits ENIEM,
- ✓ Le service après-vente (à travers ses moyens propres et un réseau d agents agréés).

2.6 Mission et objectifs de l’entreprise

2.6.1 Missions :

La mission de L’ENIEM est d’assurer la production, le montage, la commercialisation, le développement et la recherche dans les différentes branches de l’électroménager notamment :

- Les appareils de cuisson par unité cuisson.

- Les appareils de climatisation par l'unité climatisation.
- Les produits sanitaires par l'unité d'AIN DEFLA.

2.6.2 Objectifs :

- ✓ L'amélioration de la qualité des produits.
- ✓ Meilleure maîtrise des coûts de production.
- ✓ L'augmentation des capacités d'études et de développement.
- ✓ L'amélioration de la maintenance de l'outil de production des installations.
- ✓ La valorisation des ressources humaines.
- ✓ L'augmentation des taux d'intégration (Interne et Externe).
- ✓ L'augmentation du volume de production.

- **Le règlement intérieur qui englobe :**

L'organisation générale de travail (horaires de travail, et de sortie, la tenue de travail, le contrôle de présence, etc.……) l'hygiène, sécurité et médecine du travail.

Ce présent règlement intérieur a pour but :

- De contribué à l'amélioration de la production et de la productivité.
- De fixer les principes et les règles relatifs à l organisation technique de travail, ainsi que celles relatives à l'hygiène, la sécurité, la discipline et la médecine du travail.

2.7 Les activités et Mission de la Direction Générale et de chaque Unité

2.7.1 La Direction Générale :

- **L'activité :**

Elle exerce son autorité hiérarchique et fonctionnelle sur l'ensemble des directions et des unités.

- **La mission :**

La direction générale est responsable de la stratégie et du développement de l'entreprise.

2.7.2 Les unités :

Unité	Activités	Missions
Froid	• Transformation de la tôle ; • Traitement de revêtement de surface (peinture, plastification) ; • Injection plastique et polystyrène ; • Fabrication de pièce métallique (condenseur, évaporateur) ; • Isolation ; • Thermoformage ; • Assemblage.	La mission globale de l'unité est de fabriquer, assembler et développer les produits de froid domestique.
Cuisson	• Transformation de tôle ; • Traitement de revêtement de surface (émaillage, zingage, chromage) ; • Assemblage.	La mission globale de l'unité est de fabriquer, assembler et développer les produits de technologie similaire.
Climatisation	• Transformation de tôle ; • Traitement de revêtement de surface (peinture) ;	La mission globale de l'unité est de fabriquer, assembler

	• Assemblage.	et développer les produits de technologie similaire.
Commerciale	• Marketing ; • Vent ; • Service après-vente ; • Gestion de stock de produit finis	Cette unité est chargée de la commercialisation des produits de l'entreprise et du service après vent

Prestation Technique	• Conception et réalisation des outils / moules ; • Réalisation (usinage) de diverse pièce de rechange ; • Étalonnage/vérification des Instruments de mesure ; • Impression ; • Production d'énergie et de fluides ; • Entretien des bâtiments ; • Fabrication de palettes (menuiserie); • Neutralisation des rejets industriels avant évacuation vers l'ouest ; • Transport marchandise ; • Surveillance du site ; • Prestation sociale.	L'unité est chargée de fournir de la prestation technique est de services nécessaire aux unités de production

Tableau 1 : Les activités et Missions de la Direction Générale et des différents Unités

En plus des unités l'ENIEM comporte deux filiales :

- **Filiale EIMS** : Qui a pour mission de produire et de développer les produits sanitaire (baignoire, lavabo et évier).
- **Filiale FILAMP** : L'unité lampes de Mohammedia (ULM) qui a démarré en février 1979 pour fabriquer les lampes d'éclairage domestique ainsi que des

lampes de réfrigérateurs, est devenue filiale à 100% ENIEM, le premier janvier 1997.

Position du problème

Durant notre passage dans les différents services de l'entreprise nous avons remarqué l'existence de grandes quantités de chutes résultant des opérations de découpe et pour cela on doit essayer de minimiser au maximum le nombre de chutes produites lors de l'opération de découpe.

Conclusion

Dans ce chapitre nous avons pris connaissance de l'organisme d'accueil : ses moyens, ses missions et ses objectifs ce que nous a permis de cerner la problématique et les attentes des futures utilisateurs de notre application.

Chapitre II : Généralités sur la programmation linéaire

Introduction

La programmation linéaire est l'une des acquisitions la plus importantes de la théorie économique après la deuxième guerre mondiale. Elle s'est développée très rapidement, grâce aux efforts conjugués des mathématiciens, des chefs d'entreprise, des chefs militaires et des économistes.

Dans le présent chapitre on va faire le point sur la littérature de la programmation linéaire et son aspect mathématique.

Section 1 : Notions de base

1.1 Définition d'un programme linéaire :

Selon **William J.BAUMAUL : la programmation linéaire est une technique mathématique d'optimisation (maximisation et minimisation) de fonction objectif linéaire sous des contraintes ayant la forme d'inéquations linéaire car elle vise à sélectionner parmi différents actions, celle qui atteindra le plus probablement l'objectif visé.**

Selon **Robert DORMAN et Paul SAMUELSON** ajoutent que **la programmation linéaire est une méthode de détermination de meilleur plan d'action pour réaliser des objectifs données dans une situation où les ressources sont limitées.**

Donc la programmation linéaire peut se définir comme un outil mathématique qui permet d'analyser divers types de situation dans lesquelles nous retrouvons une fonction linéaire d'un certain désire à optimiser c'est-à-dire minimiser où maximiser.

1.2 Les formes d'un programme linéaire :

1.2.1 Forme générale d'un programme linéaire :

Plus généralement, on appelle programme mathématique un problème d'optimisation d'une fonction objectif de plusieurs variables en présence des contraintes. Le programme est dit linéaire si la fonction et les contraintes sont toutes des combinaisons linéaires de variables. Il a la forme générique suivante :

Il comporte n variables non négatives (3), m contraintes d'égalité ou d'inégalité (2), et la fonction objectif à optimiser (1). Le coefficient de coût ou de profit de la variable x_j est noté c_j , celui de la variable x_j dans la contrainte i et noté a_{ij}. La contrainte i a un second membre constant b_i. les contraintes simple de positivité ne sont pas incluses dans les m contraintes, car elle sont gérées à part par les algorithmes.

(1) Max ou Min $Z = \sum_{i=1}^{n} c_j x_j$

(2) $\forall\, i = 1,\ldots,m : \leq , \sum_{j=1}^{n} a_{ij} x_j = \text{ ou } \geq b_i$

(3) $\forall\, j = 1\ldots n : x_j \geq 0$.

Des valeurs des variables qui vérifient toutes les contraintes forment une solution réalisable du programme linéaire. Une solution réalisable est optimale si aucune autre n'a un profit supérieur (dans le cas où $Z \longrightarrow$ Max) .

Si les variables sont astreintes à être entier, on a un programme linéaire en nombre entiers (PLNE). Un programme linéaire en 0-1 est un cas particulier de PLNE dont les variables ne peuvent prendre que deux valeurs 0-1 ; ces variables sont dites booléennes, binaires ou de décision. Un PL mixte comprend à la fois des variables continues et des variables entiers. Enfin, à partir du moment où une contrainte ou la fonction objectif n'est plus une combinaison linéaire de variables, on a affaire à un programme non linéaire (PNL).

1.2.2 Formes matricielles classiques :

Notons $\mathbf{x} = (x_1, x_2,\ldots\ldots,x_n)$ le vecteur des variables, $\mathbf{b} = (b_1, b_2,\ldots,b_m)$ celui des secondes membres des contraintes, $\mathbf{c} = (c_1,c_2,\ldots.,c_n)$ les coûts ou profits associés aux variables, et **A la matrice m×n** des $\mathbf{a_{ij}}$, on peut alors écrire PL sous forme matricielle**.** **Deux formes** sont courantes : **la forme canonique** avec des contraintes ≤, utilisée pour la résolution graphique, et la forme standard avec égalité, pour la résolution algébrique par des algorithmes.

Forme canonique	***Forme standard***
Max c.x	**Max c.x**
A.x = b	**A.x≤ ou ≥ b**
x ≥ 0	**x ≥ 0**

Ces formes ne servent qu'à simplifier les présentations théoriques. Dans la réalité, un PL peut comporter à la fois des égalités et des inégalités. On peut facilement convertir les formes mixtes en formes classiques. Ainsi, toute contrainte d'égalité peut être remplacée par des inégalités.

$$\sum_{j=1}^{n} a_{ij}x_i = b_i \iff \begin{cases} \sum_{j=1}^{n} a_{ij}x_i \leq b_i \\ -\sum_{j=1}^{n} a_{ij}x_i \leq -b_i \end{cases}$$

On peut convertir une inégalité en égalité en ajoutant ou soustrayant une variable d'écart $e_i \geq 0$, propre à chaque contrainte i. A l'optimum, pour une inégalité $\leq$ concernant la consommation d'une ressource i, cette variable indique la quantité inutilisée de la ressource.

$$\sum_{j=1}^{n} a_{ij}x_i \leq b_i \text{ et } e_i \geq 0 \iff \sum_{j=1}^{n} a_{ij}x_i + e_i = b_i.$$

D'autres conversions sont possibles. Ainsi, on peut passer d'une maximisation à une minimisation, car maximiser Z revient à minimiser –Z. il ne faut pas oublier de multiplier par trouvée par -1 la valeur de la fonction objectif la minimisation ! L'exigence des variables positives n'est pas restrictive, car une variables x_j non contrainte en signe peut toujours s'écrire comme une différence $x'_j - x''_j$ de deux variables non négatives.

1.2.3. Interprétation économique :

Il existe une interprétation et une terminologie suffisamment générale pour s'appliquer à la plupart des cas concrets. Nous reprenons ici un problème de minimisation et de maximisation.

- ***Minimisation d'un coût des obligations de fonctionnement.***

  Min $Z = \sum_{j=1}^{n} c_j x_j$

  $\sum_{j=1}^{n} a_{ij}x_i \leq b_i$,i = 1,….,m

  $x_j \geq 0$, j=1,…,n

a) **i** : bien produit

j : activité

$\mathbf{b_i}$: demande de bien i à satisfaire

$\mathbf{a_{ij}}$: taux de production de bien i par l'activité j

$\mathbf{c_j}$: coût unitaire de l'activité j

$\mathbf{x_j}$: niveau de l'activité j

b) **i** : activité

j : bien consommé (matières première)

b_i : niveau minimum de fonctionnement de l'activité i

a_{ij} : taux de fonctionnement de l'activité i pour la consommation de bien j

c_j : coût unitaire de bien j

x_j : quantité consommée de bien j.

➢ ***Maximisation d'un gain sous des contraintes de capacité***

$$\text{Min } Z = \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_{ij} x_i \leq b_i \text{ ,} i = 1,\ldots.,m$$

$$x_j \geq 0,\ j=1,\ldots,n$$

a) **i** : bien consommé

j : activité

b_i : quantité disponible de bien i.

a_{ij} : taux d'utilisation de bien i par l'activité j

c_j : gain unitaire de l'activité j

x_j : niveau de l'activité j

b) **i** : activité

j : bien produit

b_i : capacité maximum de fonctionnement de l'activité i

a_{ij} : taux de fonctionnement l'activité i pour la production de bien j

c_j : gain unitaire du bien j

x_j : quantité produite de bien j.

Il est important de mettre en évidence et de souligner les hypothèses sous – jacentes à l'obtention de ce modèle de programmation linéaire.

- Les activités j doivent pouvoir être considérées indépendamment, les unes des autres de sorte qu'il n'y ait aucune interaction entre elles, que ce soit au niveau de l'utilisation du bien ou du point de vue de gain engendré.

Cette hypothèse est indispensable pour obtenir l'additivité sur les activités j de la fonction économique et des contraintes.

- Le gain réalisé par l'activité j ainsi que l'utilisation d'un bien i par l'activité j, doivent être proportionnels à son niveau de fonctionnement, de sorte que seuls des termes du premier degré soient nécessaires pour exprimer le gain total (c_jx_j) ou l'utilisation totale du bien i par l'activité j ($a_{ij}x_j$).

Cette hypothèse de proportionnalité est indispensable pour obtenir le caractère linéaire de la fonction économique et des contraintes.

- Les variables doivent être divisibles, de sorte que les niveaux d'activité peuvent être mesurés par des nombres fractionnaires.

Cette hypothèse de divisibilité est indispensable pour utiliser des variables continues prenant des valeurs non nécessairement entières.

Section 2 : Méthodologie de modélisation en programmation linéaire

Nous voulons présenter une démarche qui permettra, dans la plupart des cas, de structurer sans trop de difficultés un modèle de programmation linéaire. Bien que la démarche est simple, la complexité de la modélisation provient du contexte même de la situation à modeler. Comme vous allez le constater ultérieurement, la résolution par les techniques appropriées sera l'étape la plus facile, une fois que le modèle est bien structuré.

Le schéma de la figure suivant résume les étapes à suivre dans le processus de modélisation :

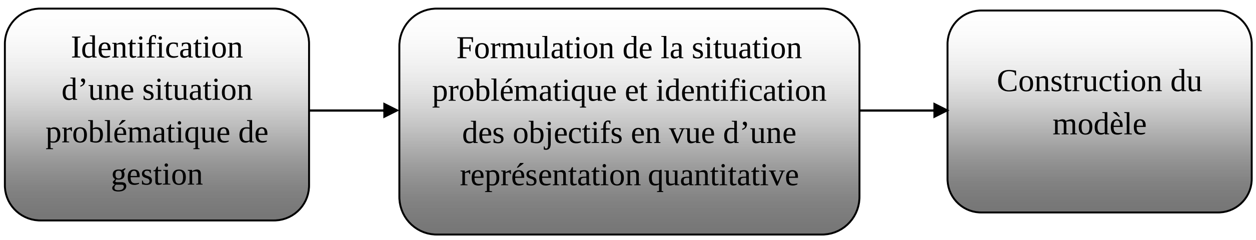
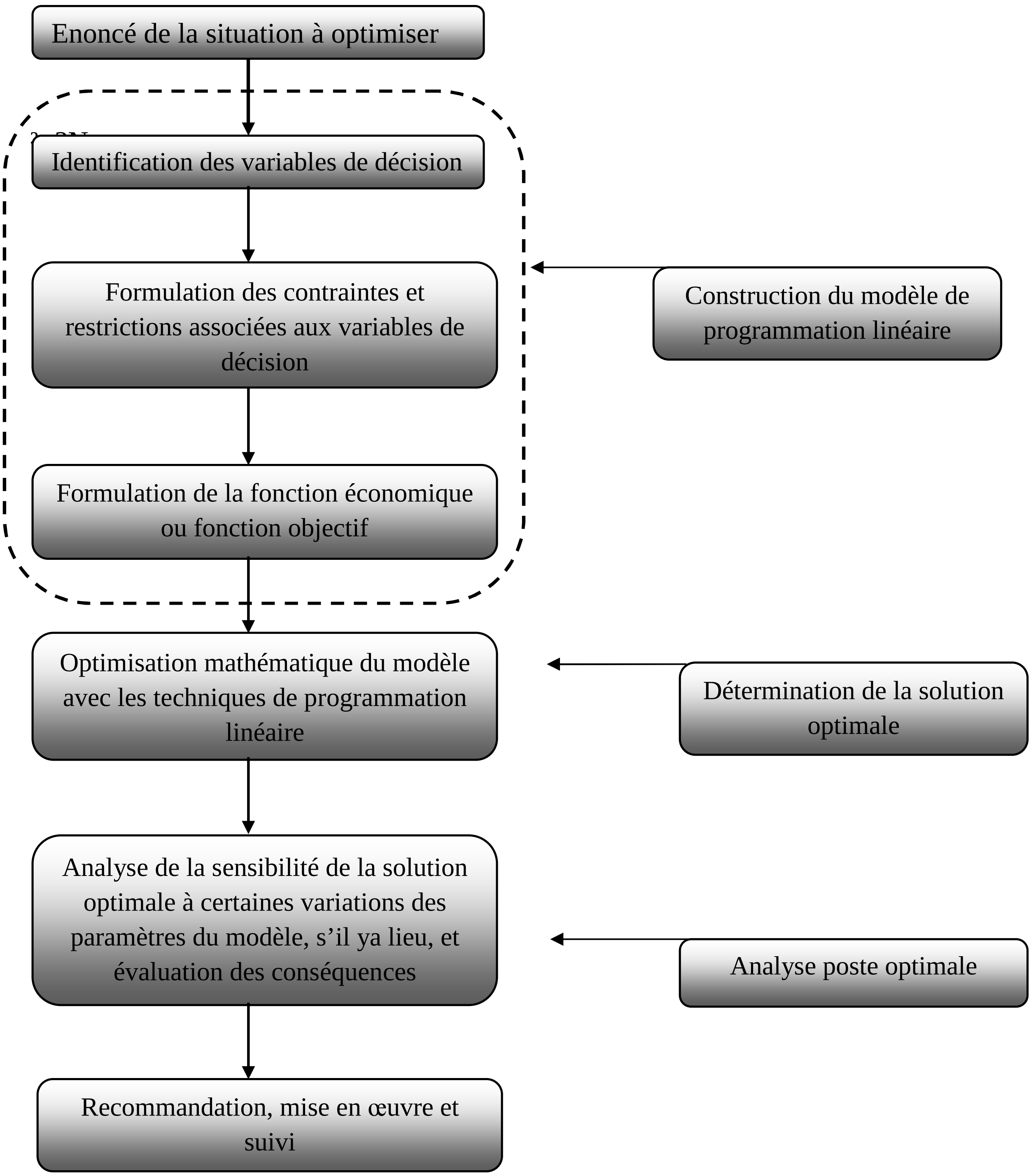


Figure I-1 : Etapes à suivre dans le processus de modélisation

Dans le cas où la situation que l'on veut analyser se prête à l'utilisation de la programmation linéaire comme outil d'aide à la décision, la démarche à suivre dans l'application de cette technique d'optimisation est résumé dans la figure suivante :



Comme l’indique ce schéma, la structure d’un modèle de programmation linéaire comporte trois éléments importants :

- Les variables de décisions ;
- Les contraintes linéaires ;
- La fonction économique.

- ***Eléments d'un modèle de programmation linéaire***

Variables de décision : la première étape dans le processus de modélisation est d'identifier correctement toutes les variables de décision (inconnues) de la situation à modeler. On peut se poser immédiatement la question suivante :

« Est ce que l'identification des variables de décision (en nombre et en description) va nous permettre, suite à la résolution du problème (avec les techniques appropriées), une prise de décision adéquate, compatible à l'aspect pratique de la situation ? »

Contraintes : *dans la problématique de la* situation, il faut être en mesure d'identifier tout genre de restriction (main d'œuvre, espace, budget,..) qui *peut limiter les valeurs que peuvent prendre les variables de décision. Existe-il également des restrictions ou exigences minimales sur les variables de décision (contraintes de marché, politiques de l'entreprise ?)*

A chaque restriction, limitation ou exigences correspond habituellement une contrainte qui prendra la forme d'une équation ou d'une inéquation linéaire.

L'ensemble des contraintes ainsi formulées constitue le domaine (région) des solutions possibles (valeurs possibles des variables de décision) au modèle de programmation linéaire.

Fonction objectif : *à chaque variable de décision qui a été identifié dans le modèle correspond un coefficient économique indiquant la contribution unitaire de la variable correspondante à l'objectif poursuivi. Par la suite, on pourra en déduire la fonction objectif que l'on veut optimiser (soit maximiser, soit minimiser).*

D'une façon générale, résoudre un problème de programmation linéaire consiste à déterminer les valeurs des variables de décision qui maximisent (ou minimisent selon le cas) une fonction économique (linéaire) soumise à un ensemble de contraintes (linéaires).

Section 3 : formulation mathématique d'un programme linéaire et exemples d'application

3.1. Formulation mathématique d'un programme linéaire

Avant d'aborder divers contexte d'application, précisons à quoi correspond la structure mathématique d'un modèle de programmation linéaire.

Le modèle mathématique de programmation linéaire est présenté habituellement en termes suivants :

Maximiser (ou minimiser selon l'objectif poursuivi) la fonction objectif

$Z = c_1x_1 + c_2x_2 + c_3x_3 + \ldots\ldots + c_nx_n$ (1-1)

Soumise aux contraintes linéaires

$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots\ldots + a_{1n}x_n \ (\leq , = , \geq)\ b_1$

$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots\ldots + a_{2n}x_n \ (\leq , = , \geq)\ b_2$

(1-2)

$a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 + \ldots\ldots + a_{in}x_n (\leq , = , \geq)\ b_i$

$a_{m1}x_1 + a_{m2}x_2 + \ldots\ldots + a_{mn}x_n \leq b_m$

et aux contraintes de non négativité

$x_1 \geq 0,\ \ x_2 \geq 0,\ \ldots\ldots\ \ x_n \geq 0$ (1-3)

Ces différents éléments du modèle ont la signification suivante :

Z représente la valeur de la fonction objectif (ou fonction économique). Cette quantité représente habituellement une valeur monétaire.

$x_1,\ x_2,\ \ldots\ldots\ x_n$, les variables de décision (inconnues) du modèle.

$c_1,\ c_2,\ \ldots\ldots\ c_n$, les coefficients des variables de a fonction objectif.

$a_{11},\ a_{12},\ \ldots\ldots\ a_{mn}$, les coefficients des variables de décision (appelés parfois coefficients technologiques) des divers contraintes. Les a_{ij} représentent la quantité de la ressource i requise par unité de x_j.

$b_1,\ b_2,\ \ldots\ldots\ b_m$ sont les seconds membres des contraintes et représentent fréquemment les quantités des divers ressources qui sont disponibles.

La notation $(\leq , = , \geq)$ qui est indiquée à la gauche de chaque b_i signifie que chaque contrainte possède l'un des trois signes mentionnés.

Optimiser un modèle de programmation linéaire, c'est déterminer les valeurs de diverses variables de décision x_j devant respecter les contraintes (1-2) et (1-3) qui maximisent (ou minimisent) la fonction objectif (1-1).

Remarques:

a) Les éléments a_{ij} b_i et c_j sont des quantités connues dans le modèle de programmation linéaire. Ils sont identifiés comme étant des paramètres du modèle qui permettent de mettre en relation les variables de décision aux contraintes et à la fonction objectif du modèle.

b) Chaque contrainte n'a qu'un seul des signes (≤, =, ≥) ; d'autre part, le signe de la contrainte peut varier d'une contrainte à l'autre. De plus, le modèle de programmation linéaire ne tient pas compte des différents types d'unités (heures, mètres, carrés, litres,…) qui peuvent exister entre les contraintes. Il est important toutefois qu'il y ait comptabilité d'unités au sein d'une même contrainte.

c) Les contraintes (1-2) s'appellent aussi **contraintes fonctionnelles** ou **contraintes technologiques** par opposition aux restrictions imposes sur les valeurs des variables de décision x_j par les contraintes de non négativité (1-3).

d) Dans la structure dans contraintes, on constate que chaque variable de décision x_j exige a_{ij} unités de la ressource i et que la somme utilisée de la ressource i par l'ensemble des variables de décision x_j donne l'utilisation effective de cette ressource i disponible en quantité $b_{i.}$

Donnons un premier exemple qui permet d'illustrer la construction d'un modèle de programmation linéaire.

3.2. Exemples d'application

3.2.1. Problème de production

Une usine produit deux ciments rapportant 50 DA et 70 DA /tonne. Pour faire une tonne de ciment 1, il faut 40min de calcination dans un four à chaud et 20 min de broyage. Pour fabriquer une tonne de ciment 2, il faut 30 min de four à chaud et 30min de broyage. Le four et l'atelier de broyage sont disponibles 360 min et 480 min par jour. La question que se pose l'exploitant de cette usine est la suivante :

Combien de ciment de chaque type peut-on produire par jour pour maximiser le bénéfice ?

***Résolution**

1- Notant par $x1, x2$ les quantités à produire des deux ciments. Ces quantités doivent vérifier les conditions suivantes le four ne peut fonctionner plus de 360min par jour et l'atelier pas plus de480min

$$40\,x1 + 30\,x2 \leq 360$$
$$30\,x1 + 30\,x2 \leq 480$$

2- Les quantités à produire de ciment sont toutes positives ou nulles : $x1 \geq 0$, $x2 \geq 0$.

Le chef de production du produit choisira le programme réalisable qui donnera le maximum de la fonction bénéfice Z : $Z(x1, x2)=50x1+ 70x2 \rightarrow max.$

En résumé le chef de production aura pour objectif, de trouver la solution optimale du problème suivant :

$$\mathbf{Z = 50x1 + 70x2 \rightarrow max}$$
$$\mathbf{40x1 + 30x2 \leq 360}$$
$$\mathbf{30x1 + 30x2 \leq 480}$$
$$\mathbf{x1, x2 \geq 0}$$

3.2.2 Problème de la coupe :

Un service de ravitaillement d'une usine à reçu d'un fournisseur 500 feuilles d'acier de longueur 5m .Celles-ci doivent être découpées en feuilles de 2m et 1.5 m désignées respectivement par *A* et *B.* Avec ces dernières feuilles on fabrique un produit dénote par *P* .Chaque produit *P* a besoin de 3 feuilles de *A* et 2 feuilles de *B.*

Le problème consiste à avoir un modèle mathématique en faisant un plan de découpage des feuilles, qui nous permet d'avoir une quantité maximale de produit P.

***Résolution :**

On peut avoir trois variantes de découpage représentées par le tableau suivant :

Variante de découpage	A	B	Reste
1	2	0	1m
2	1	2	0m
3	0	3	0.5m
Produit P	3	2	

La première variante de découpage par exemple consiste à découper une feuille de 5m en trois : (2m + 2m + 1m), c'est-à-dire 2 unités de *A* et un reste de 1m.
Soit $Z = x$ = nombre d'unités de *P*, et soit x_j = nombre de feuilles de 5m à couper par la $j^{ème}$ variante $(j=1\ldots3)$. De là, notre problème de découpe sera modélisé par :

$$
\begin{gathered}
Z = x \rightarrow \max \\
x_1 + x_2 + x_3 \leq 500 \\
2x_1 + x_2 \geq 3x \\
2x_1 + 3\,x_3 \geq 2x \\
x_j \geq 0,\ j=1\ldots3,\ x \geq 0
\end{gathered}
$$

Conclusion

Les problèmes d'application de la programmation linéaire sont, en pratique, constitués de plusieurs variables de décision dont la résolution nécessite l'utilisation de la méthode de simplexe ainsi qu'un logiciel permettant d'optimiser un modèle de programmation linéaire.

Chapitre III : L’optimisation à l’aide de l’algorithme du simplexe

Introduction

Dans le chapitre précédent, nous avons présenté divers concepts associés à la programmation linéaire. Parmi les méthodes de résolution citons la méthode graphique. Cette méthode permet d’optimiser un modèle entre le nombre de variantes et le nombre de contraintes est ≤ 2.

Néanmoins, la plupart des situations présentent un nombre important de variable de décision. Il nous faut donc une méthode d’optimisation d’un modèle de programmation linéaire qui peut s’appliquer efficacement quelque soit le nombre de variables dans le modèle. Pour ce faire, on utilise la **méthode du simplexe**.

L'algorithme du simplexe est un algorithme de résolution des problèmes d'optimisation linéaire. Il a été introduit par **George Dantzig** à partir de 1947. Il a beaucoup contribué au démarrage de l'optimisation numérique. L'algorithme du simplexe a longtemps été la méthode la plus utilisée pour résoudre les problèmes d'optimisation linéaire. Depuis les années 1985-90, il est concurrencé par les méthodes de points intérieurs, mais garde une place de choix dans certaines circonstances (en particulier si l'on a une idée des contraintes d'inégalité active en la solution).

Le nom de l'algorithme est dérivé de la notion de simplexe et a été suggéré par **Motzkin**. En réalité, l'algorithme n'utilise pas de simplexes, mais certaines interprétations de l'ensemble admissible du problème renvoient au concept de simplexe.

❖ ***Principe de l’algorithme de simplexe :***

La recherche systématique d’une solution optimale à l’aide de l’algorithme de simplexe peut se résumer comme suit :

1. Déterminer une première solution de base réalisable ; cette solution « initiale » sert comme point de départ vers la solution optimale si elle existe) ;
2. Si la solution obtenue en (1) n’est pas optimale, déterminer une autre solution de base réalisable qui permettrait d’améliorer la fonction objectif (augmentation pour une maximisation ou diminution pour une minimisation) ;

3. On répète cette procédure itérative jusqu'à ce qu'il ne soit plus possible d'améliorer la fonction objectif. La dernière solution de base réalisable obtenue constitue la solution optimale du programme linéaire.

L'algorithme nous permettra, avec divers critères, de détecter si la solution optimale est unique ou s'il existe des solutions optimales multiples ou encore si le modèle ne permet pas d'obtenir une valeur finie pour la fonction objectif.

Section 1 : Notions fondamentales associées à l'algorithme du simplexe

Considérons, le programme linéaire comportant m contraintes et n variables (variables de décision et variables d'écart). Le modèle s'écrit alors :

Maximiser (ou minimiser) $Z = c_1x_1 + c_2x_2 + c_3x_3 + \ldots\ldots + c_nx_n$

Soumise aux contraintes linéaires

$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots\ldots + a_{1n}x_n = b_1$

$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots\ldots + a_{2n}x_n = b_2$

.
.
.

$a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 + \ldots\ldots + a_{in}x_n = b_i$

.
.
.

$a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 \ldots\ldots + a_{mn}x_n = b_m$

Ce système d'équations s'écrit, sous forme matricielle :

$$\text{Maximiser } Z = c'\,X$$

Soumise aux contraintes

$$AX = b$$

$$X \geq 0$$

Avec :

$$A=\begin{pmatrix} a_{11} & a_{12}\ldots\ldots\ldots\ldots & a_{1n} \\ a_{21} & a_{22}\ldots\ldots\ldots\ldots & a_{2n} \\ . & . & . \\ . & . & . \\ . & . & . \\ a_{i1} & a_{i2}\ldots\ldots\ldots\ldots & a_{in} \\ . & . & . \\ . & . & . \\ . & . & . \\ a_{m1} & a_{m2}\ldots\ldots\ldots\ldots & a_{mn} \end{pmatrix},\ c = (c_1, c_2, \ldots, c_n)$$

$$X=\begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{pmatrix},\ b = \begin{pmatrix} b_1 \\ b_2 \\ . \\ . \\ . \\ b_i \\ . \\ . \\ . \\ b_m \end{pmatrix}$$

Donnons maintenant quelques définitions de base.

1.1. Définitions : solution de base et solution de base réalisable

Considérons un système de m équations linéaires AX =b comportant n variables (m≤n).

Une solution de base *au système d'équations* ***AX = b****, s'obtient en égalant n – m variables à zéro et en résolvant le système pour les m variables restantes. On suppose ici que la solution au système de m équations à m équations à m* *inconnues est unique. Les n – m variables qui sont annulées sont dites* ***variables hors base*** *alors que les m variables restantes sont appelées* ***variables de base****.*

Un programme linéaire admet au plus :

$$\binom{n}{m} = \frac{n!}{m!(n-m)!} \ .$$

Lorsque la solution de base satisfait également les contraintes de non – négativité, la solution est alors appelée ***solution de base réalisable.***
Les solutions de base réalisables sont la caractérisation algébrique des points extrêmes de la région des solutions admissibles. Une solution de base réalisable n’a jamais plus de m variables positives.

1.2. Variables d’écart et forme standard d’un programme linéaire

La méthode de résolution que nous employons nécessite que les contraintes fonctionnelles du modèle soient exprimées sous forme d’équations linéaires au lieu d’inéquations ; cette transformation s’effectue facilement en introduisant dans le modèle de nouvelles variables appelées **variables d’écart**.

1.2.1. Transformation d’une inéquation de signe plus petit ou égal (≤)

On transforme une inéquation linéaire ayant un signe ≤ en une équation linéaire en **additionnant** une variable non négative dite **variable d’écart.**

Une contrainte de la forme

$a_{i1}x_1+a_{i2}x_2+\ldots\ldots+a_{in}x_n \leq b_i$

S’écrira désormais

$a_{i1}x_1+a_{i2}x_2+\ldots\ldots+a_{in}x_n + x_{n+1} = b_i$,où $x_{n+1}\geq 0$

$x_{n+1} = b_i - (a_{i1}x_1+a_{i2}x_2+\ldots\ldots+a_{in}x_n)$

Un modèle comportant m contraintes ayant la forme suivante

$A_{11}x_1+a_{12}x_2+\ldots\ldots+a_{1n}x_n \leq b_1$

$A_{21}x_1+a_{22}x_2+\ldots\ldots+a_{2n}x_n \leq b_2$

. .
. .

$A_{m1}x_1+a_{m2}x_2+\ldots\ldots+a_{mn}x_n \leq b_m$

S'écrit, en introduisant m variables d'écart toutes non négatives,

$$
\begin{array}{lll}
a_{i1}x_1+a_{i2}x_2+\ldots\ldots+a_{in}x_n + x_{n+1} & & = b_1 \\
a_{i1}x_1+a_{i2}x_2+\ldots\ldots+a_{in}x_n & + x_{n+2} & = b_2 \\
. & & . \\
. & & . \\
a_{i1}x_1+a_{i2}x_2+\ldots\ldots+a_{in}x_n & \quad + x_{n+m} & = b_m
\end{array}
$$

1.2.2. Transformation d'une inéquation de signe plus grand ou égal (≥)

Dans le cas d'une contrainte de signe ≥, on doit, pour la transformer en équation, soustraire une variable d'écart, également appelée dans ce cas, **variable d'excédent.**

Une contrainte de la forme

$a_{i1}x_1+a_{i2}x_2+\ldots\ldots+a_{in}x_n \leq b_i$

S'écrira désormais

$a_{i1}x_1+a_{i2}x_2+\ldots\ldots+a_{in}x_n - x_{n+1} = b_i$ où $x_{n+1}\geq 0$

$x_{n+1} = (a_{i1}x_1+a_{i2}x_2+\ldots\ldots+a_{in}x_n) - b_i$

S'écrit, en introduisant m variables d'écart toutes non négatives,

$$
\begin{array}{lll}
a_{i1}x_1+a_{i2}x_2+\ldots\ldots+a_{in}x_n - x_{n+1} & & = b_1 \\
a_{i1}x_1+a_{i2}x_2+\ldots\ldots+a_{in}x_n & - x_{n+2} & = b_2 \\
. & & . \\
. & & . \\
a_{i1}x_1+a_{i2}x_2+\ldots\ldots+a_{in}x_n & \quad - x_{n+m} & = b_m
\end{array}
$$

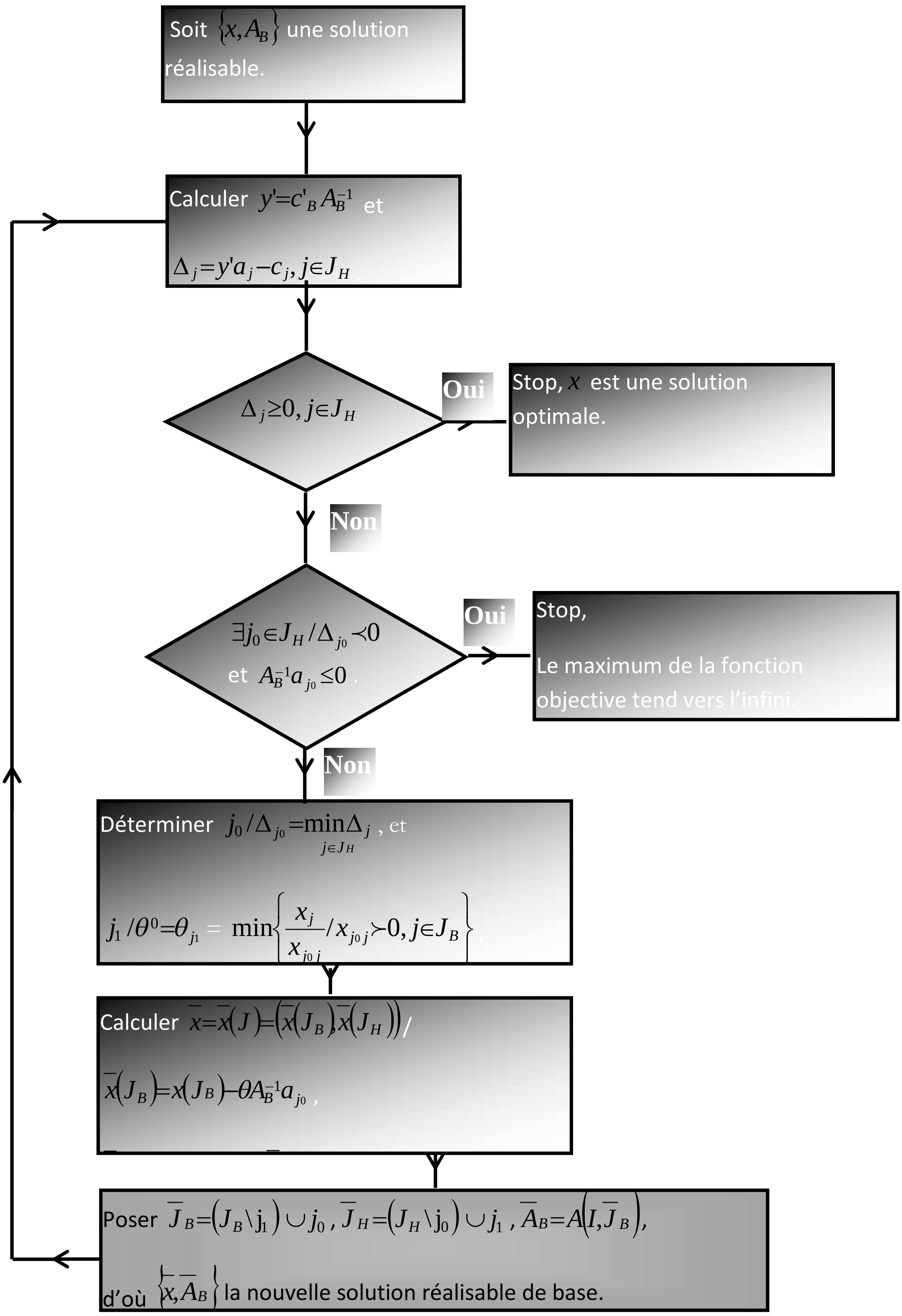


Figure 3.1 : Organigramme de l'algorithme de simplexe

Section 2 : Exemples d'application

2.1. Exemple 01 :

Prenons un programme linéaire, favori en forme standard avec ces deux contraintes d'égalité :

$$\mathbf{Max\ Z = x_1 + 2x_2}$$

$$\mathbf{x_1 + x_2 + x_3 \quad = 6}$$

$$\mathbf{x_2 \quad + x_4 = 3}$$

$$\mathbf{xj \geq 0, \forall\ j = 1, 2, 3, 4.}$$

L'algorithme du simplexe, construit une suite de solution de base réalisable (SBR) de profit croissant, jusqu'à ce qu'il n'y ait plus de gain possible. Géométriquement, il visite une suite de sommets adjacents du polyèdre. Le passage d'une base à l'autre va se faire par des opérations de pivotage. Le départ s'effectue de la base évidente (matrice identité) formé par les variables d'écart. La fonction objectif est ajustée de la même manière, mais au début il n'y a rien à faire puisque les variables d'écart ont un coefficient nul.

$$\mathbf{x_3 = 6 - x_1 - x_2}$$

$$\mathbf{x_4 = 3 - x_2}$$

La SBR initiale associée s'obtient en mettant les variables hors base des seconds membres à 0. Elle correspond au point 0 de la résolution géométrique. Pour changer de SBR, on imagine que les variables hors base sont nulles dans le système précédent. Une variable hors base, actuellement à 0, va être choisie pour entrer en base et augmenter jusqu'à annuler une variable de base. A la suite de cette opération de pivotage, la variable hors base qu'on augmente dite entrante, remplace celle qui s'annule dans la solution de base, dite sortante.

Pour déterminer la variable entrante, on regard les coefficients des variables hors base dans Z, appelés profits marginaux ou réduits (coûts marginaux en minimisation). Ils donnent le gain obtenu en augmentant de 1 la variable associée. En fait, l'algorithme converge si on augmente une des variables de profit marginal strictement positif, mais plus rapidement en moyenne si on choisit parmi elles celle de profit marginal maximal.

Voyons maintenant comment trouver la variable sortante. D'après la contrainte, x_2 ne peut pas augmenter au-delà de 6, sinon deviendrait négative .La seconde contrainte est encore plus limitant, car x_2 va aller jusqu'à 3 seulement, de ce faite, x_4 s'annule. La variable sortante est donc la première qui s'annule quand on augmente la variable entrante. Les nouvelles variables de base sont donc x_3 et x_2, et on doit les écrire, ainsi que Z, uniquement en fonction des nouvelles variables hors base x_1 et x_4. On obtient :

$\mathbf{x_2 = 3 - x_4}$

$\mathbf{x_3 = 6 - x_1 - x_2}$

$\mathbf{x_3 = 6 - x_1 - (3 - x_4) = 3 - x_1 + x_4}$

$\mathbf{Z = 0 + x_1 + 2(3 - x_4) = 6 - x_1 - x_2}$

La SBR actuelle se lit affectant un zéro aux variables hors base : **$x_2 = 3$, $x_3 = 6$, $x_1 = x_2 = 0$, avec un profit Z=6. Pour augmenter encore le profit, on peut seulement augmenter x_1, car c'est la seule variable hors base à avoir un profit unitaire strictement positif. x_1 peut augmenter jusqu'à 3 et remplace dans la base x_3, qui s'annule on écrit donc x_1, x_2 et Z en fonction de x_3 et x_4 :**

$\mathbf{X_1 = 3 - x_3 + x_4}$

$\mathbf{x_2 = 3 - x_4}$

$\mathbf{Z = 6 + (3 - x_3 + x_4) - 2x_4 = 9 - x_3 - x_4}$

A ce stade la SBR associée est x= (3, 3, 0,0) de profit 9. On est à l'optimum, car toutes les variables hors base ont un profit unitaire négatif : le profit diminuerait si on augmentait l'une d'entre elles.

Le processus est facilement adaptable au cas d'une minimisation. La seule différence est la règle pour la variable entrante : il faut prendre celle ayant le plus petit coût marginal strictement négatif (le plus grand en valeur absolue), le but d'obtenir un coût minimal.

2.2. Tableau simplexe :

Les différents calculs qu'on aura à effectuer dans les différentes étapes de résolution seront disposés dans le tableau suivant :

c			c_1	c_2	c_3	……	c_m	c_{m+1}	…..	c_j	……	c_n	
c_B	**Base**	b	a_1	a_2	a_3	……	a_m	a_{m+1}	…..	a_j	…..	a_n	θ_j
c_1	a_1	$b_1=x_1$	1	0	0	……	0	$x_{1,m+1}$	…	x_{1j}	…..	x_{1n}	θ_1
c_2	a_2	$b_2=x_2$	0	1	0	……	0	$x_{2,m+1}$	…..	x_{2j}	…..	x_{2n}	θ_2
c_3	a_3	$b_3=x_3$	0	0	1	……	0	$x_{3,m+1}$	…..	x_{3j}	…..	x_{3n}	θ_3
⋮ ⋮	⋮ ⋮	⋮ ⋮	⋮ ⋮	⋮ ⋮	⋮ ⋮		⋮ ⋮	⋮ ⋮		⋮ ⋮		⋮ ⋮	⋮ ⋮
c_m	a_m	$b_m=x_m$	0	0	0	……	1	$x_{m,m+1}$	…..	x_{mj}	…..	x_{mn}	θ_m
$Z=c'_B x_B$		Δ_j	$\Delta_1=0$	$\Delta_2=0$	$\Delta_3=0$	……	$\Delta_m=0$	Δ_{m+1}	…..	Δ_j	…..	Δ_m	

Nous allons résoudre le problème de programmation linéaire suivant, par la méthode du simplexe :

$$\begin{cases} Z = 2x_1 + x_2 \to \max, \\ x_1 + x_2 + x_3 = 4 \\ -2x_1 + 3x_2 + x_4 = 5 \\ 2x_1 - 3x_2 + x_5 = 6 \\ x_j \geq 0,\ j = \overline{1,5}. \end{cases}$$

On a $J=\{1,2,3,4,5\}$ et $J_B=\{3,4,5\}$, $J_H=\{1,2\}$ avec $A_B=I_3$, donc la solution réalisable de base est $x=(0,0,4,5,6)$, dressons alors ***le premier tableau du simplexe :***

c			2	1	0	0	0		
c_B	*Base*	b	a_1	a_2	a_3	a_4	a_5	θ_j	
0	a_3	4	1	1	1	0	0	4	L_1^1
0	a_4	5	-2	3	0	1	0	/	L_2^1
0	a_5	6	2	-3	0	0	1	3	$L_3^1 \rightarrow$vecteur sortant
Z=0		Δj	-2	-1	0	0	0		
			↑vecteur rentrant						

On remarque que la relation $\Delta_j \geq 0, \forall j \in J_H$, n'est pas vérifiée, donc la solution réalisable de base initiale n'est pas optimale, on doit alors changer la base de la manière suivante :

$\min\limits_{j \in J_H} \Delta_j = \Delta_1 = -2$, donc $j_0 = 1$, de là le vecteur a_1 va rentrer dans la nouvelle base, et calculons $\theta^0 = \min\limits_{j \in J_B} \theta_j$:

$\theta_3 = \frac{4}{1} = 4, \theta_5 = \frac{6}{2} = 3$, d'où $\theta^0 = \theta_{j_1} = \min\limits_{j \in J_B} \theta_j = \theta_5 = 3$, de là, le vecteur a_5 va sortir de la base, et la nouvelle base est $\overline{J}_B = \{3,4,1\}, \overline{J}_H = \{5,2\}$, Pour déterminer la nouvelle solution $\overline{x}$, dressons ***le 2ème tableau du simplexe :***

c			2	1	0	0	0		
c_B	*Base*	b	a_1	a_2	a_3	a_4	a_5	θ_j	
0	a_3	1	0	$5/2$	1	0	$-1/2$	$2/5$	$\rightarrow L_1^2 = L_1^1 - 1/2\, L_3^1$
0	a_4	11	0	0	0	1	1	/	$L_2^2 = L_2^1 + L_3^1$
2	a_5	3	1	$-3/2$	0	0	$1/2$	/	$L_3^2 = \frac{1}{2} L_3^1$
$\overline{Z}=6$		$\overline{\Delta}_j$	0	-4	0	0	1		
				↑					

La nouvelle solution de base est donc $\bar{x}=(3,0,1,11,0)$ de plus elle n'est pas optimale, car $\bar{\Delta}_2=-4\prec 0$.

On doit alors changer la base une autre fois :

$\min_{j\in \bar{J}_H}\bar{\Delta}_j=\bar{\Delta}_2=-4,$ donc le vecteur a_2 va rentrer dans la base.

Comme $\theta_{j_1}=\min_{j\in \bar{J}_B}\theta_j=\theta_3$, donc le vecteur a_3 sortira de la base.

D'où, on obtient : $\bar{\bar{J}}_B=\{2,4,1\},\bar{\bar{J}}_H=\{5,3\}$, pour déterminer la nouvelle solution $\bar{\bar{x}}$, dressons ***le 3ème tableau du simplexe :***

C			2	1	0	0	0		
c_B	Base	b	a_1	a_2	a_3	a_4	a_5	θ_j	
1	a_2	$2/5$	0	1	$2/5$	0	$2/5$		$\rightarrow L_1^3=\frac{5}{2}L_1^2$
0	a_4	11	0	0	0	1	1		$L_2^3=L_2^2$
2	a_5	$18/5$	1	0	$3/5$	0	$1/5$		$L_3^3=L_3^2+\frac{3}{5}L_1^2$
$\bar{\bar{Z}}=\frac{38}{5}$		$\bar{\bar{\Delta}}_j$	0	0	$\frac{8}{5}$	0	0		

L_2^3 par exemple, désigne la 2ème ligne du 3èm

La nouvelle solution de base est donc $\bar{\bar{x}}=\left(18/5,2/5,0,11,0\right)$, comme $\bar{\bar{\Delta}}_j\geq 0$, $\forall j\in \bar{\bar{J}}H$, l'algorithme s'arrête et la solution obtenue est optimale, avec $\bar{\bar{Z}}=\frac{38}{5}$

2.3. Initialisation de l'algorithme de simplexe

Considérons le problème de programmation linéaire suivant:

$$\begin{cases} Z = c'x \to \max, \\ Ax = b, \\ x \geq 0. \end{cases} \qquad \text{(P)}$$

2.3.1. La méthode des deux phases

A. Première phase :

La première phase de résolution du problème (P) consiste à déterminer une solution de base réalisable de (P). Pour cela, on construit la problème suivant :

$$\begin{cases} -\sum_{i=1}^{m} x_{n+i} \to \max, \\ [Ax]_i + x_{n+i} = b_i, \; i = \overline{1, m}, \\ x \geq 0, \; x_{n+i} \geq 0, \; i = \overline{1, m}; \end{cases} \qquad (P_1)$$

Où les x_{n+i} sont appelés des variables artificielles.

Le problème (P_1) possède n+m variables $X = (x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m})$ et m équations. Le vecteur $X = (0, \ldots, 0, b_1, \ldots, b_m)$ est réalisable pour (P_1) donc l'ensemble des solutions admissibles de (P_1) est non vide.

D'un autre côté la fonction objectif est bornée supérieurement :

$-\sum_{i=1}^{m} x_{n+i} \leq 0$, donc le problème ($P_1$) admet une solution optimale $X^0 = (x^0, x_a^0)$

Théorème 1:

Si x_a^0 est différent de zéro alors les contraintes du problème de départ (P) sont contradictoires.

Preuve :

Supposons que $x_a^0 \neq 0$ et les contraintes de (P) ne sont pas contradictoires, donc $\exists x^1 / Ax^1 = b$ et ceci implique que $(x^1, x_a^1 = 0)$ est une solution admissible de (P_1) et

(x^0, x_a^0) est optimale de (P_1), donc $-\sum_{i=1}^{m} x_{ai}^1 \leq -\sum_{i=1}^{m} x_{ai}^0 \Rightarrow 0 \leq -\sum_{i=1}^{m} x_{ai}^0$, et comme $x_a^0 > 0$, ce qui est impossible, donc il n'existe pas de $x^1 / Ax^1 = b$.

B. Deuxième phase

Soit (x_0, x_a^0) une solution optimale de (P_1) avec des variables artificielles nulles, alors on utilise la solution x_0 avec sa matrice A_B^0, comme solution de base de départ du problème (P), et ceci constitue la deuxième phase.

Si $x_{n+i}^0 = 0, \forall i$ et $\exists$ un indice $i_0 / a_{i_0} \in A_B$, alors pour revenir a la deuxième phase , il faut exclure cette colonne de A_B.

Exemple 1:

Considérons le problème suivant :

$$\begin{cases} Z = x_1 + 2x_2 + x_3 \rightarrow \max \\ -x_1 + x_2 - x_3 = 10 \\ 2x_1 - x_2 + x_3 = 5 \end{cases} \qquad (P)$$

$$x_j \geq 0,\ j = \overline{1,3}$$

Du problème (P) on construit le problème suivant :

$$\begin{cases} F = -x_4 - x_5 \rightarrow \max \\ -x_1 + x_2 - x_3 + x_4 = 10 \\ 2x_1 - x_2 + x_3 + x_5 = 5 \end{cases} \qquad (P_1)$$

$$x_j \geq 0,\ j = \overline{1,5}\ .$$

$X = (0, 0, 0, 10, 5)$ est une solution réalisable de base de (P_1), Avec

$A_B = (a_4, a_5) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

Dressons ***le premier tableau du simplexe :***

c			0	0	0	-1	-1		
c_B	Base	b	a_1	a_2	a_3	a_4	a_5	θ_j	
-1	a_4	10	-1	1	-1	1	0	/	L_1^1
-1	a_5	5	2	-1	1	0	1	$5/2$	$\rightarrow L_2^1$
F= -15		Δ_j	-1	0	0	0	0		
			$\uparrow$						

$\Delta_{j_0} = \min_{j \in J_H} \Delta_j = \Delta_1 = -1$, donc la solution de départ n'est pas optimale, et $j_0=1$ va rentrer dans la nouvelle base.

De plus $\theta_{j_1} = \min_{j \in J_B} \theta_j = \theta_5 = 5/2$, donc $j_1=5$ va sortir de la base, de là on obtient la nouvelle base $\bar{J}_B = \{4, 1\}$, $\bar{J}_H = \{5, 2, 3\}$.

Dressons ***le 2ème tableau du simplexe :***

c			0	0	0	-1	-1		
c_B	Base	b	a_1	a_2	a_3	a_4	a_5	θ_j	
-1	a_4	$25/2$	0	$1/2$	$-1/2$	1	$1/2$	25	$\rightarrow L_1^2 = L_1^1 + \frac{1}{2} L_2^1$
0	a_1	$5/2$	1	$-1/2$	$1/2$	0	$1/2$	/	$L_2^2 = \frac{1}{2} L_2^1$
$\bar{F} = 25/2$		$\bar{\Delta}_j$	0	$-1/2$	$1/2$	0	$1/2$		
				$\uparrow$					

La nouvelle solution est $\bar{x} = \left(5/2\ , 0\ , 0\ , 25/2\ ,\ 0\right)$, elle n'est pas optimale, car

$\bar{\Delta}_2 = {-1}/{2} < 0.$

$\bar{\Delta}_{j_0} = \min\limits_{j \in J_H} \bar{\Delta}_j = \bar{\Delta}_2 = {-1}/{2}$, donc $j_0 = 2$, va rentrer dans la base et

$\theta_{j1} = \min\limits_{j \in J_H} \theta_j = \theta_4$, de là $j_1 = 4$ va sortir de la base.

D'où $\bar{\bar{J}}_B = \{2, 1\}$, $\bar{\bar{J}}_H = \{5, 4, 3\}$.

Dressons ***le 3^e tableau du simplexe :***

C			0	0	0	-1	-1	
c_B	**Base**	b	a_1	a_2	a_3	a_4	a_5	θ_j
0	a_2	25	0	1	-1	2	1	
0	a_1	15	1	0	0	1	1	
$\bar{\bar{F}} = 0$		$\bar{\bar{\Delta}}_j$	0	0	0	1	1	

$\bar{\bar{\Delta}}_j \geq 0,\ \forall j \in J \Rightarrow x^0 = (15, 25, 0, 0, 0)$ est solution optimale de (P_1), comme $x_4 = x_5 = 0$, donc $x^0 = (15, 25, 0)$ est une solution de base admissible du problème (P), avec $A_B = (a_2, a_1)$.

On passe à la deuxième phase en utilisant cette dernière solution basique, pour résoudre le problème (P).

Ici on utilise le dernier tableau obtenu à la 1ère phase avec une solution optimale dont les variables artificielles sont nulles.

c			1	2	1	
c_B	**Base**	B	a_1	a_2	a_3	θ_j
2	a_2	25	0	1	-1	
1	a_1	15	1	0	0	
Z =65		Δ_j	0	0	-3	

Le vecteur $A^{-1}a_{j0} = A_B^{-1}a_3 = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, donc la solution du problème (P) est infinie, c'est à dire Z $\to \infty$.

2.3.2. La M-méthode :

L'Américain **TCHARNESS** a proposé une méthode en rassemblant les deux phases en une seule.

Du problème (P), on construit un problème (P_2) de la manière suivante :

$$\begin{cases} \overline{Z} = c'x - M\sum_{i=1}^{m} x_{n+i} \to \max, \\ [Ax]_i + x_{n+i} = b_i, i = \overline{1,m} \quad , \\ x_j \geq 0, j = \overline{1, n+m}; \end{cases} \qquad (P_2)$$

où M>>0 (un nombre positif très grand) et $x_{n+i}, i = \overline{n+m}$, des variables artificielles.

Le vecteur $X = (0,b) = \left(x = 0, x_{n+i} = b_i, i = \overline{1,m}\right)$ est une solution de base réalisable de (P_2) avec $A_B = (a_{n+1}, \ldots\ldots, a_{n+m}) = E_n$.

Ici on choisit M suffisamment grand de telle sorte à avoir

$c'x - M\sum_{i=1}^{m} x_{n+i} \leq 0, \forall\, x \geq 0, \forall\, x_{n+i} \geq 0$ et ceci dans le but de prouver que le problème (P_2) admet une solution optimale car c'est le maximum d'une fonction bornée sur un ensemble de contraintes non vide.

On résout le problème (p_2) par la méthode du simplexe avec une solution de base de départ $\{(0,b), A_B\}$ et on obtient une solution optimale $\left\{X^0 = \left(x^0, x^0_{n+i}, i = \overline{1,m}\right), A^0_B\right\}$.

- Interprétation de la solution de (P_2) :

a. S'il existe au moins un indice i_0 tel que $x^0_{n+i} \neq 0$, alors les contraintes de (P) sont contradictoires, c'est à dire, $\nexists\, x / Ax = b$.

b. Si $x^0_{n+i} = 0, \forall i = \overline{1,m}$, alors x^0 est solution optimale de (P).

Exemple 2 :

Considérons le problème de programmation linéaire suivant :

$$\begin{cases} Z = 2x_1 - x_2 + x_3 \rightarrow \max, \\ 3x_1 - x_2 + x_3 = 1 \\ x_1 + x_2 = 5 \\ 2x_1 - x_2 = 3 \\ x_j \geq 0,\ j = \overline{1,3}. \end{cases}$$

Le M-problème correspondant est :

$$\begin{cases} \overline{Z} = 2x_1 - x_2 + x_3 - Mx_4 - Mx_5 \rightarrow \max, \\ 3x_1 - x_2 + x_3 = 1 \\ x_1 + x_2 + x_4 = 5 \\ 2x_1 - x_2 + x_5 = 3 \\ x_j \geq 0,\ j = \overline{1,5}. \end{cases}$$

Avec x_4, x_5 des variables artificielles.

Pour ce problème, $\{x = (0,0,1,5,3), A_B = (a_3, a_4, a_5)\}$ est une solution réalisable de base.

Calculons le potentiel $y' = c'_B A_B^{-1}$ = (1, -M, -M) $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ = (1, -M, -M), et

les estimations $\Delta_j = y'a_j - c_j,\ j \in J_H$:

$\Delta_1 = y'a_1 - c_1 =$ (1, -M, -M) $\begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix}$ -2 = -3 M +1.

$\Delta_2 = y'a_2 - c_2 =$ (1, -M, -M) $\begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix}$ +1 = - M –1 ;$\Delta_j = 0, \forall\ j \in J_B$.

Dressons ***le premier tableau du simplexe :***

c			*2*	*-1*	*1*	*-M*	*-M*		
c_B	*Base*	b	a_1	a_2	a_3	a_4	a_5	θ_j	
1	a_3	1	3	-1	1	0	0	$1/3$	$\rightarrow L_1^1$
-M	a_4	*5*	*1*	*2*	*0*	*1*	*0*	$5/1$ =5	L_2^1
-M	a_5	*3*	*2*	*-1*	*0*	*0*	*1*	$3/2$	L_3^1
$\overline{Z}$ =-8M+1		Δj	*-3M+1*	*-M-1*	*0*	*0*	*0*		
			↑						

La solution de départ n'est pas optimale car $\min\limits_{j\in J_H} \Delta_j<0$, donc l'indice j_0=1 va rentrer dans la nouvelle base, comme $\min\limits_{j\in J_B} \theta_j=\theta_3$, de là l'indice j_1=3 sortira de la base de départ.

D'où $\overline{J}_B$={1,4,5}, $\overline{J}_H$={3,2}, déterminons alors la nouvelle solution en dressons ***le deuxième tableau du simplexe :***

C			2	-1	1	-M	-M		
c_B	Base	b	a_1	a_2	a_3	a_4	a_5	θ_j	
2	a_1	$1/3$	1	$-1/3$	$1/3$	0	0	/	$L_1^2 = \frac{1}{3} L_1^1$
-M	a_4	$14/3$	0	$7/3$	$-1/3$	1	0	2	$\rightarrow L_2^2 = L_2^1 - \frac{1}{3} L_1^1$
-M	a_5	$7/3$	0	$-1/3$	$-2/3$	0	1	/	$L_3^2 = L_3^1 - \frac{2}{3} L_1^1$
$\overline{Z}$ =-7M+$\frac{2}{3}$		$\overline{\Delta}_j$	0	-2M+$\frac{1}{3}$	M-$\frac{1}{3}$	0	0		
			↑						

La nouvelle solution est $\overline{x}=\left(\frac{1}{3},0,0,\frac{14}{3},\frac{7}{3}\right)$, elle n'est pas optimale car

$\min_{j \in \bar{J}_H} \bar{\Delta}_j = \bar{\Delta}_2 < 0$, on doit alors changer la base une autre fois avec $j_0 = 2$, comme indice rentrant, et $j_1 = 4$, l'indice sortant, on aura donc :

$\bar{\bar{J}}_B = \{1, 2, 5\}$, $\bar{\bar{J}}_H = \{3,4\}$, la nouvelle solution $\bar{\bar{x}}$ sera déterminer

en dressons ***le troisième tableau :***

C			2	-1	1	-M	-M		
c_B	Base	b	a_1	a_2	a_3	a_4	a_5	θ_j	
2	a_1	1	1	0	$2/7$	$1/7$	0	/	$L_1^3 = L_1^2 + \frac{1}{7} L_2^2$
-1	a_2	2	0	1	$-1/7$	$3/7$	0	2	$L_2^3 = \frac{3}{7} L_2^2$
-M	a_5	3	0	0	$-5/7$	$1/7$	1	/	$L_3^3 = L_3^2 - \frac{1}{7} L_2^2$
$\bar{\bar{Z}} = -3M$		$\bar{\bar{\Delta}}_j$	0	0	$\frac{5}{7}M - \frac{2}{7}$	$\frac{6}{7}M - \frac{1}{7}$	0		

Le critère d'optimalité est vérifié, car tous les $\Delta_j > 0$.

La solution optimale du M-problème est donc $x^0 = (1,2,0,0,3)$, le problème initiale n'admet pas de solution, car il existe une variable artificielle non nulle aller à une nouvelle ligne ($x_5^0 = 3$).

Conclusion :

On a présenté un algorithme utilisant les tableaux : Il faut savoir modéliser un problème, Il faut comprendre l'algorithme du simplexe et il faut être capable de le faire tourner sur des exemples simples.

Chapitre IV : résolution avec visuel Xpress

Section 1 : modélisation d'un problème de découpe

Dans le cas pratique qui a été lieu à l'ENIEM et dans les ateliers où se déroule la découpe on a constaté de nombreuse déchets pour faire un modèle de produit t pour cela nous avons pris un produit qui est désigne par le bahut.

Pour la fabrication d'un seul bahut :

On dispose des grandes plaques qui s'appelle plaques mères (la bobine) de tôle de 900 mètre×1mètre c'est là on découpe de plaques de taille différents désigné dans le tableau suivant :

Données en mm

Composant \ Dimensions	Longueur	Largeur	Epaisseur
Rai roulon	1324	940	1,25
Grand fond	1370	695	0,7
Tôle pré – paint	2782	861	0,7
Base compresseur	812	692	1,5
Petit fond	693	205	0,7
Angle d'acier en avant	1000	205	0,7

Comment on peut minimiser les déchets ?

Dans ce genre de problème, extrêmement combinatoire, on exploite le fait qu'il existe un nombre réduits de combinaisons de rectangle qui peuvent être extraits de la plaque mère. Ces motifs de découpe (patterns) peuvent être énumérés la figure qui suit nous désigne les sous – ensemble qu'on peut trouvés.

Il s'agit des sous ensembles maximaux, c'est-à-dire qu'au aucun autre rectangle commandé ne peut leur être ajouté.

Notons *m* le nombre de rectangle de tailles différentes présente dans la commande, *n* le nombre de motifs. Le nombre de rectangle de type i est noté b_i. Un motif j a un coût cj (cj = 1 si on veut simplement minimiser le nombre de plaques mères). Les contenus des motifs en nombres de rectangles des différents types peuvent être décrits par une matrice A, $m \times n$;

Il s'agit de calculer le nombre de plaques à découper dans chaque motif pour placer tous les rectangles, tout en minimisant le nombre total de plaques mères consommées.

TABLEAU

PATTERNS	1	2	3	4	5	6	7	8	9	10
1324*940	110	110	150	115	120	90	100	100	140	128
1370*695	143	142	142	112	110	100	120	100	135	130
2782*861	110	100	100	112	112	120	120	120	110	124
812*692	100	90	90	112	110	100	120	120	100	86
693*205	100	90	90	112	110	140	100	120	86	80
1000*205	100	90	90	112	111	130	101	114	80	80
CHUTES	0,17	0,81	0,56	0,316	0,366	0,34	0,02	0,63	0,16	0,14

$$110\,x_1+110x_2+150\,x_3+115\,x_4+120\,x_5+90\,x_6+100\,x_7+100\,x_8+140\,x_9+128x_{10} \geq 1000$$

$$143\,x_1+142x_2+142\,x_3+112\,x_4+110\,x_5+100\,x_6+120\,x_7+100\,x_8+135\,x_9+130x_{10} \geq 1000$$

$$110\,x_1+100x_2+100\,x_3+112\,x_4+112\,x_5+120\,x_6+120\,x_7+120\,x_8+110\,x_9+124x_{10} \geq 1000$$

$$100\,x_1+90x_2+90\,x_3+112\,x_4+110\,x_5+100\,x_6+120\,x_7+120\,x_8+100\,x_9+86x_{10} \geq 1000$$

$$100\,x_1+90x_2+90\,x_3+112\,x_4+110\,x_5+140\,x_6+100\,x_7+120\,x_8+86\,x_9+80x_{10} \geq 1000$$

$$100\,x_1+90x_2+90\,x_3+112\,x_4+111\,x_5+130\,x_6+101\,x_7+114\,x_8+80\,x_9+80x_{10} \geq 1000$$

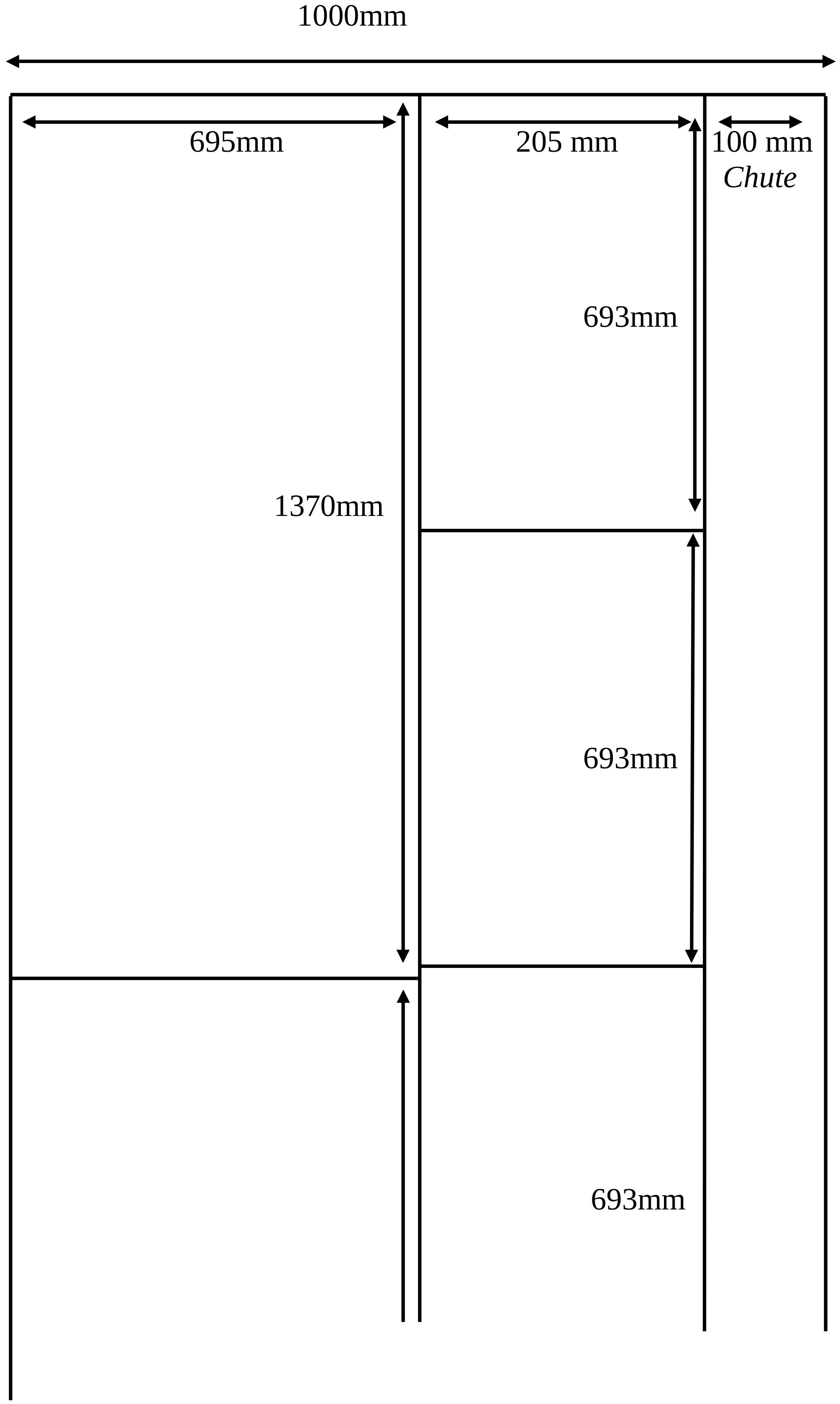
$$\text{MIN } Z=0{,}17\,x_1+0{,}81x_2+0{,}56\,x_3+0{,}316x_4+0{,}366\,x_5+0{,}34\,x_6+0{,}02\,x_7+0{,}63\,x_8+0{,}16\,x_9+0{,}14x_{10}$$

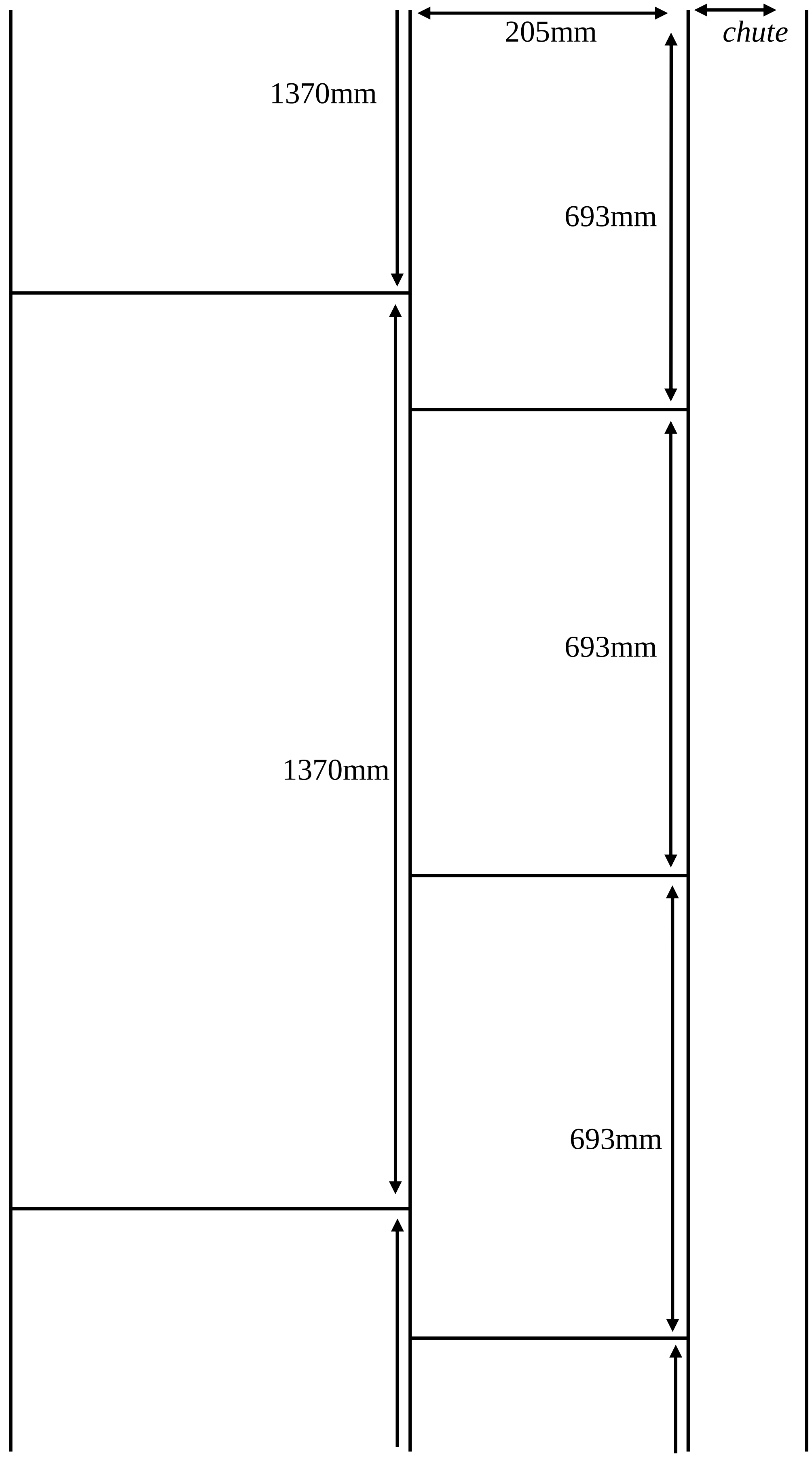
$$x_j \geq 0,\ j=1,\ldots,n$$

On va découper le problème en deux sous problème selon la largeur :

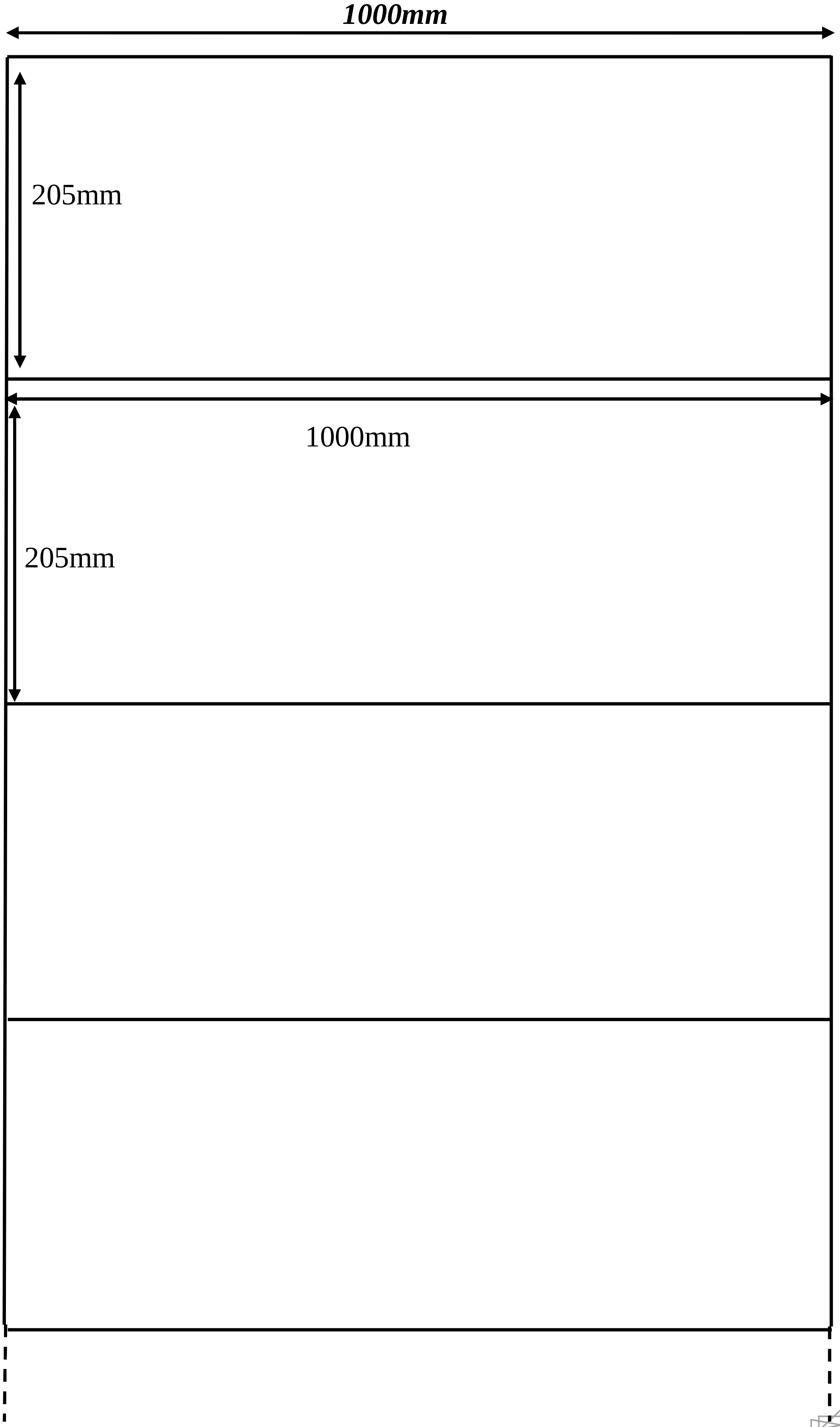
- 695mm + 205mm = 900 mm donc il ya une chute de 100mm sur la largeur.
- Pour l'angle d'acier la découpe doit être prendre d'une façon horizontale c'est-à-dire la largeur de la bobine et on ne va pas avoir de chute.
- Pour le reste des plaques on ne peut pas minimiser les chutes donc les autres problèmes ne peuvent pas être optimisés.

Pour le premier sous – problème le schéma sous dessous indique la découpe :




205mm
chute
1370mm
693mm
693mm
1370mm
693mm

Pour le deuxième sous problème le schéma ci-dessous l'indique :



Section 2 : Logiciel visuel Xpress

Grâce aux progrès de l'informatique, l'offre de logiciels commerciaux de résoudre des programmes linéaires de plus en plus gros considérablement augmenté.

Ce chapitre à pour but de présenter le logiciel Visuel Xpress et de résoudre le problème de découpe avec ce logiciel.

Visuel Xpress contient à la fois un modeleur et un solveur, mais leur accès est bien séparé dans l'interface, ce qui facilite la compréhension de la méthodologie de résolution d'un modèle.

2.1 Présentation de fonctionnement de logiciel

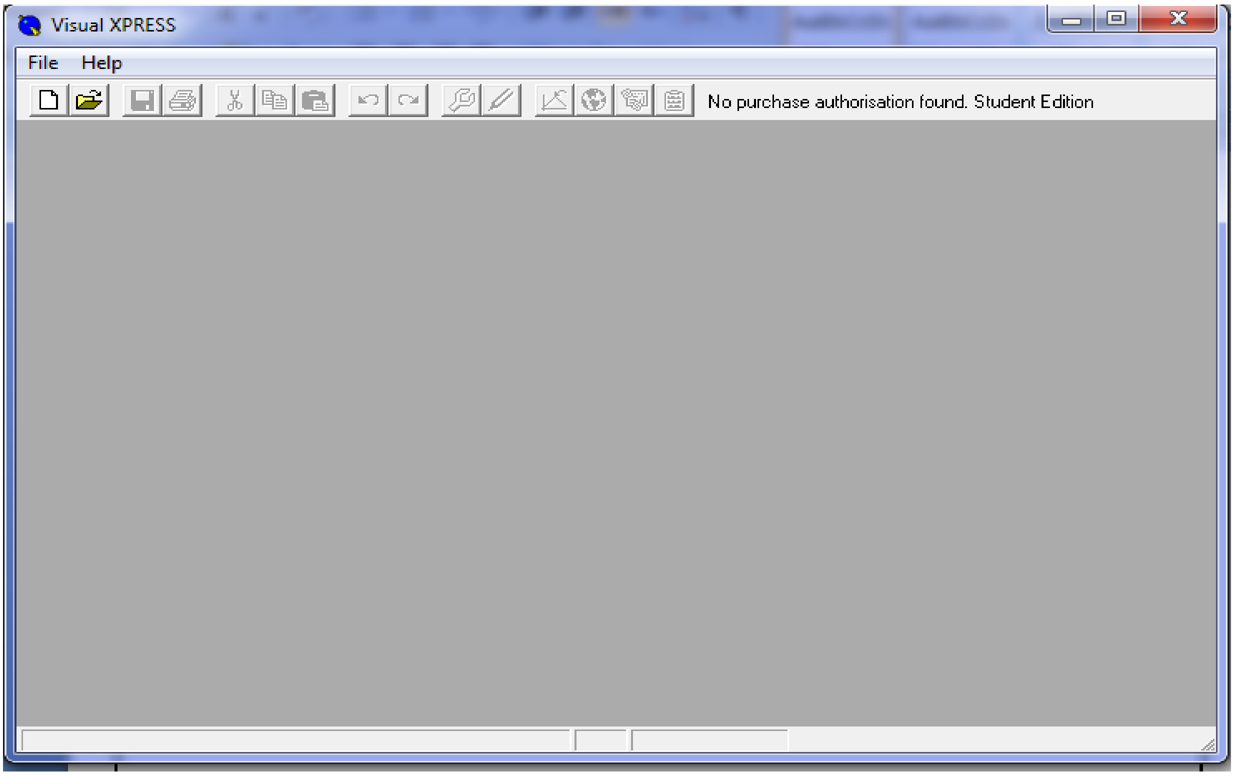


Figure IV.1 : Interface de Visuel Xpress

Nous cliquons sur l'icône **File** puis sur **new** figure IV.3 et l'écran de l'espace de travail s'ouvre :

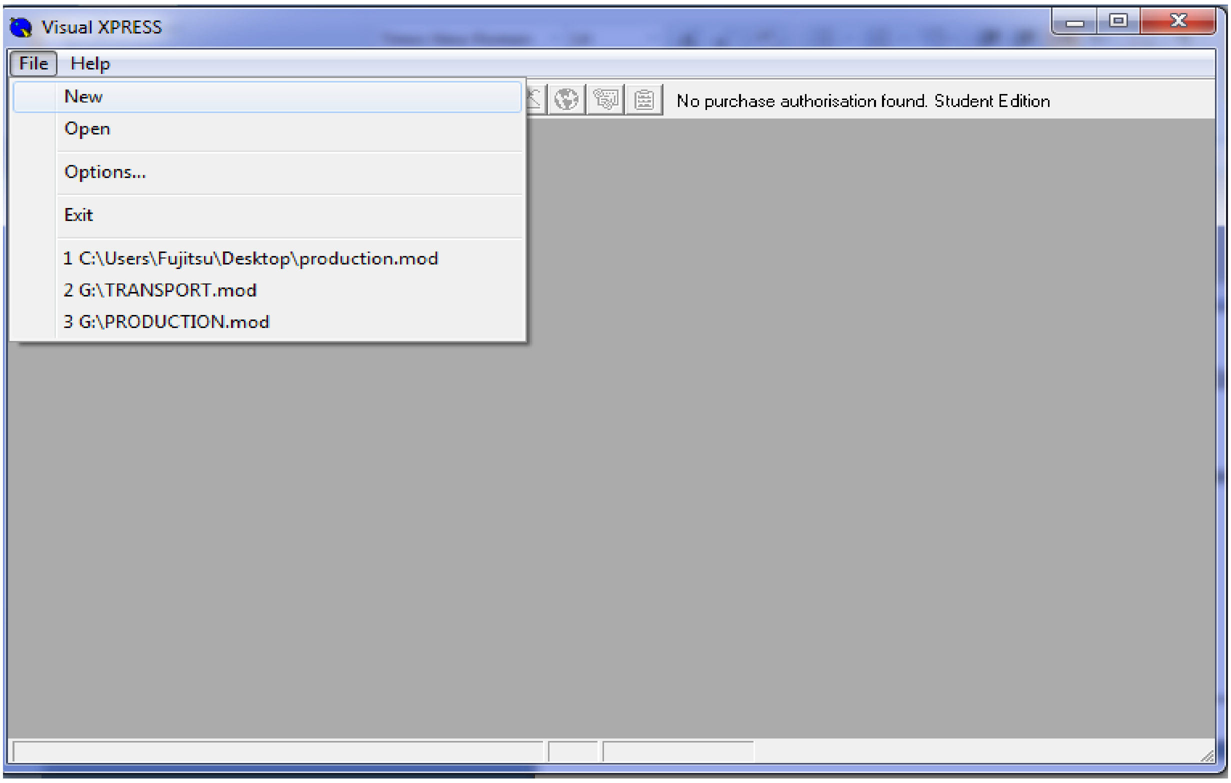


Figure IV.2

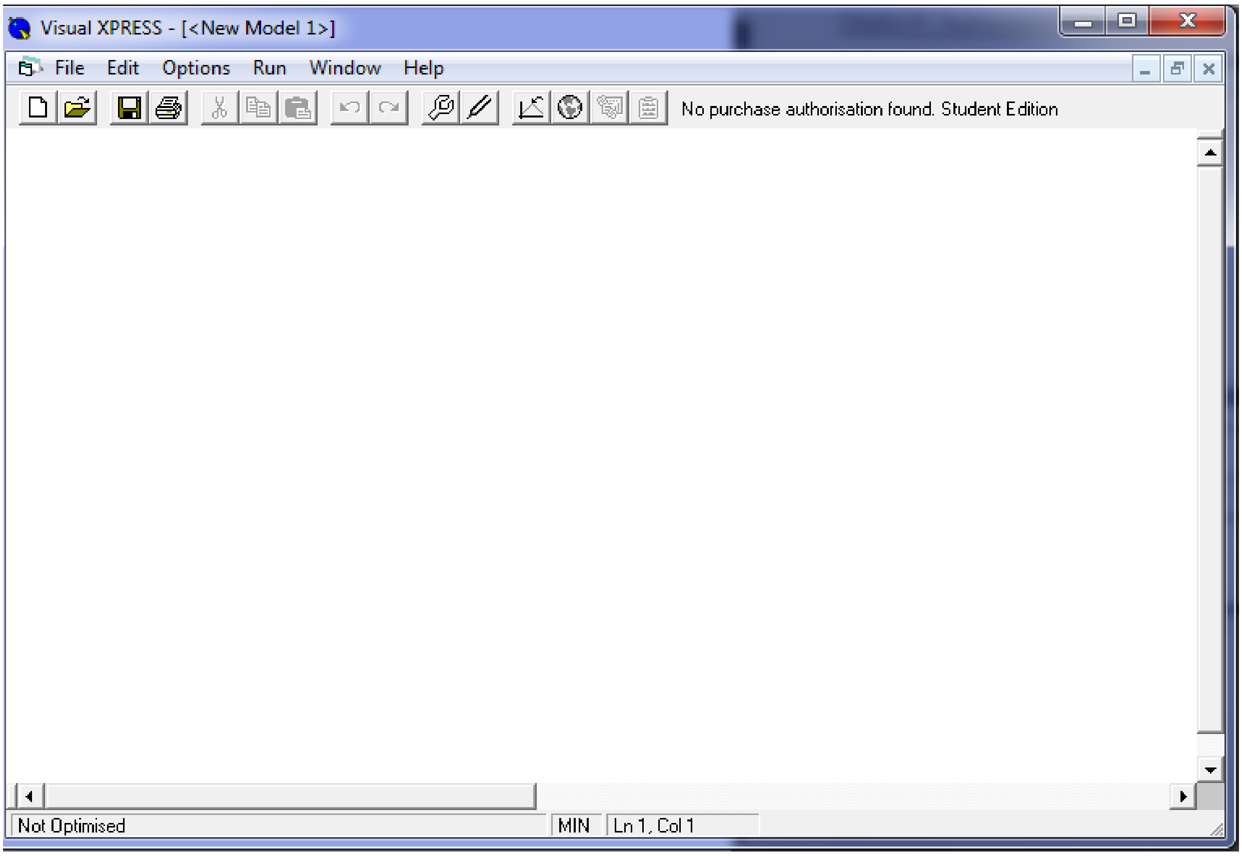


Figure IV.4 : L'espace de travail de Visuel Xpress

2.2 Les étapes de programme :

On saisie notre programme dans l'espace de travail et l'exécution sera comme suit :

Avant d'exécuter un problème chargé dans l'éditeur, il faut désigner le sens de l'optimisation (**MIN** ou **MAX**) dans **optimiser/options.**

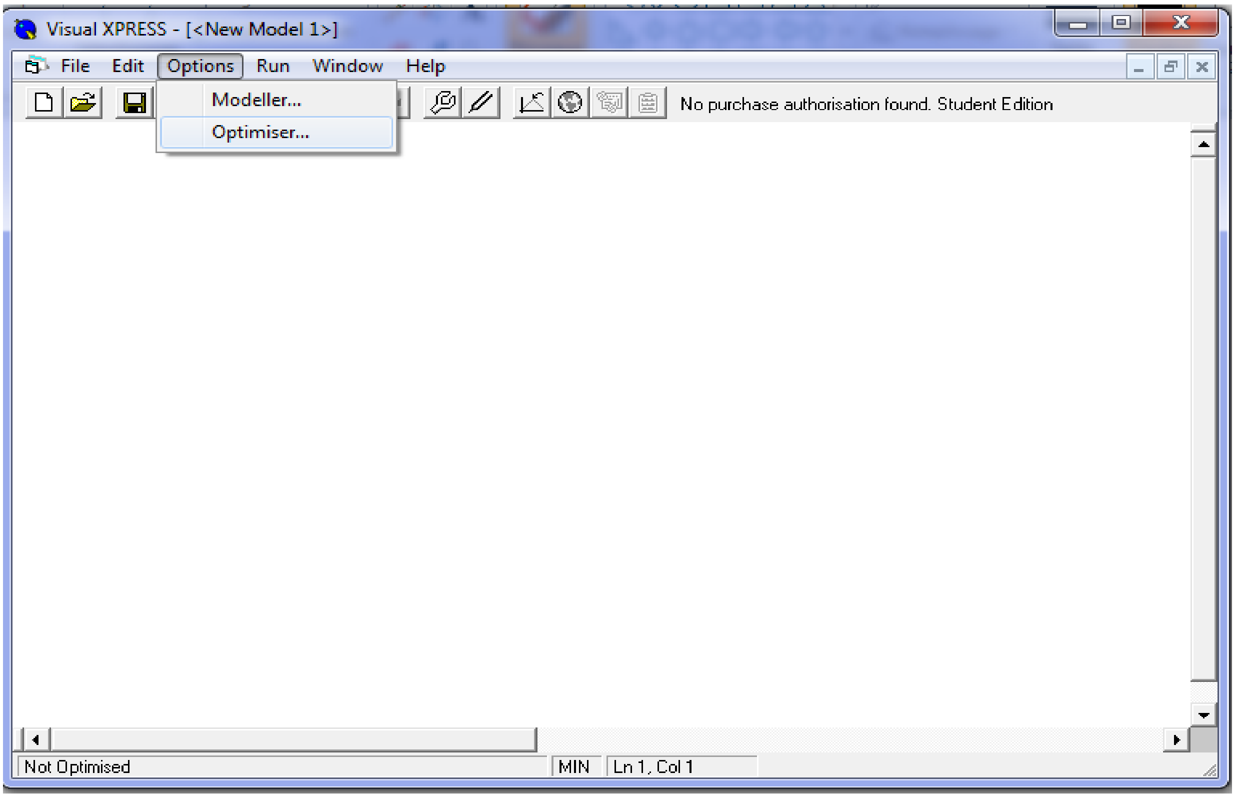


Figure IV.5

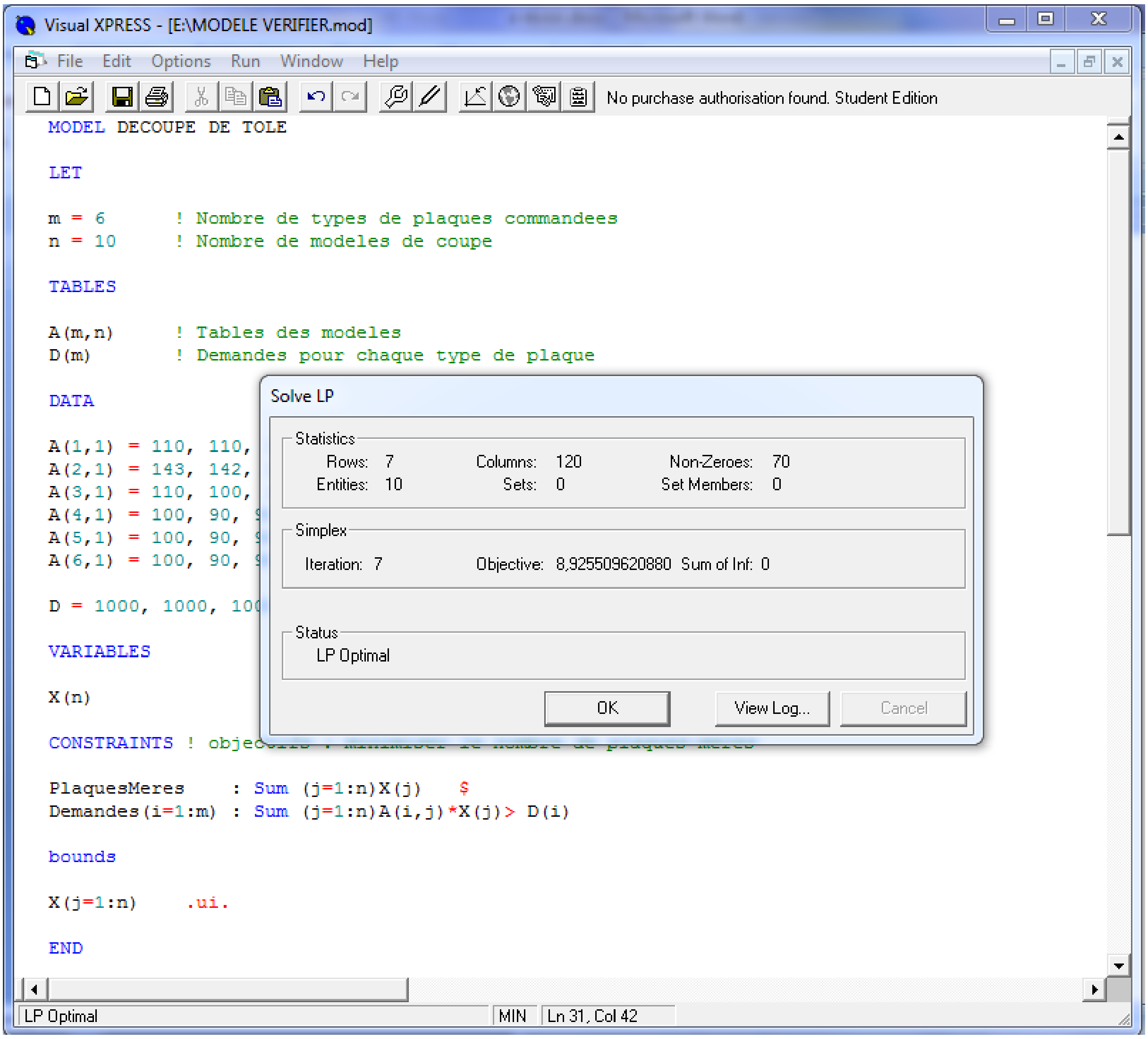
Section 3 : Résolution du problème de découpe avec Visuel Xpress

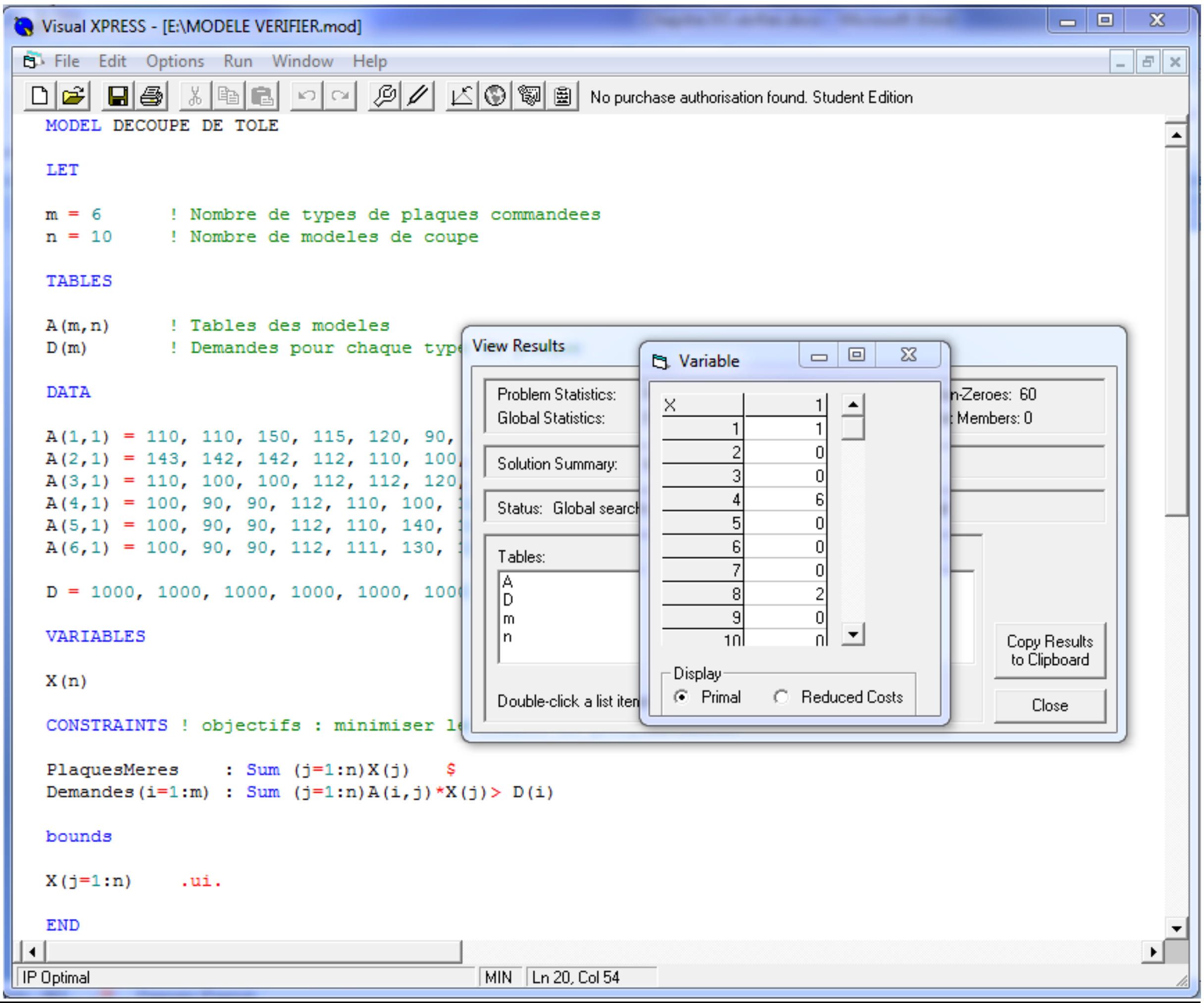
Le problème posé c'est qu'on dispose de grande plaques de tôles de 900m×1 m, appelées plaque mères. Pour la construction de 1000 bahut nous avons besoins de **1000** plaques de chaque type de plaques Rai roulon, Grand fond, Tôle pré – paint, Base compresseur, Petit fond, Angle d'acier en avant.

Donc nous prenant notre tableau :

PATTERNS	1	2	3	4	5	6	7	8	9	10
1324*940	110	110	150	115	120	90	100	100	140	128
1370*695	143	142	142	112	110	100	120	100	135	130
2782*861	110	100	100	112	112	120	120	120	110	124
812*692	100	90	90	112	110	100	120	120	100	86
693*205	100	90	90	112	110	140	100	120	86	80
1000*205	100	90	90	112	111	130	101	114	80	80

Modèle linéaire avec Visuel Xpress :





Résultats :

L'algorithme de simplexe trouve 8.99551 plaques mères en 7 itérations, plusieurs variables x_j sont fractionnaires. Le solveur en nombre entiers trouve 9 plaques, à découper en 01 motif N°1, 6 motif N° 4 et 2 motif N° 8.

Conclusion Générale

A travers cette étude, et parmi les remarques qu'on a constatées, il ne suffit pas d'appliquer les résultats obtenus à travers ce modèle seulement, mais on peut encore améliorer la solution bien plus que ça:

- **Pourquoi ce réapprovisionner seulement en bobine de l m ? l'ENIEM peut bien exiger de son fournisseur une partie des bobines en largeur 80cm et 120cm.**
- **Exploiter une partie des chutes pour fabriquer des glacières électriques.**
- **Le problème de l'épaisseur :**

Les bobines ont une épaisseur de 1,5 ; et parfois on a besoin de plaques de 0,7 d'épaisseur, alors le service production transforme à l'aide d'une machine spéciale les plaques de 1,5 d'épaisseur pour obtenir autant de 0,7 d'épaisseur, ceci engendre une perte de 0,8 d'épaisseur.

Nous recommandons ainsi à l'ENIEM d'exiger de son fournisseur qu'une partie des bobines soient en 0,7 d'épaisseur.

BIBLIOGRAPHIE

1. **Christelle GUERET**, Christian PRINS, Marc SEVAUX, Programmation linéaire : 65 d'optimisation modélisés et résolus avec visuel Xpress, édition Eyrolles, 2000 ;
2. **Jacques TEGHEM**, programmation linéaire, édition de l'université de Bruxelles ; Ellipses édition, 1996 ;
3. **Gérald BAILLARGEON** ; programme linéaire appliqué : outils d'optimisation et d'aide à la décision, édition SMG 1996 ;
4. **Brahim OUKACHA**, **Mohamed AIDEN**, programmation linéaire, page bleue, 2005
5. **KESSI Karima ; SIDENNAS Djedjiga,** méthodes de résolution de problème de programmation linéaire en nombre entiers, étude pratique avec visuel Xpress, promotion 2010.