UNIVERSITÉ ABDELHAMID IBN BADIS-MOSTAGANEM
FACULTÉ DES SCIENCES EXACTES ET INFORMATIQUE
DÉPARTEMENT DE MATHÉMATIQUES

Mémoire de fin d'étude
Pour l'obtention du diplôme de Master en Mathématiques
Cycle LMD
Spécialité :Modélisation Contrôle et Optimisation

Thème :
Distribution de fioul

Présenté par :
Maradj Samira

Soutenu le .09-06-2014

Les membres de jury

DAHMANI	Zoubir	**Président**	MCA	**U. MOSTAGANEM.**
ABLAOUI	Hocine	**Examinateur**	MAA	**U. MOSTAGANEM.**
AMIR	Abdessamad	**Encadreur**	MCA	**U. MOSTAGANEM.**

Dédicace

Je dédie ce modeste travail à mon époux
ainsi qu'à mes chers enfants Abdennour, Miloud et Fadoua.

Remerciements

Tout d'abord je tiens à remercier Dieu le tout puissant et miséricordieux, qui m'a donné la force et la patience d'accomplir ce modeste travail.
En second lieu, je tiens à exprimer mes sincères remerciments à mon encadreur Mr AMIR Abdessamad, pour ses précieux conseils et son aide durant toute la période du travail.
Mes vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à cette recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.
Je souhaite aussi dire merci à ma très chère mère et mon frère pour m'avoir aidé et encourager durant ses deux années d'étude.
Enfin, je tiens également à remercier toutes les personnes qui ont participés de près ou de loin à la réalisation de ce travail.

Résumé

Dans ce mémoire, nous formulons le problème de distribution de fioul comme programme linéaire en variables mixtes (binaires et continues), qui constitue une classe de problème de programmation mathématique difficile à résoudre. Nous proposons Une méthode de type " Branch and Bound" (Méthode d'évaluation et séparation) pour le résoudre.

Table des matières

INTRODUCTION

Le fioul est un élément indispensable dans le développement économique. On souhaite attirer l'attention sur les difficultés qu'éprouvent les entreprises de distribution pour répondre à la demande des consommateurs, et qui engendre une perte de temps et accroissement de charges financiéres non négligeables du à une mauvaise gestion.

Afin de parvenir à des solutions concrètes, à savoir ; comment minimiser le nombre de kilomètres parcourus par des camions citernes afin de satisfaire la demande des clients dans les differentes villes à partir d'un dépôt, on introduit ce problème dans un cadre mathématique on le formulant comme un problème de programmation linéaire en variables binaires et continues (mixtes) qui constitue une classe particulière de la programmation mathématique, difficile à résoudre .

Les problèmes linéaires à variables mixtes comportent des variables de décision discrètes et continues avec des contraintes et un objectif exprimés de manière linèaire en fonction de ses variables de décision .

Dans ce mémoire, le problème de distribution de fioul est modelisé comme programme linéaire en variable mixte avec des variables discrètes binaires. La résolution de ce problème est abordée par une méthode exacte appelée Branch and Bound (Méthode d'évaluation et séparation)(Méthode d'évaluation et séparation) qui permet de séparer le problème initial en sous problèmes afin de faciliter sa résolution. Le manuscrit contient trois chapitres :

- Dans le Premier chapitre nous introduisons les problèmes linéaires en variables discrètes, et quelques propriétés comme la relaxation linéaire et l'unimodularité.
- Dans le deuxième chapitre, nous formulons le problème de distribution de fioul comme un problème de programmation linéaire en variable mixte (0-1 et continues).
- Dans le troisième chapitre nous présentons une méthode Branch and Bound basée sur la relaxation linéaire pour résoudre un exemple réel de distribution de fioul posé dans la région de Donges dans le nord ouest de la France.
- Nous concluons le mémoire en dernier lieu.

Chapitre 1

Programmation linéaire en variables entières :

1.1 Formulation Mathématique

Considérons la formulation mathématique d'un problème de programmation linéaire en variables continues noté (LP)[11] :

$$(LP)\begin{cases} z = \min\ c^T x \\ x \in\ D \end{cases} \tag{1.1.1}$$

où $c \in \mathbb{R}^{n\times 1}$ et D est un polyèdre convexe des solutions admissibles dans $\mathbb{R}^n$,

$$D = \{x \setminus Ax \leq\ b,\ \ x \geq 0\}, \tag{1.1.2}$$

avec $A \in \mathbb{R}^{m\times n}$ et $b \in \mathbb{R}^{m\times 1}$.

Lorsque toutes les variables sont entières, le problème résultant est un problème de programmation linéaire en variables entières, noté (ILP) ("integer linear programming") ;

$$(ILP)\begin{cases} z = \min c^T x \\ x \in S = D \cap \mathbb{Z}_+^n \end{cases}. \tag{1.1.3}$$

Si la variable x est restreinte à prendre que les valeurs 0 ou 1, on dira qu'on a un programme linéaire en 0-1, ou un programme linéaire en variables binaires noté (BIP) ("binary integer programming") ;

$$(BIP)\begin{cases} z = \min c^T x \\ Ax \leq b \\ x \in \{0,1\}^n \end{cases}. \tag{1.1.4}$$

Si une partie seulement des variables doit être entière, le problème résultant est un problème de programmation mixte en variables entières, noté $(MILP)$("mixed integer linear programming"). En séparant les deux types de variables dans la fonction objectif et les contraintes, un $MILP$ se formule :

$$(MILP)\begin{cases} z=\min(c_1^T x + c_2^T y) \\ A_1 x + A_2 y \leq b \\ x \in \mathbb{Z}_+^q \qquad y \in \mathbb{R}^{n-q} \qquad \{1...q\} \subset \{1...n\} \end{cases} \tag{1.1.5}$$

où $A_1 \in M_{(m,q)}$ et $A_2 \in M_{(m,n-q)}$.

1.2 Relaxation linéaire continue

Le problème (LP) donné par la relation (1.1.1) est appelé la relaxation linéaire de son (ILP), il s'agit de relâcher la variable, $x \in D \cap \mathbb{Z}_+^n$ en $x \in D$ donné par la relation (1.1.2). L'exemple suivant, nous dit que la solution du problème relaxé n'a parfois rien à voir avec la solution du (ILP) associé.

Exemple 1.2.1 *Considérons le problème (ILP) définit par*

$$\begin{cases} z=\max 10x_1+11x_2 \\ 10x_1+12x_2 \leq 59 \\ x_1\ ,\ x_2 \geq 0 \\ x_1, x_2\ entiers \end{cases}$$

En variables continues, la solution optimale est $x^* = (x_1^*, x_2^*)^T = (5.9, 0)^T$ qui n'est pas entière avec $z^* = 59$
Un arrondi de cette solution serait $x^* = (6, 0)^T$ .la solution du (ILP) est $x^* = (1\ ,\ 4)^T$ avec $z^* = 54$.

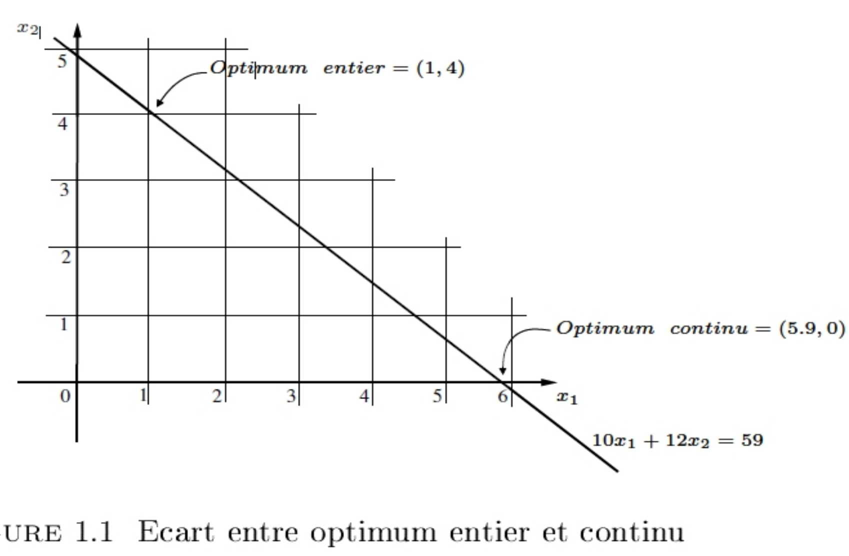


FIGURE 1.1 Ecart entre optimum entier et continu

On note par $\mathcal{V}_{\ (p)}$la valeur d'un problème d'optimisation (p). Alors, on a toujours :

$$\mathcal{V}_{(LP)} \leq \mathcal{V}_{(ILP)} \text{ (pour un problème de minimisation)}$$

et

$$\mathcal{V}_{(LP)} \geq \mathcal{V}_{(ILP)} \text{ (pour un problème de maximisation).}$$

Un problème linéaire à coefficients entières peut avoir une solution entière sous certaines conditions.

Définition 1.2.1 *Une matrice A $(m \times n)$ à coefficient entières est dite totalement unimodulaire (TUM) si les déterminants de toutes ses sous -matrices carrées valent 0, 1 ou -1 (voir [11]).*

Considérons le problème (LP) donné par (1.1.1) et D un polyèdre non vide défini par la relation (1.1.2). On a le théorème suivant :

Théorème 1.2.1 ([10]) *Si A est une matrice TUM alors tous les sommets du polyèdre D sont entiers pour tout vecteur entier b. En particulier, si le problème (LP) possède une solution optimale, alors il admet une solution optimale entière.*

Le résultat suivant donne une condition suffisante pour qu'une matrice soit TUM.

Théorème 1.2.2 ([10]) *Soit A une matrice contenant seulement les éléments 0,1ou -1 et satisfaisant les deux conditions suivantes :*

1. *Chaque colonne contient au plus deux éléments non-nuls.*
2. *Les lignes de A peuvent être partitionnées en deux sous ensembles I_1 et I_2 tels que pour chaque colonne contenant deux éléments non nuls :*
   -Si les deux éléments non-nuls ont le même signes alors l'un est dans I_1 et l'autre dans I_2.
   -Si les deux élément non-nuls sont de signes différents alors ils sont tous les deux dans I_1 ou tous les deux dans I_2 . Alors, A est TUM.

Le problème d'affectation est un exemple de problème $(BILP)$, dont la matrice des contraintes est TUM, il sera abordé à la section suivante.

1.3 Problème d'affectation

Le probème d'affectation ("Assignement problem"(AP)) est un problème de programmation linéaire en variables binaires, il consiste à affecter n ressources (personnes, équipements, ...) à n activités (travaux, services,...), en minimisant le coup global des affectations. Il s'agit de définir une bijection de $\{1...n\} \to \{1...n\}$, ou encore une permutation de n objets . Désignons par $i = 1, ..., n$ les ressources et par $j = 1, ..., n$ les activités, et introduisons les n^2 variables x_{ij} :

$$x_{ij} = \begin{cases} 1 & \text{si la ressource } i \text{ est affectée à l 'activité } j \\ 0 & \text{autrement} \end{cases}$$

La contrainte ressource i, doit être affectée à une seule activité j, est modélisée par les n équations

$$\sum_{j=1}^{n} x_{ij} = 1 \text{ pour } i = 1, ..., n.$$

La contrainte activité j, ne peut être affectée qu'à une et une seule ressource i, est modélisée par les n équations

$$\sum_{i=1}^{n} x_{ij} = 1 \text{ pour } \ j = 1, ..., n.$$

Si on note par c_{ij} le coût d'affecter une ressource i à une activité j, Le problème d'affectation est :

$$(AP) \begin{cases} \min \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} c_{ij} \, x_{ij} \\ \sum\limits_{j=1}^{n} x_{ij} = 1 \qquad i = 1...n \\ \sum\limits_{i=1}^{n} x_{ij} = 1 \qquad j = 1, ...n \\ x_{ij} = \ 0, 1 \end{cases}$$

La matrice des contraintes $A \in \mathbb{R}^{2n \times n^2}$ est donnée par

$$A = \begin{pmatrix} 1 & \cdots & \cdots & 1 & & & & & & & & \\ & & & & 1 & \cdots & \cdots & 1 & & & & \\ & & & & & & & & \ddots & & & \\ & & & & & & & & & 1 & \cdots & \cdots & 1 \\ 1 & & & 1 & & & & & & 1 & & & \\ & \ddots & & & & \ddots & & & & & & \ddots & \\ & & 1 & & & & 1 & & & & & & 1 \end{pmatrix}$$

D'aprés le théorème (1.2.2) la matrice A est TUM, le second membre b est entier (b_i et b_j égale à 1), d'aprés le théorème(1.2.1) les coordonnées des sommets de D, valent 0 ou 1.

Remarque

Il est possible de résoudre ce problème avec les techniques et méthodes de la programmation linéaire, après avoir relâché les variables binaires. Mais étant donné que le rang de la matrice A est égale à $2n-1$ (parmi les $2n$ contraintes il y en a $2n-1$ linéairement indépendantes) et que dans chaque solution n variables seront positives (égale à1). Les solutions de bases sont toutes fortement dégénérées ($(n-1)$ variables de base sont nulles). L'utilisation de la méthode du simplexe pour résoudre le problème d'affectation engendrera forcément des problèmes de cyclage voir[**?**].En pratique, pour résoudre le problème d'affectation linéaire, il est préferable d'utiliser par exemple la méthode hongroise .

Chapitre 2

Problème de distribution de fioul

2.1 Présentation du problème

Le problème de distribution de fioul consiste à déterminer les tournées à réaliser afin de livrer tous les clients de façon à minimiser le nombre de kilomètres parcourus et en satisfaisant la demande de chaque client ainsi que la capacité maximale du véhicule. Nous le présentons ici sous forme d'un exemple réel. Il s'agit d'un transporteur qui doit livrer du fioul à un certain nombre de clients de la région ouest à partir de la raffinerie de Donges(France). Ses clients se situent à six villes différentes . Le tableau 1 contient les demandes en litres pour les différents sites de ce probléme. La matrice des distances en kilomètres séparant les clients et la raffinerie est donnée au tableau 2. Pour faire ses livraisons, le transporteur dispose de camions citernes pouvant contenir jusqu'a 39000 milles litres voir [7].

Brain-sur-Authion	carquefou	Donges	Guérande	Haie fouassiére	Mésanger	Ponts-de-Cé
14000	3000	0	6000	16000	5000	15000

Tableau1 : Demandes des sites (en litres)

	Brain-sur-A	carquefou	Donges	Guérande	Haie F	Mésanger	Ponts-de-Cé
Brain-sur-Authion	-	93	148	180	99	72	12
carquefou	93	-	55	85	20	28	83
Donges	148	55	-	32	70	73	140
Guérande	180	85	32	-	100	99	174
Haie fouassiére	99	20	70	100	-	49	85
Mésanger	72	28	73	99	49	-	73
Ponts-de-Cé	12	83	140	174	85	73	-

Tableau2 distancier (km)

2.2 Modèlisation du problème

Dans ce paragraphe nous allons formuler le modèle mathématique pour n villes. Soient des variables binaires x_{ij} valant 1 si la ville i est avant la ville j dans une tournée,0 sinon. Soit

D_{ij} la distance séparant deux villes i et j et Q_i la quantité demandée par le client i. L'objectif est de minimiser le nombre de kilomètres parcourus, la fonction objectif à minimiser est donc

$$\sum_{j=1}^{n}\sum_{i=1}^{n} D_{ij}\, x_{ij} \tag{2.2.1}$$

Chaque ville doit être livrée en une seule fois , ceci se traduit par les deux contraintes 2.2.2 et 2.2.3 qui imposent respectivement qu'on doit entrer une et une seule dans chaque ville (sauf le dépôt) et doit quitter chaque ville (sauf le dépôt) une et une seule fois.

$$\forall j = 2, ..., n : \sum_{i=1, i\neq j}^{n} x_{ij} = 1 \tag{2.2.2}$$

$$\forall\ i = 2, ..., n \quad \sum_{j=1, i\neq j}^{n} x_{ij} = 1 \tag{2.2.3}$$

Afin de respecter la capacité des camions citernes, nous allons utiliser des variables c_i correspondant à la quantité de fioul livrée aux différent clients le long du trajet allant du dépôt au client i inclus. Cette quantité c_i doit être supérieure à la quantité Q_i demandée par le le client i et inférieure à la capacité $Q_{\max}$ des camions-citernes. De plus, si le client i est le premier de la tournée, alors c_i doit être égal à la quantité demandée par le client i. Cette contrainte se traduit par les deux contraintes 2.2.4 et 2.2.5.

$$\forall i = 2, ..., n \quad Q_i \leq c_i \leq Q_{\max} \tag{2.2.4}$$

$$\forall i = 2, ..., n \quad c_i \leq Q_{\max} - (Q_{\max} - Q_i)x_{1i} \tag{2.2.5}$$

En effet si i est le premier client d'une tournée , alors x_{1i} vaut 1 et,après simplification, la contrainte 2.2.5 est équivalente à la contrainte 2.2.6.

$$c_i \leq Q_i \tag{2.2.6}$$

Alors 2.2.5 et 2.2.4 impliquent que c_i est égale à la demande du client i ($c_i = Q_i$). Dans le cas ou i n'est pas le premier de tournée,x_{1i} vaut 0 ,et la contrainte 2.2.5 est équivalente à la contrainte 2.2.7. Cette relation est redondante puisque déjà exprimée dans la contrainte 2.2.4

$$c_i \leq Q_{\max} \tag{2.2.7}$$

$$\begin{aligned} \forall j &= 2...n, \forall i = 2...n, i \neq j \\ c_j &\geq c_i + Q_j - Q_{\max} + Q_{\max}.x_{ij} + (Q_{\max} - Q_j - Q_i)x_{ji} \end{aligned} \tag{2.2.8}$$

Considérons maintenant le cas où i n'est pas le premier client de la tournée .Alors c_i doit être égale à la somme des quantités livrées entre le le dépôt et i inclus. Ainsi,si le client j est aprés le client i sur une tournée, on peut écrire que c_j doit être égal à la quantité livrée sur le trajet du dépôt à i inclus , quantité à laquelle s'ajoute la quantité demandée par j. Cette relation est traduite par la contrainte 2.2.8 . En effet ,si j est juste après i sur une tournée , x_{ij}vaut 1 et x_{ji} vaut 0.La contrainte 2.2.8 est équivalente à la contrainte 2.2.9.

$$c_j \geq c_i + Q_j \tag{2.2.9}$$

Dans le cas où j n'est pas juste après i , la contrainte 2.2.8 reste vérifiée. En effet dans le cas où j est juste avant i, la contrainte2.2.8 devient 2.2.10 .

$$c_j \geq c_i - Q_i \tag{2.2.10}$$

Cette contrainte signifie que la quantité livrée sur le trajet du dépôt à j est supérieure à la quantité livrée entre le dépôt et le successeur i de j sur la tournée, quantité à laquelle on doit retirer la quantité livrée en i. Enfin, si i et j ne sont pas côte à côte sur une tournée, on obtient la contrainte 2.2.11. Comme elle a un second membre inférieur ou égal à Q_i ,cette contrainte est redondante puisque déjà exprimée par la contrainte 2.2.4.

$$c_j \geq c_i + Q_i - Q_{\max} \tag{2.2.11}$$

$$\forall i = 2...n : c_i \geq 0 \tag{2.2.12}$$

$$\forall i = 1...n, \forall j = 1...n, i \neq j : x_{ij} \in \{0, 1\} \tag{2.2.13}$$

Enfin les contraintes 2.2.12 et 2.2.13 indiquent que les variables c_i sont positives et que les x_{ij} sont des variables binaires. L'affectation de variables c_i à chaque sommet i garantit le respect de la capacité des camions, tout en introduisant des tournées ne passant pas par le dépôt.
Finalement nous obtenons le problème $(MILP)$ suivant :

$$\begin{cases} \min \sum\limits_{j=1}^{n} \sum\limits_{i=1}^{n} D_{ij} x_{ij} \\ \sum\limits_{i=1,\ i \neq j}^{n} x_{ij} = 1 \quad ; \quad j = 2...n \\ \sum\limits_{j=1,\ j \neq i} x_{ij} = 1 \ ; \quad i = 2...n \\ Q_i \leq c_i \leq Q_{\max} \quad ; i = 2...n \\ c_i \leq Q_{\max} - (Q_{\max} - Q_i) x_{1i} \quad ; i = 2...n \\ c_i + Q_j - Q_{\max} + Q_{\max}.x_{ij} + (Q_{\max} - Q_j - Q_i) x_{ji} \leq c_j \quad ; j = 2...n \ , i = 2...n, i \neq j \\ c_i \geq 0 \quad ; i = 2...n \\ x_{ij} \in \{0, 1\} \qquad i = 1...n; j = 1...n; i \neq j \end{cases}$$

L'objectif du chapitre suivant est de développer au moins une méthode pour résoudre ce problème.

Chapitre 3

Algorithme Branch and Bound

Nous nous intéressons dans ce chapitre à la méthode de séparation et d'évaluation en anglais dites Branch and Bound ($B\&B$) qui est l'une des méthodes les plus efficaces pour la résolution exacte des problèmes de programmation linéaire en variables mixtes. Elle est basée sur trois éléments principaux :(voir [11])

- Une procédure de séparation("branching process")

Qui à chaque fois qu'il sera nécessaire de séparer un sous ensemble de solutions, déterminera combien de nouveau sous ensembles considérer et comment les constituer. Le nombre de sous ensembles créés n'est pas nécessairement identique à chaque séparation, il peut dépendre du résultat de l'analyse du sous-problème parent. En appliquant systématiquement le principe de séparation, le nombre total de noeuds doit être fini.

- Une procédure d'évaluation("Bounding process")

Qui consiste à analyser un sous-problème, cette analyse vise à évaluer la valeur optimale de la fonction économique du sous-problème, c-à-d déterminer une borne inférieure (supérieure pour un problème de maximisation). Pour obtenir une borne inférieure, la technique la plus utilisée est de procéder à une relaxation du problème qui consiste à élargir l'ensemble des solutions admissibles.

- Une procédure de cheminement

Qui indique quels sous-ensembles analyser et dans quel ordre. Il est souhaitable d'examiner le moins de sous-ensembles possibles : certains d'entre eux pourront ne pas être séparés (ne pas contenir de solutions meilleures que celle déjà trouvées), Nous dirons qu'un tel sous-ensemble ou le noeud correspondant de l'arborescence est sondé. Lorsqu'un noeud de l'arborescence est sondé, il conviendra de remonter dans l'arborescence vers un autre noeud situé à un niveau inférieur ou égal dans l'arborescence ("*Bracktraking process*").

3.1 Algorithme Branch and Bound pour un MILP

Considérons le problème de programmation linéaire en variables mixtes suivant [2] :

$$\begin{cases} z = \min c_1^T x + c_2^T y \\ A_1 \begin{pmatrix} x \\ y \end{pmatrix} = b_1 \ \wedge \ A_2 \begin{pmatrix} x \\ y \end{pmatrix} \leq b_2 \\ y_{\min} \leq y \leq y_{\max} \quad x \in \{0,1\} \end{cases}$$

3.1.1 Relaxation linéaire

La relaxation linéaire (LP) du problème $(MILP)$ consiste à relâcher la variable discrète en variable continue ;

$$x_i \in \{0,1\} \rightarrow \ 0 \leq x_i \leq 1$$

Cette relaxation linéaire (LP) apporte au problème discret toute la richesse théorique de la programmation linéaire ; on l'utilise pour évaluer les sous-problèmes qui sont généralement plus faciles à résoudre que le problème initial. La valeur optimale de la relaxation (LP) du problème de minimisation $(MILP)$ fournit une borne inférieure sur la valeur optimale de ce dernier ; si la solution optimale de la relaxation (LP) est admissible pour le problème $(MILP)$, alors elle est aussi pour ce dernier. L'algorithme de Branch and Bound combine la stratégie de séparation des sous-ensembles par énumération et les techniques de relaxation. Des classes de sous-ensembles de solutions sont formées et examinées pour déterminer si elles contiennent ou non des solutions optimales. Les sous-ensembles analysés en détails sont ceux qu'on estime contenir la meilleure solution.

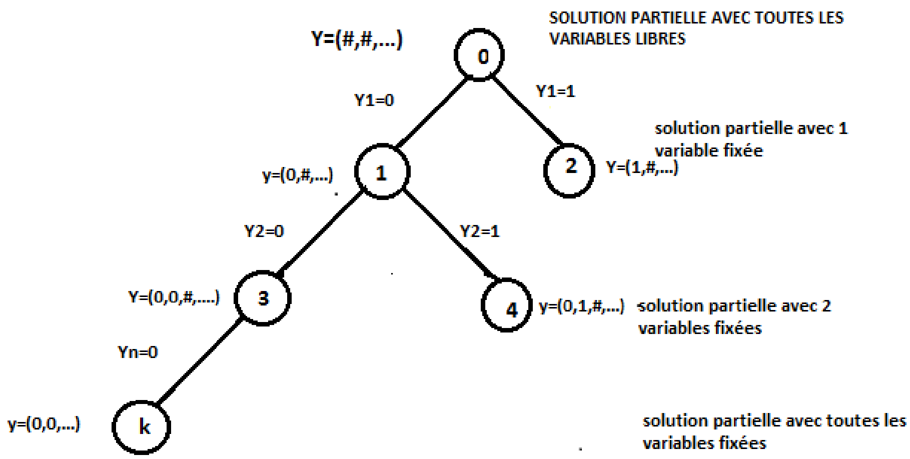
3.1.2 Les solutions partielles et complètes

-les solutions partielles ont certaines variables discrètes fixées tandis que les autres libres (fractionnaires) noté $\#$;
-Les solution complètes sont toutes les solutions possibles créés à partir de solution partielle en rendant binaires tous les variables fractionnaires.

Exemple 3.1.1 *En prenant* $x = (1, \#, 0, \#)$ *une solution partielle avec* $x_1 = 1$ *et* $x_3 = 0$, x_2, x_4 *sont libres ; alors les solutions complètes sont* $(1,0,0,0), (1,1,0,0), (1,0,0,1), (1,1,0,1)$.

3.1.3 L'arbre Branch and Bound

La figure ci-dessous donne par suite la méthode $(B\&B)$ comme arborescence.



-Les noeuds de l'arbre sont les solutions partielles.
- Les nombres, dans l'arbre indiquent l'ordre des noeuds analysés.
- Le nombre de noeuds est donné par : $1 + 2 + 2^2 + 2 + \cdots + 2^{n_x} = \sum_{i=0}^{n_x} 2^i > 2^{n_x}$
- Les branches de l'arbre de B&B indiquent les variables fixées.
On commence par le premier noeud (0) la procédure $(B\&B)$ débute par une solution partielle initiale $x^{(0)} = (\#, \#, \cdots)$ avec toutes les variables discrètes libres.

$$\left\{ \begin{array}{l} \quad z = \min c_1^T x + c_2^T y \\ A_1 \begin{pmatrix} x \\ y \end{pmatrix} = b_1 \ \wedge \ A_2 \begin{pmatrix} x \\ y \end{pmatrix} \leq b_2 \\ y_{\min} \leq y \leq y_{\max} \quad x \in \{0,1\} \end{array} \right. \qquad \begin{array}{c} LP \\ \rightarrow \\ relaxation \end{array} \qquad \left\{ \begin{array}{l} \quad z = \min c_1^T x + c_2^T y \\ A_1 \begin{pmatrix} x \\ y \end{pmatrix} = b_1 \ \wedge \ A_2 \begin{pmatrix} x \\ y \end{pmatrix} \leq b_2 \\ y_{\min} \leq y \leq y_{\max} \quad 0 \leq x \leq 1 \end{array} \right.$$

-La solution de la relaxation (LP) pour le premier noeud nous fournit une borne inférieure sur l'optimum global du problème de minimisation $(MILP)$. Il existe trois possibilités :
1-Pas de solution admissibles donc fin de la recherche et le problème $(MILP)$ n'a pas de solution optimale
2-Toutes les variables relaxées x_i sont 0 ou 1 à l'optimum ;
Procédure terminé, la solution trouvée est optimale pour le $(MILP)$.
3-Certaines variables relaxées x_i ont des valeurs fractionnaires.

Procédure de séparation des branches

on choisit de fixer une des variables relaxées et on crée deux nouveaux noeuds.

3.1.4 Résolution des sous problèmes

Le sous problème associé à une solution partielle du problème $(MILP)$ est le problème restreint obtenu en fixant une variable discrète dans la solution partielle.

$$\begin{cases} z = \min c_1^T x + c_2^T y \\ A_1 \begin{pmatrix} x \\ y \end{pmatrix} = b_1 \ \wedge \ A_2 \begin{pmatrix} x \\ y \end{pmatrix} \leq b_2 \\ y_{\min} \leq y \leq y_{\max} \quad x \in \{0,1\} \end{cases} \qquad \begin{array}{l} solution\ partielle \\ \rightarrow \\ \mathcal{F}^{(k)} = ensemble\ fix\acute{e} \end{array} \qquad \begin{cases} z = \min c_1^T x + c_2^T y \\ A_1 \begin{pmatrix} x \\ y \end{pmatrix} = b_1 \ \wedge \ A_2 \begin{pmatrix} x \\ y \end{pmatrix} \leq b_2 \\ y_{\min} \leq y \leq y_{\max} \quad x \in \{0,1\} \\ x_i \begin{cases} fix\acute{e} & i \in \mathcal{F}^{(k)} \\ \in \{0,1\} & autrement \end{cases} \end{cases}$$

on formule et on résout la relaxation (LP) du sous problème.

Solution meilleure (en anglais incumbent solution)

C'est la meilleure solution admissible trouvée durant l'évaluation du noeud k au cour de la procédure B&B (pour la valeur de la fonction objectif). La meilleure solution pourra être trouvée en analysant le noeud actuel ou bien les noeuds précédents. la solution meilleure obtenue nous offre une borne supérieure pour le $(MILP)$.
La procédure B&B est efficace lorsqu'on peut sonder quelque sous-problèmes à un niveau précoce, soit en exploitant la relaxation (LP) des sous-problèmes, soit en exploitant les solutions meilleures.

Sondage des sous-problèmes

1. La relaxation (LP) du sous-problème n'a pas de solution optimale ; procédure terminée par sondage du noeud.
2. La valeur de la solution optimale n'est pas meilleure que celle déjà trouvée ; procédure terminée par dominance de la meilleure valeur.
3. La solution du sous-problème relaxé est binaire alors :

- soit c'est la solution optimale du sous-problème ;
- sinon, il convient d'actualiser la meilleure solution.

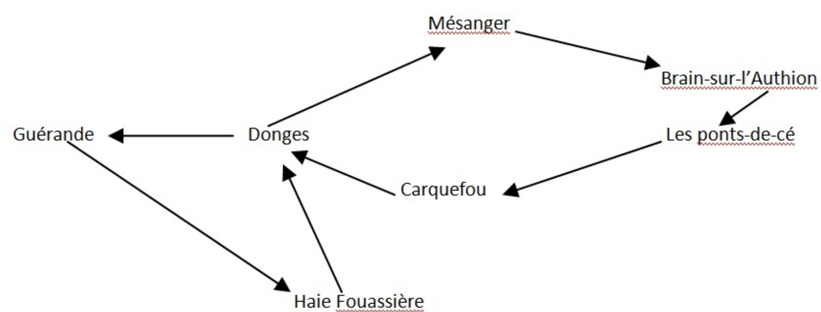
Autrement, il faut continuer le branchement.

Arrêt de la recherche B&B :

La procédure sera terminée quand tout les noeuds seront séparés ou sondés.
-La meilleure solution finale trouvée est l'optimum global pour le $(MILP)$ dans le cas échéant ; sinon le problème n'a pas de solution (impossible).
-On peut arrêter la recherche B&B si on estime la solution trouvée suffisamment proche de l'optimum :

$$\frac{z_{rel}^{(k)} - z_{inc}^{(k)}}{\frac{1}{2}\left| z_{rel}^{(k)} + z_{inc}^{(k)} \right|} \leq \epsilon_r$$

avec ϵ_r l'erreur utilisée.
Nous avons utilisé l'environnement de calcul Matlab pour résoudre le problème de distribution de fioul (2.1).Posé au problème comme fonction d'évaluation on a utilisé la fonction Matlab Linprog, qui utilise la méthode 'Active-Set' comme algorithme de programmation linéaire.La solution optimale consiste à faire deux tournées. La première dessert Guérande, puis la Haie Fouassiére. La seconde livre en premier le client de Mésanger, puis ceux de Brain-sur-l'Authion et des ponts-de-Cé pour terminer par celui situé à Carquefou. Dans la première tournée, 22000 litres de fioul sont livrés 37000 dans la seconde. Les tournées sont représentées sur la figure ci dessous. Le nombre total de kilomètres parcourus est 497.



CONCLUSION

Après avoir formulé le problème de distribution de fioul comme un problème $MILP$, nous avons essayé d'appliquer la méthode $B\&B$ pour le résoudre. Les tests numériques réalisés montrent que cette méthode est sensible par rapport au choix de la méthode qui résout la relaxation LP, à titre d'exemple et malgré que le problème est de petite taille ($n = 7$, un dépôt et 6 villes), la méthode Active-Set donne des résultats meilleurs que le simplexe ou une méthode de point intérieur.

En effet, le problème de livraison de fioul est un cas typique du problème de tourné de véhicule VRP (En anglais : vehicle routing problem). La méthode présentée ici convient seulement à des problèmes de petites taille (20 à 30 client). Il existe des méthodes arborescentes spécialisées pouvant résoudre d'une manière optimale des cas à 100 clients [8] [1], y compris si les clients doivent être livrés dans des fenêtres horaires fixées[5]. Au-delà de 100 clients, il est conseillé d'utiliser des méthodes heuristiques comme celle de Clark et Wright[4] . Une synthèse d'heuristiques classiques figure chez Cristofides [3]. Des métaheuristiques comme la méthode taboue trouvent de très bonnes solutions pour des problèmes de grandes taille [6].

Bibliographie

[1] **L. Bianco,** A.Mingozzi, S. Rcciardelli.- A set partionning approch to the multiple depot vehicle sheduling problem, Optimisation methods and software, 3,163-194, 1994.

[2] **B. Chachuat** : Mixed Linear Programming (MILP) Branch-Bound-Search, McMaster university.

[3] **N. Christofides, A. Mingozzi, P. Toth.** - Loading problems, pp. 339-369 dans combinatriol optimisation,coordonné par N. Christofides et al, Wiley, 1979.

[4] **G. Clarke,J. W. Wright -** Schelding of vehicles from a central depot to a number of delivery points, operations research, 12(4), 568-581, 1964.

[5] **M. Desrochers, J. Desrosiers, M. Slomon**.-A new optimisation algorithm for the vehicle routing problem with time windows, Operations research, 40(2), 342-354, 1992.

[6] **M. Gendreau, A. Hertz, G. Laporte**. -A tabu search algorithm for the vehicle routing problem, Management Sience, 40, 1276-1290, 1994.

[7] **C. Gueret, Marc Sevaux, Christian Prins et Jean Cristophe Culioli :**Programmation linéaire, 65 Problèmes d'optimisation modelisés et résolus avec Visual Xpress Paris 2000

[8] **G. Laporte**, M. Desthrokers, Y. Nobert. -Tow exact for the distance-constrained vehicle routing problem, networks, 14, 161-172, 1984

[9] **G. L. Nemhauser et L. A.Wolsey**.Integer and Combinatorial Optimisation.Wiley, New York 1998..

[10] **C. H Papadimitriou et K. Steiglitz** : Combinatorial Optimisation : Algorithms and Complexity, Dover publication, INC.

[11] **J. Teghem** : Programmation linéaire. Editions de l'Université de Bruxelles (2003).