

MINISTÈRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI, TIZIOUZOU
FACULTE DES SCIENCES
DEPARTEMENT DE MATHEMATIQUES

MEMOIRE DE FIN D'ETUDE

En vue de l'obtention du diplôme Master
en Recherche Opérationnelle

Sujet :

*Optimisation non linéaire avec contraintes,
méthodes d'optimisation méta heuristiques et
problème de routage de véhicules : Applications
aux problèmes pétroliers, cas de l'entreprise
SONATRACH*

Présenté par

M. Karim BENALIA

Encadré par

Mr. Mohamed Aidene

Devant le jury d'examen composé de:

M. OUANES Mohand

UMMTO

Président

M. AIDENE Mohamed

UMMTO

Rapporteur

M. MERKEB Abdelkader

UMMTO

Examinateur

Septembre 2014



Louange à dieu, le miséricordieux, sans lui rien de tout cela n'aurait pu être.

Je tiens tout d'abord à témoigner toute ma reconnaissance à mon promoteur **M. Mohamed AIDENE** pour avoir accepté d'encadrer mon travail et pour tous ses précieux conseils.

J'exprime également ma profonde gratitude à **Mr Hakkar** Chef de Département Evaluations Amont et Transport à la SONATRACH, de nous avoir proposé le thème de mon mémoire, de son soutien et de son apport pertinent pour la réalisation de ce mémoire.

J'adresse également tous mes remerciements à l'ensemble des enseignants qui m'ont suivi pendant mon cursus.

Enfin, merci à tous ceux qui ont contribué de près ou de loin à ce mémoire, du point de vue scientifique ou administratif.



Je dédie ce modeste travail :

A mes très chers parents qui sont la source de
mon éducation, mon savoir et mes principes.

A mes très chers frères et sœurs pour leur soutien.

A mes beaux-frères et belles-sœurs.

A tous les gens que j'aime et dont je n'ai pas cité les noms.

Avant propos

Le travail présenté dans ce mémoire rentre dans le cadre d'obtention du diplôme Master 2 en Recherche Opérationnelle de l'Université Mouloud Mammeri de Tizi Ouzou (UMMTO).

Les travaux décrits dans ce manuscrit, sont réalisés dans le laboratoire de recherche et de contrôle MCE&SONATRACH en collaboration avec la société SONATRACH (Société Nationale de transport et de commercialisation des hydrocarbures), et plus précisément à la sous Direction Opération Production.

Ce travail a été possible grâce aux différentes équipes des services de la SONATRACH qui par leur diversité, complémentarité et compétences m'ont permis de bien mener mes travaux.



Sommaire.....	4
Introduction générale.....	10
Chapitre 1 : Présentation de l'organisme d'accueil et problématiques	
1.1. Introduction.....	14
1.2. Présentation de l'entreprise SONATRACH	14
1.2.1. Histoire de la Sonatrach.....	14
1.2.2. Sonatrach aujourd'hui.....	14
1.2.3. Organigramme général de la sonatrach.....	15
1.3. Généralités et problématiques.....	16
1.3.1. Problème 1 : Optimisation de la taille et de la structure d'une flotte maritime pour l'exportation des produits GPL	16
1.3.1.1. Définitions et généralités.....	16
1.3.1.1.1. Les gaz GPL.....	16
1.3.1.1.2. Types de livraisons.....	16
1.3.1.1.3 Types de transactions.....	17
1.3.1.1.4 Saisonnalité.....	17
1.3.1.1.5 Coûts de transport.....	18
1.3.1.1.6. Etude de marché.....	18
1.3.1.2. La problématique.....	20
1.3.1.2.1. Position du problème.....	18
1.3.1.2.2. Objectifs.....	25
1.3.2. Problème 2 : Optimisation sur les opérations d'exploration et d'exploitation du pétrole et du gaz de sonatrach.....	26
1.3.2.1. Problème 2.1 : Optimisation des déplacements des camions vibrateurs au niveau d'un chantier sismique pour l'exploration du pétrole et du gaz...26	26

Sommaire

1.3.2.1.1. Définitions et généralités.....	26
1.3.2.1.1.1. La prospection et exploration gazières et pétrolières.....	26
1.3.2.1.1.2. La sismique réflexion.....	27
1.3.2.1.1.3. Définition des paramètres d'acquisition en 3D.....	28
1.3.2.1.2. La Problématique.....	33
1.3.2.1.2.1. Etude de l'existant	33
1.3.2.1.2.2. Problème posé	33
1.3.2.1.3. Objectifs fixés.....	34
1.3.2.2. Problème 2 : Elaboration d'un schéma optimal de la fermeture des puits de pétrole et de gaz de Sonatrach.....	35
1.3.2.2.1. Définitions et généralités.....	35
1.3.2.2.1.1. Abandon définitif de puits	35
1.3.2.2.1.2. Pourquoi doit-on fermer les puits abandonnés	37
1.3.2.2.1.3. L'appareil Rig « Plate-forme de forage »	37
1.3.2.2.2. La problématique.....	37
1.3.2.2.3. Objectifs.....	38
1.4. Conclusion.....	38

Chapitre 2 : Modélisation mathématique

2.1. Introduction.....	40
2.2. Modélisation mathématique du problème 1.....	40
2.2.1. Les hypothèses.....	40
2.2.2. Les Contraintes.....	41
2.2.3. Modèle mathématique.....	43
2.2.3.1. Nomenclatures.....	43
2.2.3.2. Définition des variables.....	44
2.2.3.3. Définition de la fonction objective.....	44
2.2.3.4. Définition des contraintes.....	45
2.2.3.5. Résumé du modèle.....	45
2.2.4. Identification du modèle.....	46

Sommaire

2.3. Modélisation mathématique du problème 2	46
2.3.1. Modélisation mathématique du problème 2.1	46
2.3.1.1. Modèle mathématique	46
2.3.1.1.1. Définitions des données	46
2.3.1.1.2. Définition des variables	47
2.3.1.1.3. Les contraintes du problème	48
2.3.1.1.4. Fonction objective	49
2.3.1.5. Résumé du modèle	51
2.3.1.2. Identification et évaluation du modèle	51
2.3.2. Modélisation mathématique du problème 2.2	52
2.3.2.1. Définitions des variables et des données	52
2.3.2.2. Le modèle mathématique	52
2.4. Conclusion	53

Chapitre 3 : Optimisation combinatoire et méthodes d'optimisation métahéuristiques : théories et méthodes de résolution du problème 1

3.1. Introduction	55
3.2. Optimisation combinatoire	56
3.2.1. Problème d'optimisation combinatoire	56
3.2.2. La théorie de la complexité	56
3.2.2. Les méthodes d'optimisation combinatoire	58
3.3. Intérêt et domaines d'applications des méthodes heuristiques	59
3.4. Méthode de recherche Tabou	60
3.4.1. Principe de la méthode de Recherche Tabou	61
3.4.2. Algorithme général de la méthode Recherche Tabou	62
3.5. Les algorithmes génétiques	64
3.5.1. Principe des algorithmes génétiques	64
3.5.2. Les composantes d'un algorithme génétique	65
3.5.3. Description détaillée des éléments de l'algorithme génétique	66

Sommaire

3.5.4. Critère d'arrêt d'un algorithme génétique	69
3.6. Conclusion.....	69

Chapitre 4 : Programmation mathématique non linéaire et Problème de V.R.P : Théories et méthodes de résolution du problème 2

4.1. Introduction.....	71
4.2. Programmation mathématique non linéaire avec contraintes : théories et méthodes du résolutions du problème 2.1.....	71
4.2.1. Notions fondamentales	71
4.2.1.1. Eléments de topologie	72
4.2.1.2. Convergence.....	72
4.2.1.3. Rappels de calcul différentiels.....	73
4.2.1.4. Rappels de quelques outils d'analyse convexes.....	74
4.2.2. Programmation mathématique.....	75
4.2.2.1. Formulation générale de programmation mathématique	76
4.2.2.2. Classification d'un programme mathématique	76
4.2.2.3. Qualification des contraintes	77
4.2.2.4. Résolution d'un programme mathématique.....	77
4.2.2.4.1. Principaux résultats d'existence.....	77
4.2.2.4.2. Conditions d'optimalité.....	78
4.2.2.4.2.1. Cas sans contraintes.....	78
4.2.2.4.2.2. Cas avec contraintes.....	79
4.2.3. Aperçu sur les méthodes de programmation mathématique avec Contraintes.....	81
4.2.3.1. Méthode des directions admissibles	82
4.2.3.1.1. La méthode de Frank et Wolfe	83
4.2.3.2. Les méthodes de pénalité intérieure (ou méthodes de barrière)	84
4.2.3.3. Les méthodes de pénalité extérieure.....	85
4.2.3.4. Un simple exemple d'applications.....	87
4.3. Problème de VRP : Théories de résolution du problème 2.2.....	88
4.3.1. Aperçu sur la Programmation linéaire en nombres entiers	88

Sommaire

4.3.1.1. Formulation mathématique d'un programme linéaire en nombres entiers.....	89
4.3.1.2. Les méthodes de séparation et évaluation	89
4.3.1.2.1. Principe de la méthode.....	89
4.3.2. Rappels sur la théorie des graphes.....	92
4.3.2.1. Un bref historique de la théorie des graphes.....	92
4.3.2.2. Définition d'un graphe.....	93
4.3.2.3. Représentation d'un graphe.....	94
4.3.2.4. Quelques concepts fondamentaux.....	94
4.3.2.4.1. Principales définitions.....	95
4.3.2.4.2. Connexité dans un graphe.....	96
4.3.2.4.3. Arbre et arborescence	96
4.3.2.4.4. Parcours eulériens et hamiltoniens	96
4.3.3. Problème de voyageur de commerce (PVC).....	97
4.3.3.1. Définition.....	98
4.3.3.2 Formulation mathématique	99
4.3.4. Le problème de tournées de véhicules (VRP) et ces extensions.....	99
4.3.4.1. VRP de base.....	99
4.3.4.2. Quelques aspects pratiques du VRP	100
4.3.4.3. Présentation de quelques méthodes de résolution du VRP de base....	102
4.3.4.3.1. Les méthodes exactes.....	102
4.3.4.3.2. Les méthodes approchées	103
4.4. Conclusion.....	106

Chapitre 5 : Résolution du problème 1 et réalisation d'un logiciel

5.1. Introduction.....	108
5.2. Choix des méthodes	108
5.3. Justifications du choix	108
5.4. Adaptation de la méthode génétique.....	109
5.4.1. Codage de la solution et des données	109
5.4.2. Procédure Recherche de solutions réalisables	113
5.4.3. Justification de l'algorithme SolReal	114

Sommaire

5.4.4. Paramètres de l'algorithme Génétique (Population initiale)	116
5.4.5. Adaptation des opérateurs	117
5.5. Adaptation de la méthode de la recherche tabou	118
5.5.1. Le codage de la solution.....	118
5.5.2. Génération d'une solution réalisable.....	118
55.3. Génération des solutions voisines.....	119
5.5.4. Listes Tabous adaptées à notre problème.....	120
5.5.5. Critère d'aspiration et critère d'arrêt.....	121
5.5.6. Algorithme Recherche Tabou Adapté à notre problème	122
5.6. Réalisation d'un logiciel pour notre problème 1.....	127
5.6.1. Qu'est-ce que Delphi ?.....	127
5.6.2. Qu'est ce qu'une application MDI?.....	127
5.6.3. Présentation du notre logiciel « Flot GPL ».....	128
5.7. Conclusion.....	133

Chapitre 6 : Approche de résolution du problème 2 et réalisation d'un logiciel d'application

6.1. Introduction.....	135
6.2. Résolution du problème 2.1	135
6.2.1. Résolution par des méthodes exactes.....	135
6.2.2. Résolution par des heuristiques.....	136
6.2.3. Implémentation et résultats.....	138
6.2.3.1. Langage de programmation C++Builder.....	139
6.2.3.2. Présentation du notre logiciel et résultats.....	139
6.3. Résolution du problème 2.2.....	143
6.4. Conclusion	146
Conclusion générale	147
Annexe.....	150
Bibliographie.....	152

Introduction générale

La recherche opérationnelle consiste en l'application des méthodes scientifiques pour résoudre les problèmes complexes rencontrés dans la direction et la gestion de grands systèmes d'hommes, de machines, de matériaux et d'argent dans l'industrie, le commerce, l'administration et la défense.

C'est pour cela que beaucoup d'entreprise s'intéressent à cette discipline, dont le but est d'aider les directions à déterminer leur politique de manière scientifique. C'est dans ce contexte que l'entreprise Sontarach nous a proposé deux problèmes de la recherche opérationnelle :

Problème 1 : « Détermination de la taille et de la structure d'une flotte maritime à affréter pour l'exportation des produits GPL de l'entreprise Sonatrach »

La production nationale de GPL toutes origines confondues (raffinage, liquéfaction et production aux champs) atteindrait plus de 14 millions de tonnes en l'année 2015 grâce au plan de développement des filières Amont et Aval. L'exportation est estimée à 11 millions de tonnes à cet horizon. Ces perspectives ont amené Sonatrach à préconiser une politique axée sur quatre grands volets :

1. privilégier les contrats coût et fret (CFR) aussi bien vers les marchés traditionnels que vers les nouveaux marchés et établir des relations directes avec les utilisateurs finaux ;
2. cibler les marchés les plus valorisants, particulièrement la méditerranée ;
3. être un acteur sur le marché à travers une présence active sur le marché spot ;
4. diversifier les débouchés.

La tendance concurrentielle du marché international confirme la nécessité de se doter d'une stratégie de commercialisation des GPL. Cette stratégie lui permettra d'assurer une fluidité dans l'écoulement de son exportation et une valorisation optimale pour ses produits. Comme le frais de transport des GPL occupe une bonne part dans les coûts d'exportation de ces produits, alors une identification en moyens de transport s'impose pour la sonatrach. D'où la détermination de la taille et la structure d'une flotte maritime optimale qui fera l'objet de ce première problème.

Introduction générale

Problème 2 : Optimisation des opérations d'exploration et d'exploitation du pétrole et du gaz

Ce problème est décomposé en deux sous problèmes :

Problème 2.1. (Exploration) : Optimisation des déplacements des camions vibrateurs au niveau d'un chantier sismique pour l'exploration du pétrole et du gaz

Sonatrach possède une expérience de plus de 40 ans dans le domaine de la recherche des hydrocarbures par sismique réflexion, une des méthodes les plus utilisées en Algérie et dans le monde. Des ingénieurs et techniciens hautement qualifiés ont permis à l'entreprise d'atteindre une maîtrise parfaite dans l'utilisation d'instruments de dernière technologie tels que les laboratoires télémétriques à grand nombre de canaux récepteurs, les camions vibrateurs et les récepteurs G.P.S pour la sismique 2D et 3D. L'entreprise est dotée d'équipements de dernière technologie pour faire face à toute demande de ses clients dans le domaine de l'acquisition sismique. C'est dans ce contexte que l'entreprise Sonatrach nous a proposé le sujet s'intitulant : « Optimisation des déplacements des camions vibrateurs au niveau d'un chantier sismique pour l'exploration du pétrole et du gaz ».

Problème 2.2 (Exploitation) : Elaboration d'un Schéma optimal de la fermeture des puits de pétrole et de gaz

L'abandon et la fermeture des forages constituent donc un aspect à part entière dans la protection de l'environnement et en particulier dans la protection des ressources d'eau minérale. Les techniques impliquées pour ces actions nécessitent une analyse détaillée du contexte hydrogéologique et des conditions de réalisation d'un ouvrage pour répondre à chaque cas particulier.

L'abandon d'un ouvrage est donc un problème sérieux qu'il faut traiter avec soin. Sur le principe, il s'agit d'une opération qui entre dans le cadre très large de la politique de l'environnement (conservation des ressources).

C'est ainsi que Sonatrach se soucie constamment de l'impact des coûts dans la fermeture des puits de pétrole et de gaz.

L'objectif de chaque opération d'abandon de puits est de parvenir à une étanchéité de surface satisfaisante, tout en réduisant les risques de fuites. Les mouvements de fluides dans le puits

Introduction générale

sont limités jusqu'au rétablissement naturel de l'équilibre de pression qui existait avant le forage, afin de protéger les réserves restantes, notamment les réserves d'eau dans les aquifères proches, et l'environnement. Une isolation totale inclut les opérations de cimentage primaires, car la qualité du ciment entre le casing et la formation rocheuse est un facteur clé d'un abandon de puits durablement fiable. Les principaux critères de choix des matériaux de fermeture sont l'intégrité à long terme, et une faible perméabilité.

Le groupe Sonatrach est chargé de forage et de fermeture des puits de pétrole et de gaz il fait de son mieux pour gérer le réseau de déplacement des appareils « the well closing » en poursuivant un certain effort technologique et économique pour pouvoir organiser rationnellement au moindre coût et au meilleur service rendu cette distribution.

Pour ce faire, il est nécessaire d'introduire deux concepts qui se sont forgé une place importante dans le domaine économique : l'organisation et l'optimisation.

C'est dans ce but, que la direction Amont de l'entreprise Sonatrach nous a confié le travail à effectuer dans le cadre de ce mémoire et qui consiste à établir un programme de parcours des appareils optimisant le coût et le temps total de l'opération. Notre tâche consiste à concevoir un outil d'aide pour la gestion des appareils chargés de fermer les puits abandonnés.

A présent nous présentons notre travail sous les chapitres suivants :

Le premier chapitre est consacré à la présentation de l'entreprise et les problématiques.

Le deuxième chapitre consiste en la modélisation mathématique du problème 1 et 2.

Le troisième chapitre est réservé aux méthodes de résolution pour le problème 1.

Le quatrième chapitre est réservé aux méthodes de résolution pour le problème 2.

Le cinquième chapitre est consacré à la résolution du problème 1 et réalisation d'un logiciel d'application.

Et dans le sixième chapitre nous avons décrit l'approche de résolution adoptée au Problème 2 et réalisation d'un logiciel d'application.

Enfin nous terminerons par une conclusion générale.



Chapitre 1

*Présentation de l'organisme d'accueil et
problématiques*



1.1. Introduction

La première partie de ce chapitre est consacrée à la présentation de l'entreprise d'accueil. Dans la deuxième partie, nous allons expliquer quelques définitions et généralités pour chaque problème posé. Dans la troisième partie, nous allons expliquer les problèmes proposés par l'entreprise et les objectifs attendus.

Pour la bonne compréhension du mémoire, il est utile de présenter dans ce chapitre les définitions de quelques notions de base et les équipements nécessaires pour chaque problème posé.

1.2. Présentation de l'entreprise Sonatrach

1.2.1. Historique de la Sonatrach [01]

La société nationale de transport et de commercialisation des hydrocarbures SONATRACH a été créée le 31 décembre 1963 pour assurer la responsabilité de la production, du transport et de commercialisation des hydrocarbures.

Le développement de la SONATRACH est passé par plusieurs phases :

- ❖ **Phase 1 : 1963/1971 « Cr éation et constitution de Sonatrach »**
- ❖ **Phase 2 : 1971/1982 « Phase de croissance et d'intégration »**
- ❖ **Phase 3 : 1982/1987 « la restructuration et l'essaimage de Sonatrach »**
- ❖ **Phase 4 : 1987/1998 « la maturité et la modernisation »**
- ❖ **Phase 5 : 1998/2000 « les nouveaux statuts organisant Sonatrach en société par actions (S.P.A) »**

1.2.2. Sonatrach aujourd'hui [01],[02]

Sonatrach est la compagnie algérienne de recherche, d'exploitation, de transport par canalisation, de transformation et de commercialisation des hydrocarbures et de leurs dérivés. Elle intervient également dans d'autres secteurs tels que la génération électrique, les énergies nouvelles et renouvelables et le dessalement d'eau de mer. Elle exerce ses métiers en Algérie et partout dans le monde où des opportunités se présentent. Sonatrach est la première entreprise du continent africain. Elle est classée 11ème parmi les compagnies pétrolières mondiales, 2ème

Chapitre 1. Présentation de l'organisme d'accueil et problématiques

exportateur de GNL et 3^{ème} exportateur de gaz naturel. Elle emploie 150000 personnes dans l'ensemble du groupe.

Aujourd'hui, Sonatrach ne conçoit pas de développement économique sans un développement durable.

L'activité de SONATRACH s'articule autour de quatre branches principales :

- 1- Activité amont** (Exploration, Recherches et développement, Production et forage)
- 2- Activité aval** (Liquéfaction du gaz naturel, Raffinage du pétrole et pétrochimie)
- 3- Activité transport par canalisation**
- 4- Activité commercialisation**

1.2.3. Organigramme général de la Sonatrach

L'organigramme suivant ci-dessous présente l'ensemble de l'organisation de la société

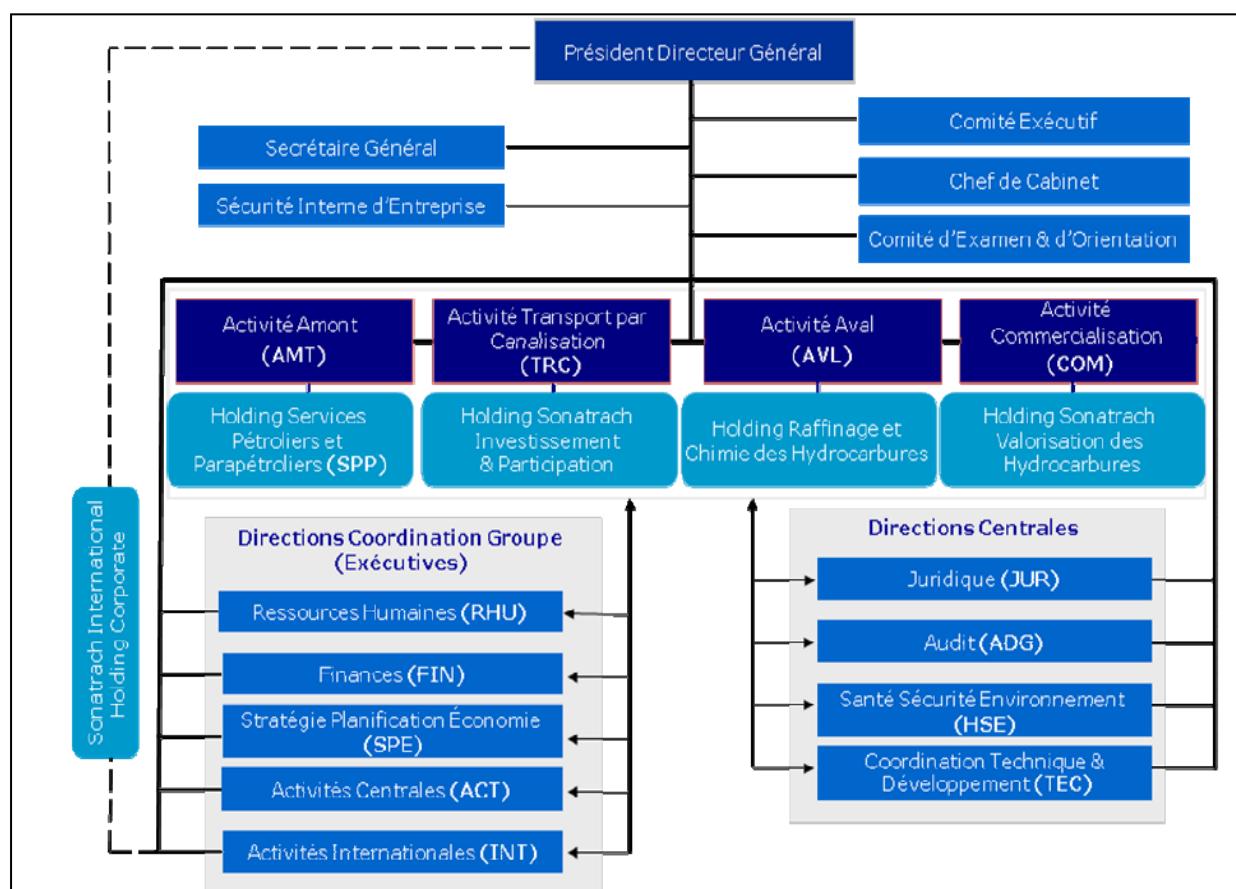


Fig.1.1. Organigramme général de la Sonatrach

1.3. Généralités et problématiques

1.3.1. Problème 1 : Optimisation de la taille et de la structure d'une flotte maritime pour l'exportation des produits GPL

1.3.1.1. Définitions et généralités

1.3.1.1.1. Les gaz GPL [04]

Les G.P.L (Gaz des pétroles liquéfiés) sont deux hydrocarbures, le propane C₃H₈ et le butane C₄H₁₀. Ces corps sont à l'état gazeux aux conditions normales de température et de pression et peuvent être facilement liquéfiables par abaissement de la température à -42°C pour le propane et -6°C pour le butane, ou par compression modérée.

Décomposition molaire des GPL

C ₂ H ₆	C ₃ H ₈	iC ₄ H ₁₀	nC ₄ H ₁₀	iC ₅ H ₁₂	nC ₅ H ₁₂
1.83%	60.62%	13.61%	23.55%	0.34%	0.05%

Tab.1.1 Décomposition molaire des GPL.

Origines : Les GPL sont obtenus à partir de :

- 1. Les GPL à partir des champs :** Les GPL sont produit au sud à partir des champs pétroliers de Hassi Messaoud Nord et Sud, Haoud Berkaoui et à partir du traitement de gaz à Hassi R'mel.
- 2. Production à partir des unités GNL :** Trois unités de liquéfaction de gaz naturel produisent des GPL, il s'agit de GL1K Skikda, GL2K Béthioua et GL4Z Arzew.
- 3. Production des raffineries :** La production des raffineries varie en fonction des quantités de brut traitées.

La totalité de la production est orientée vers le marché national.

1.3.1.1.2. Types de livraisons [05]

Sonatrach dispose de plusieurs formules de livraisons :

❖ **F.O.B. " Fret On Bord " ou " Franco en Bord " :**

Signifie que le vendeur a rempli ses obligations de livraison, quand la marchandise passe le bastingage du navire au port d'embarquement désigné. Cela signifie que l'acheteur doit supporter tous les frais et les risques de perte ou de dommage que peut courir la marchandise à partir de ce point.

❖ **C.F.R. " Cost And Freight " (en Français Coût et Fret) :**

Signifie que le vendeur doit payer tous les frais y compris le fret nécessaire pour acheminer la marchandise au port de destination désigné.

❖ **C.I.F. " Cost Insurance Freight " (en Français Coût Assurance Fret " :**

Signifie que le vendeur a les mêmes obligations que dans le cas précédent, perte ou de dommage que peut courir la marchandise au cours du transport. Le vendeur contracte avec l'assureur et paie une prime d'assurance.

1.3.1.1.3. Types de transactions [05],[03]

Le marché des GPL se caractérise par deux types de ventes ou de transactions :

1. Par Contrat :

Un contrat prévoit :

- Les quantités annuelles de base et les prix de facturation ;
- Les types de livraisons et les programmes de livraisons ;
- La durée du contrat : Elle est généralement d'une année ;
- La destination du produit ou les ports de déchargement.

2. En spot :

Signifie vente du moment et du lieu où les deux parties se mettent d'accord sur toutes les conditions du contrat. Ces ventes sont issues de cinq centres de production :

- Le nord ouest Europe ;
- L'Amérique (principalement les USA) ;
- L'Afrique (principalement l'Algérie) ;

- Le Moyen Orient (principalement l'Arabie Saoudite).

1.3.1.1.4. Saisonnalité [05]

La consommation des GPL varie selon deux périodes : Hiver, Eté.

- **Hiver** : (Octobre à Mars) : Il se caractérise par une forte demande domestique à cause du froid qui touche les grands centres de consommation (Europe, USA, Japon) ;
- **Été** : (Avril à Septembre) : La consommation domestique des GPL diminue en tirailleur une baisse des prix favorisant la substitution du Naphta par les GPL dans l'industrie pétrochimique. Les GPL ne sont utilisés dans la pétrochimie qu'à condition que leurs prix soient égaux à environ 85 pour cent de celui du Naphta.

1.3.1.1.5. Coût de transport

Les G.P.L exportés à partir de l'Algérie sont vendus en CIF&CFR, leurs prix de facturation sont composés d'une partie FOB et d'une partie 'fret' qui est le coût de transport.

L'élément 'fret' est déterminant pour la pénétration d'un produit sur un marché donné.

1. Les frais d'exploitation

Se décomposent en deux parties

- une partie fixe ou semi fixe telle que les frais du personnel, assurance, taxes ;
- Une autre proportionnelle au tonnage tel que le remorquage, et les services divers.

2. Consommation du fuel

- Fuel oil en mer et aussi diesel oil au port.

3. Coût d'affrètement des navires.

1.3.1.1.6. Etude de marché :

L'Algérie dans le marché international :

Deuxième exportateur mondial des GPL derrière l'Arabie Saoudite, L'Algérie a exporté plus de 7,9 millions de tonnes de GPL en 2013. Sur les quatre continents, 23 pays ont été approvisionnés. La Méditerranée représente le principal débouché de Sonatrach, avec plus de

80% des exportations. Elle est suivie par les USA, l'Amérique Latine, l'Asie et l'Europe du Nord [05], [07].

Les exports par région :

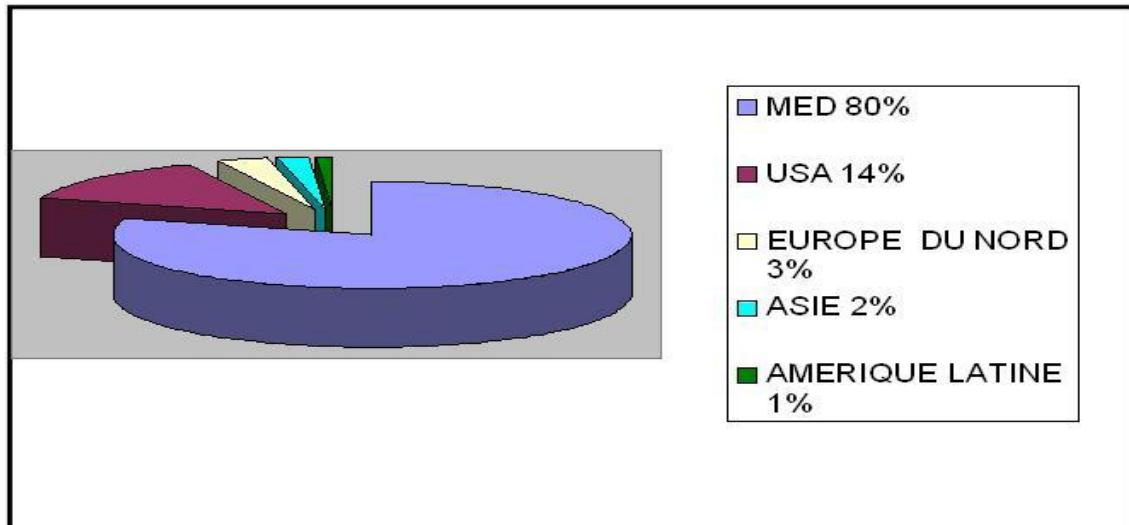


Fig.1.2. Organigramme Des Exportations Par Région

Perspectives d'exports des gaz GPL par région à l'horizon 2015



Fig.1.3 Carte Des Exportations Par Région

Perspectives en matière de transport maritime [06]

Sonatrach Pétroleum Corporation (SPC), négocie l'achat de navires pour le transport des gaz de pétrole liquéfiés (butane et propane). La compagnie, qui est une filiale à 100% de la Sonatrach, possède déjà sept transporteurs de G.P.L. qui sont :

- Jemila (mise en service 1983) : capacités (8000 m³) ;
- Nejma (mise en service 1983) : capacités.(57000 m³) ;
- Reggane (mise en service 1999) : capacités.(84000 m³) ;
- Djanet (mise en service 2000) : capacités(84000 m³) ;
- Alrar (mise en service 2004) : capacités(59000 m³) ;
- Rhoud Nouss (mise en service 2004) : capacités (59000 m³) ;
- Hassi Messaoud (mise en service 2005) : capacités (59000 m³).

Elle affrète aussi de façon permanente une quinzaine de navires pour le transport des hydrocarbures. Par ailleurs, une soixantaine de bateaux par an sont affrétés ponctuellement pour des échanges spot.

1.3.1.2. La problématique

Compte tenu de l'augmentation constante de la production de Sonatrach en produits GPL, ainsi que pour tirer le meilleur produit de ces produits, Sonatrach a décidé dans la mesure du possible de vendre ses GPL en (CIF, CFR).

1.3.1.2.1. Position du problème :

Pour vendre ses produits en (CIF, CFR), Sonatrach a besoin de déterminer avec précision la taille et la structure de la flotte nécessaire pour assurer le transport des produits GPL sur les deux saisons (hivernale et estivale), avec le meilleur coût possible.

Cette flotte sera chargé de transporter les quantités vendues, dans le cadre des ventes par contrat (CIF, CFR), ces quantités seront prélevées essentiellement des ports d'Arzew et Batioua pour les transporter directement vers des ports de déchargement bien définis, qui ont une demande mensuelle avec une marge de flexibilité de livraison bien définie.

Chapitre 1. Présentation de l'organisme d'accueil et problématiques

Les demandes mensuelles de chaque client, et les marges de flexibilité de la livraison de ces demandes mensuelles, sont stipulées dans les contrats annuels de vente.

Les deux gaz GPL (Butane, Propane) seront transportés dans des réservoirs séparés. Un navire peut transporter un seul ou les deux types de gaz en même temps.

Une quantité de produit chargée sur un navire, est considérée livrée au client, dès que ce navire reçoit le document prouvant que cette quantité a passé le bastingage de ce dernier, et elle est prête pour être transporté vers ce client, la date de parution de ce document est considérée comme la date de livraison.

Les ports de chargements :

Sonatrach dispose de 3 ports de chargement :

- 1- **Arzew** : Equipé de 3 postes de chargement S1, S2 et S3 permettent de charger des navires de 1000 à 20 000 tonnes.
- 2- **Betioua** : Equipé de 2 postes de chargement D1 (3 000 à 45 000 tonnes) et M6 (supérieur à 30 000 tonnes).
- 3- **Skikda** : Un poste de chargement (P5).

Les navires :

Sonatrach dispose de sa propre flotte de transport des GPL, mais cette dernière n'est pas suffisante pour assurer l'évacuation de toute sa production destinée à l'exportation. Pour assurer la livraison de ses produits en (CIF, CFR), Sonatrach doit affréter la majorité des GPLiers¹

➤ L'acceptabilité des navires :

Les navires sont acceptés ou non, selon les capacités de chargement seulement, car les navires d'une même capacité possèdent aussi, (à-peu-près) le même tirant d'eau², et la même longueur. Le tirant d'eau et la longueur du navire doivent être respectivement inférieurs, à la profondeur et à la longueur du poste à quai de chargement.

¹ Navires de transport des gaz GPL

² La hauteur de la partie submergée du navire

➤ Temps de chargement des navires :

Le temps de chargement des navires varie en fonction de la taille de ces derniers, il est mesuré en heure, et il varie de 16h pour les navires de 4.000 tonnes jusqu'à 30h pour ceux de 42.000 tonnes.

Le tableau (Tab.1.2) résume l'acceptabilité de chaque classe de navire pour chaque poste de chargement, et donne les temps de chargement.

➤ Durées de rotations des navires :

Les rotations sont définies par un aller-retour entre le port de chargement et une destination de déchargement, car les navires ne peuvent pas être chargés partiellement pour des raisons techniques.

Les durées de rotation entre le port de chargement et les différents ports de déchargement sont connues, elles sont différentes d'une classes de navire à une autre, l'unité de mesure des durées de rotation est le jour, elles varient entre 4 à 37 jours, d'après la destination, la vitesse du navire, ainsi que le temps de déchargement.

Taille des navires (Par Tonne)	Postes de chargement				
	S1 (Heures)	S2 (Heures)	S3 (Heures)	D1 (Heures)	M6 (Heures)
4.000	20	20	20	16	-
8.000	22	22	22	20	-
17.000	-	-	-	21	-
22.000	-	-	-	24	22
32.000	-	-	-	30	26
42.000	-	-	-	-	30

Tab.1.2. Temps de chargement

➤ Affrètement des navires :

L'affrètement des navires se fait généralement sur une année, cette formule est la moins coûteuse, car les affrètements par voyage reviennent beaucoup plus cher.

Les ports de déchargement :

Le tableau (Tab.1.3) nous renseigne sur l'ordre de grandeur de notre problème, et nous donne les classes de navires acceptées par chaque port.

Pays	Ports	Taille Min	Taille Max
Brésil	Santos	32 .000	42.000
	Rocife	32.000	42.000
	Raranague	32.000	42.000
Espagne	Tarragone Carthagène	4.000	32.000
France	Lavera	4.000	32.000
Italie	Naples	4.000	8.000
	Brindisi	4.000	22.000
		4.000	22.000
Japon	Japon	32.000	42.000
Mexique	Pagaritos	32.000	42.000
Maroc	Agadir	4.000	4.000
	Djorf	4.000	4.000
	Mohammadia	4.000	22.000
	Nador	4.000	4.000
N.W.Europe	N.W.E	22.000	42.000
Portugal	Lisbonne	22.000	4.000
Turquie	Izmir	22.000	42.000
	Izmit	22.000	42.000
	Dortyol	22.000	42.000
USA	Cote Est(USEC)	32.000	32.000
	Golf du Mexique(USGC)	32.000	42.000

Tab.1.3. Ensembles des ports de destination

Les destinations de 2014 :

La figure (1.4) nous renseigne sur les ports de destinations pour l'année 2014

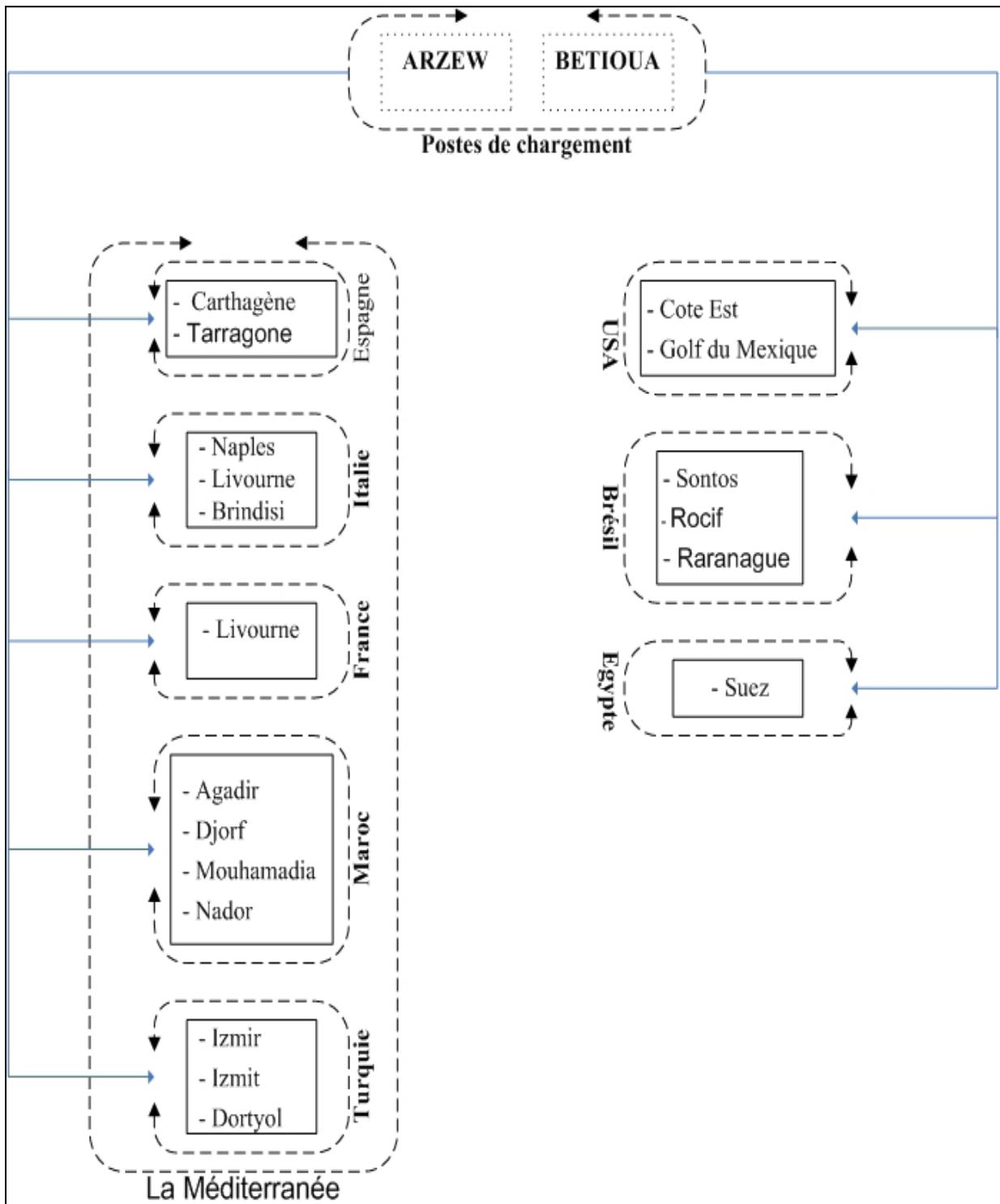


Fig.1.4. Ensemble des ports de destination pour l'année 2014

Description de la procédure actuelle

Actuellement, le plan d'affrètement contractuel se fait d'après les expériences des années précédentes, et de cette façon Sonatrach se retrouve soit avec un déficit, soit avec un excédant de navires, dans le cas d'un déficit elle a recourt à l'affrètement Spot (par voyage), qui est plus cher que l'affrètement par contrat, et dans le cas de l'excédant Sonatrach est obligée d'affréter ces navires à son tour pour les rentabiliser.

Notre contribution

Notre contribution consiste à déterminer la taille et la structure de la flotte nécessaire pour l'acheminement de toutes les exportations de Sonatrach vendu en (CIF, CFR), et la détermination pour chaque navire des dates de chargements vers chaque destination, qui permette de satisfaire les demandes mensuelles des clients avec une certaine marge de flexibilité, car une flotte ne peut pas être optimale pour tous les plans de transport.

1.3.1.2.2. Objectifs

Les objectifs principaux de notre étude sont :

- Identification de la taille et la structure de la flotte à affréter pour l'exportation des produits GPL;
- Détermination des dates de chargement de chaque navire vers les différents clients, permettant à cette flotte de livrer les GPL selon les lots mensuels stipulés dans les contrats annuels.

Nos objectifs nous renvoient à deux notions d'optimisation :

- 1- Maximisation des recettes ;
- 2- Minimisation des coûts de transport.

On note que le premier objectif est plus large que le deuxième en termes d'optimisation, cependant cette option fait intervenir la notion de prix qui à l'instar des autres prix des produits énergétiques varie dans le temps d'une façon imprévisible. A défaut de prévision rigoureuse quant à l'évolution des prix des GPL sur les différents marchés, cette alternative ne sera pas retenue comme fonction objectif dans notre étude.

Quand à la deuxième, elle peut être réalisée compte tenu de la disponibilité :

- ❖ Des prévisions sur l'évolution des coûts d'affrètement des navires ;
- ❖ Des données sur les infrastructures portuaires des marchés ciblés par Sonatrach.

Le problème se reformulera comme suite :

" Quelle serait la structure de la flotte à affréter susceptible de permettre à Sonatrach de réaliser son programme prévisionnel d'exportation en minimisant les coûts de transport"

Une fois la fonction objective a été retenue pour l'étude du problème, les contraintes de notre étude sont à définir.

1.3.2. Problème 2 : Optimisation des opérations d'exploration et d'exploitation du pétrole et du gaz de la Sonatrach

Ce problème est décomposé en deux sous problèmes :

1.3.2.1. Problème 2.1 : Optimisation des déplacements des camions vibrateurs au niveau d'un chantier sismique pour l'exploration du pétrole et du gaz

Avant d'entamer notre étude, il est nécessaire de commencer par analyser l'environnement du problème, afin de mieux maîtriser le domaine de notre travail. Pour cela, nous commencerons d'abord par faire connaissance avec la sismique réflexion.

Dans notre cas, l'acquisition sismique, et des termes essentiels utilisés dans ce problème, ainsi les données sismiques qui sont acquises au moyen d'un système qui comprend essentiellement : un dispositif d'émission (source), un dispositif de détection (géophone) et un laboratoire d'enregistrement.

1.3.2.1.1. Définitions et généralités

1.3.2.1.1.1. La prospection et exploration gazières et pétrolières

Prospection et exploration gazières et/ou pétrolières visent à découvrir de nouveaux gisements de gaz naturel ou de pétrole.

L'objectif de la prospection est de localiser un gisement. L'exploration doit en vérifier l'existence et en évaluer l'importance et la qualité grâce à des forages dont l'emplacement est

déterminé en associant géologie et géophysique. En cas de succès, ces deux phases en amont sont suivies par les phases d'exploitation, de transport et de commercialisation.

1.3.2.1.1.2. La sismique réflexion [08],[10]

La sismique réflexion est la méthode principale des géophysiciens pour repérer des gisements potentiels :

- sur terre (*onshore*), à partir d'un choc ou de vibrations sonores ébranlant le sol, on détecte par un réseau de géophones les échos réfléchis partiellement par les couches géologiques. On obtient ainsi une échographie 2D de la structure des couches prospectées.

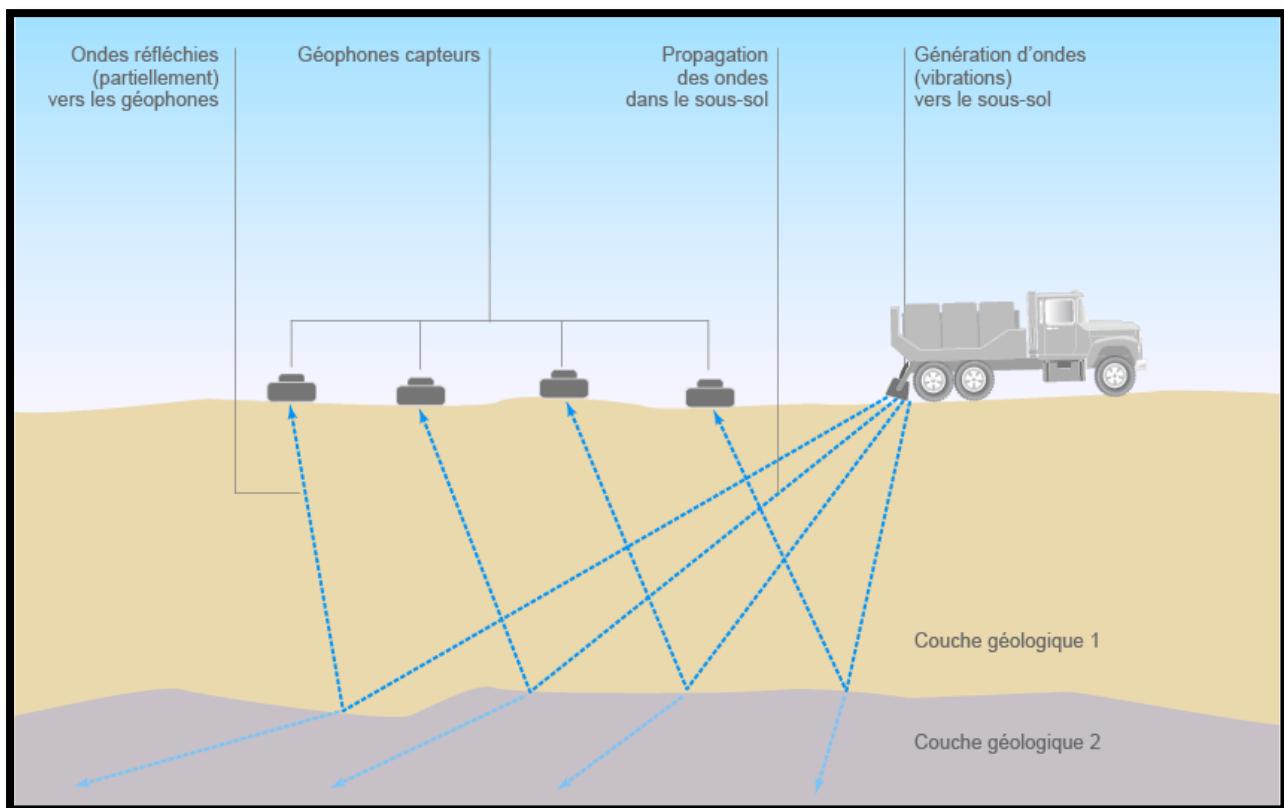


Fig. 1.5. Prospection géophysique sur terre

- En mer (*offshore*), on produit l'onde sismique par air comprimé à haute pression et on recueille les échos sur des hydrophones flottants (flûtes), la couche d'eau étant considérée comme homogène

Pour passer en sismique 3D, on multiplie les flûtes et les angles de production des ondes sismiques pour permettre de construire des images du sous-sol en volume.

Beaucoup plus onéreuse, l'imagerie sismique 3D est aussi beaucoup plus précise et permet de visualiser les volumes des gisements. En intégrant le facteur temps, on peut analyser l'évolution des gisements en cours d'exploitation en 4D.

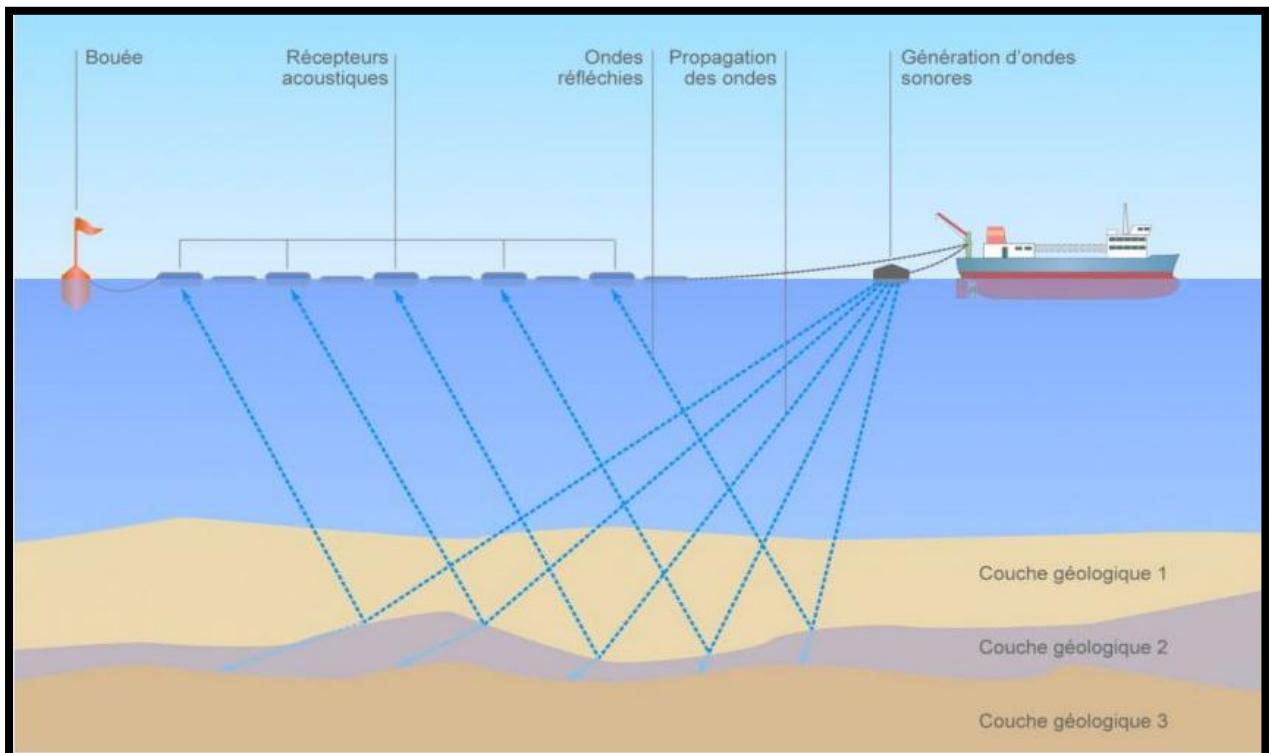


Fig.1.6. Prospection géophysique offshore

Principe de fonctionnement

La sismique réflexion est l'outil de base de la prospection pétrolière et son importance dépend de la précision et de la résolution, pour aboutir à une information sismique. Cette dernière qui passe par deux importantes étapes (**l'acquisition et le traitement**).

➤ Acquisition sismique

Systèmes d'acquisition : Ils comprennent essentiellement un dispositif d'émission, un dispositif de détection et un laboratoire d'enregistrement numérique.

➤ Traitement des données sismiques

Les données de terrain brutes enregistrées sur band magnétique sont envoyées au centre de calcul, afin de subir les différentes étapes de traitements, dans le but de fournir aux interprétateurs, des sections sismiques de bonne qualités reflétant au mieux la configuration du sous-sol.

1.3.2.1.1.3. Définition des paramètres d'acquisition en 3D [08], [10]

Dans la plupart des compagnies pétrolières, les acquisitions sismiques marines 3D ont augmenté de façon exponentielle entre 1990 et 1996 pour couvrir la majorité de leurs champs. Aujourd'hui, la sismique 3D est de plus en plus utilisée aussi en terrestre.

1- lignes de récepteurs

Ligne de récepteur est une ligne où les récepteurs sont situés à une distance.

Les lignes de récepteur sont conservées aussi droite que possible dans l'étude 3D.

Notations :

RY : Espacements entre deux lignes de réception

RX : Espacements entre deux points de réception

Rd : la densité d'un récepteur est le nombre de récepteurs par unité de surface, en général Kilomètre carré

2- lignes de sources (tirs)

Ligne de source est une ligne où des points de sources sont situés à une distance régulière. Dans l'étude 3D les lignes de sources peuvent être orthogonales ou parallèlement aux lignes de récepteurs.

Notations : **SX** : Espacements entre deux lignes de tirs

SY : Espacements entre deux points de tirs

3- Box

Dans un plan orthogonal le "**Box**" (Figure 1.7) correspond à la zone couverte par deux lignes de récepteur consécutive (espacées de Ry) et deux lignes de source consécutive (espacées Sx).

La surface de "**Box**" est alors : **Sb= Ry * Sx**

4- Template et salvo:

Tous les récepteurs actifs correspondant à un point de tir donné correspondent à un "template".

Ces récepteurs sont situés sur plusieurs lignes parallèles.

Salvo le nombre de points de tir entre deux lignes de réceptions.

- "**Maximum offset**" (**Xmax**) : la distance de la demi-diagonale du "template".

- " **Maximum Minimum offset** " (X_{min}) : la longueur de la diagonale de box formée par deux lignes consécutive de récepteurs et sources

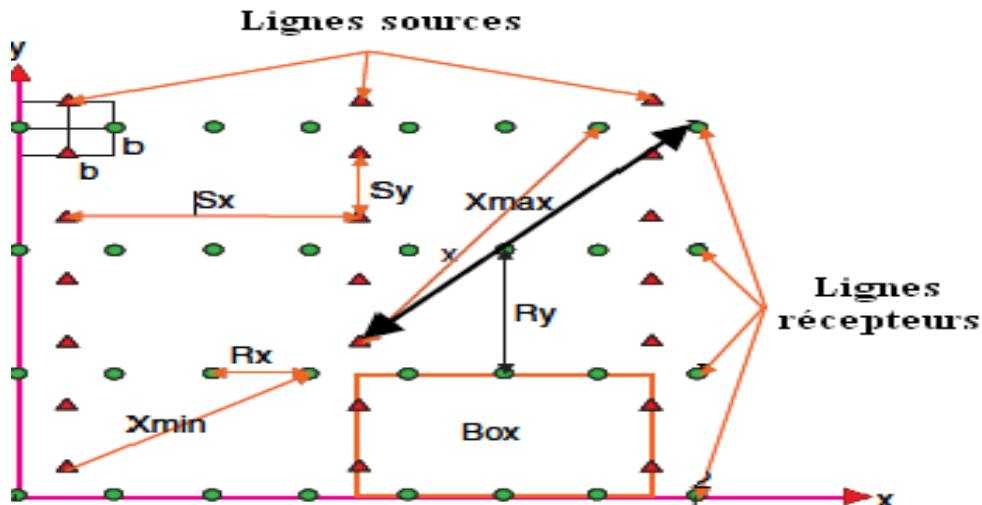


Fig.1.7. Lignes récepteurs et de sources

5- Directions : Deux types d'accès sont à considérer (Figure 1.8)

- " **In line direction** " : ce qui est parallèle aux lignes de récepteur. Prélèvement d'échantillonnage dans cette direction est globalement satisfaisant.
- " **Cross line direction** " : ce qui est orthogonal aux lignes de récepteur. Prélèvement d'échantillonnage dans cette direction est généralement faible et a été étudiée attentivement.

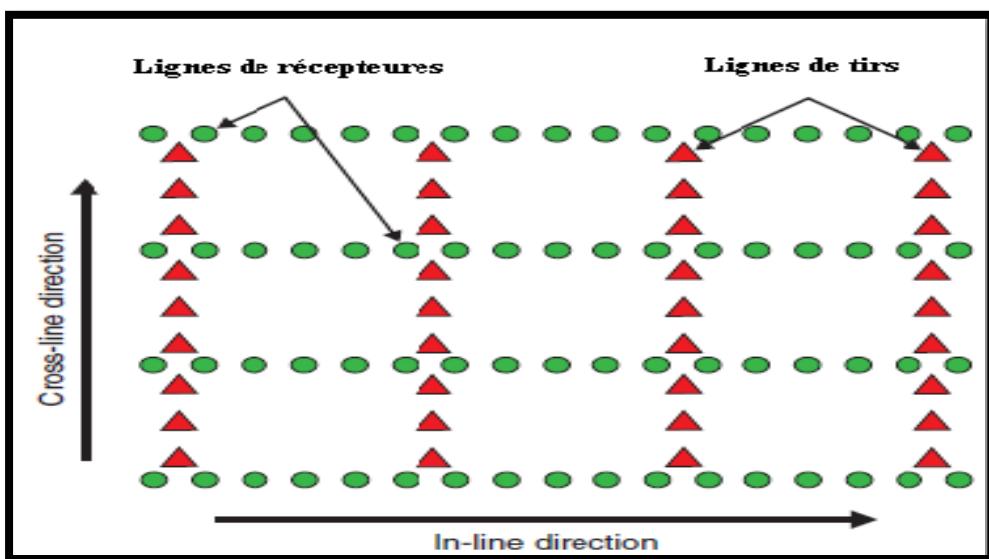


Fig.1.8. Les deux types d'accès (directions)

6- Midpoint (le point médian) : Est un point situé exactement au milieu de la distance

source - récepteur. Dans la pratique, ils forment rarement une grille régulière.

- " **Common midpoint** " (CMP) : Est un point situé au milieu de différents accouplés source-récepteur de réflexion qui correspondent au même sous-point.
- " **CMP bin** " : Est une surface carré ou rectangulaire, qui contient tous les milieux qui correspondent à le même CMP.
- **Taille de " Bin "** : taille de bin correspond à la longueur et à la largeur du bin. La plus petite dimension des bins est égale ($S_y / 2 * R_x / 2$).

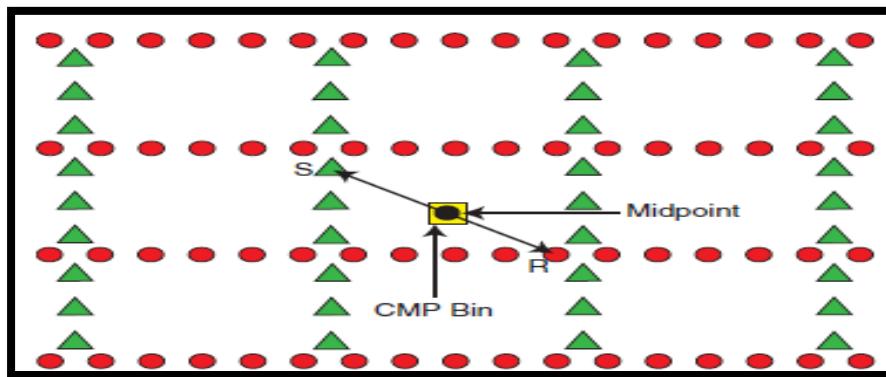


Fig.1.9. Midpoint

- 7- La couverture" fold "** : Est le nombre de points médians " Midpoint " qui tombent dans le même carrée " bin " et qui seront empilés.

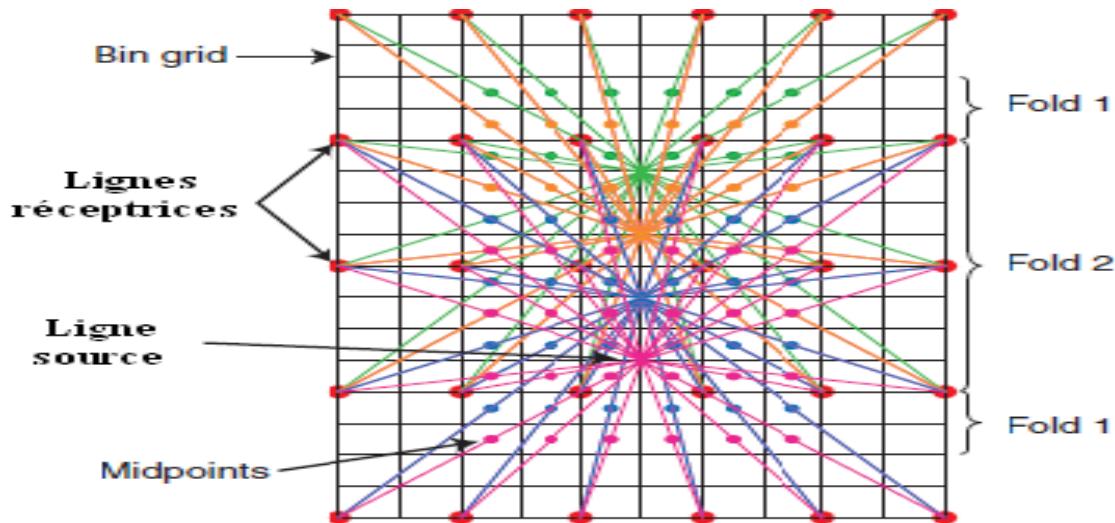


Fig.1.10. la couverture (fold)

- 8- Move-uns** : Deux types de mouvements " Move-ups " peut être considéré pour l'étude 3D
 - " **In-line move-up** " : se produit lorsque le " template " se déplace vers le haut le long de l'étude de sa position initiale après l'exécution de " salvo " de tir.

- " **Cross-line move-up** ": se déplacer lorsque le " template " atteint le bord de surface d'étude et se déplace vers le haut à travers l'étude pour commencer une nouvelle ligne à déplacer.

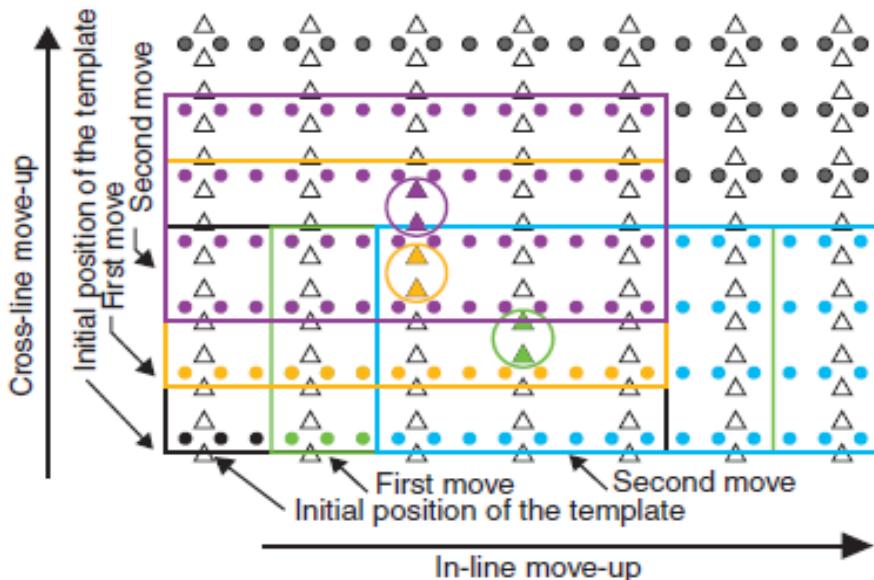


Fig.1.11. les mouvements " Move-ups "

9- offset : Le prélèvement dans le Compte tenu de la configuration du " template " 3D, différentes " Offsets " peuvent être définis (Figure 1.12) :

- " **In line offset** ": est la distance représentant la moitié de la longueur du " template " dans "in-linedirection".
- " **Cross-ligne offset** ": est la distance représentant la moitié de la longueur du "template" dans "cross-linedirection".

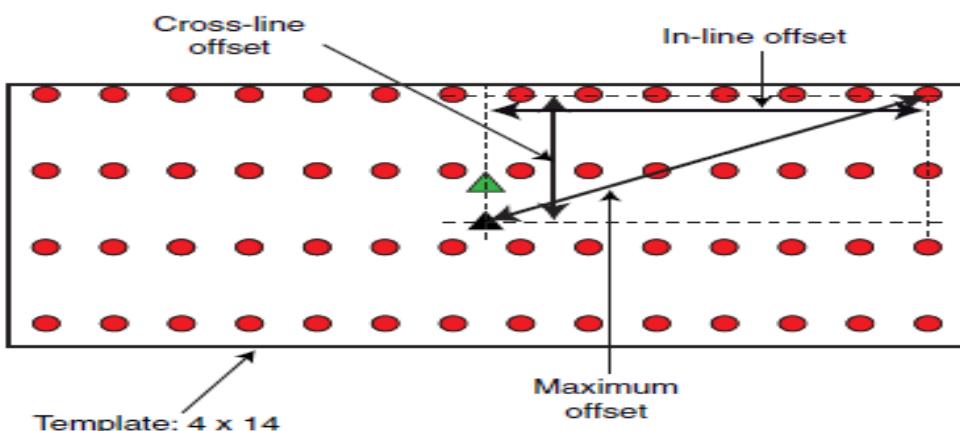


Fig.1.12 Différentes type de Offset

1.3.2.1.2. La problématique :

Sonatrach possède une expérience de plus de 30 ans dans le domaine de la recherche des hydrocarbures par sismique réflexion, une des méthodes les plus utilisées en Algérie et dans le monde. Des ingénieurs et techniciens hautement qualifiés ont permis à l'entreprise d'atteindre une maîtrise parfaite dans l'utilisation d'instruments de dernière technologie tels que les laboratoires télémétriques à grand nombre de canaux, les vibrateurs et les récepteurs G.P.S pour la sismique 2D et 3D. L'entreprise est dotée d'équipements de dernière technologie pour faire face à toute demande de ses clients dans le domaine de l'acquisition sismique.

1.3.2.1.2.1. Etude de l'existant :

Sonatarch compte 13 équipes sismiques, ces équipes représentent le fer de lance de l'activité de l'entreprise, chaque équipe est organisée en échelons ou sections solidaires et interdépendants : topographie, vibrateurs, pose et ramassage du dispositif d'enregistrement, contrôle qualité... Dès que l'équipe sismique se voie confier un nouveau projet, elle analyse son cahier de charge ou figurent tous les paramètres de l'étude en question dont les coordonnées des points limites du périmètre à étudier, grâce au logiciel Mesa dans lequel on introduit les données du client (distance entre deux points vibrés, entre deux récepteurs ...), l'équipe obtient toutes les coordonnées de tous les points à faire vibrer à l'intérieur du permis, leurs nombre, le nombre de traces ... et là commence l'activité des autres échelons.

Echelon topographie

Le topographe est le premier à aller sur le terrain, il implante le profil en temps réel par positionnement par satellite, son rôle est de matérialiser sur le terrain les profils ou lignes sismiques par l'implantation de piquets (sources et réception), le nombre de piquet implantés peut atteindre 800 en une seule journée ; ce sont des kilomètres à pied qui sont journalement parcourus.

Echelon layonnage

Les Bulldozers passent ensuite pour ouvrir les pistes longeant les lignes des piquets sources uniquement, ceci afin de faciliter et de permettre, principalement, le passage des vibrateurs.

Remarque 1.1 : dans les zones du permis où la surface est plate et accessible aux vibrateurs l'opération Bulldozers n'est pas nécessaire.

Echelon vibrateurs

Les vibrateurs constituent des sources d'énergie, ils évoluent le long des profils (piquets sources) et selon les modalités prédefinies par le client. L'énergie produite par les vibrateurs pénètre, sous forme d'onde sismique, dans le sous sol ; ces ondes reviennent réfléchies par les interfaces séparant les couches géologique, elles sont réceptionnées par le dispositif d'enregistrement (les récepteurs appelés aussi géophones positionné au niveau des piquets de réceptions), ces récepteurs sont reliés à un laboratoire d'enregistrement télémétrique grâce à des câbles. Traduites en énergie électrique, ces ondes sont enregistrées sur bandes magnétique.

Echelon pose et ramasse

Il s'agit de personnel qualifié qui pose et ramasse le dispositif d'enregistrement (les grappes de géophones) à mesure de l'avancement de l'étude, vue que la disponibilité de ces derniers au niveau d'un chantier sismique est limitée.

Section contrôle qualité

Les données topographiques et les bandes magnétiques enregistrés sur le terrain, sont ramenées le soir au camp où elles subissent un contrôle de la qualité sur les stations de traitements. Quand le produit est jugé de qualité, il est transféré au client

1.3.2.1.2.2. Problème posé :

L'intérêt accordé jusque là à l'optimisation des seules paramètres géophysiques en acquisition sismique (3D), dans un contexte économique caractérisé par une concurrence féroce et des marges réduites, s'avère insuffisant pour la performance de l'entreprise.

La recherche en permanence par l'entreprise à réaliser des économies de coûts, un avantage concurrentiel considérable, nous incite aussi à l'optimisation de l'aspect opérationnel (exécution des opérations logistiques dans le dispositif sismique 3D).

Jusqu'à présent Sonatarch exécute fidèlement les consignes et applique les valeurs des paramètres de l'étude exigés dans le cahier de charge du client, sans se soucier de l'aspect coût. Sa principale finalité était de maximiser la production (le nombre de points vibrés par jour) quelque soit le coût de l'opération, mais pour faire face à la concurrence de plus en plus féroce, le temps est venu d'essayer de minimiser ces coûts d'où est venu l'idée de nous proposer ce sujet

qui a pour titre « Optimisation des déplacements des camions vibrateurs au niveau d'un chantier sismique». A cet effet, l'étude portera sur l'optimisation du dispositif d'acquisition comprenant à la fois le tracé des lignes de passage des camions vibrateurs et celles de la pose des récepteurs, dans le but d'avoir de meilleurs résultats lors de l'application en termes de coûts et de temps de réalisation des projets.

1.3.2.1.3. Objectifs fixés :

L'étude aura essentiellement pour objectif ce qui suit :

- Optimisation des déplacements des camions vibrateurs au niveau d'un chantier sismique afin de minimiser les coûts et le temps de réalisation d'un projet d'exploration du pétrole et du gaz.

1.3.2.2. Problème 2.2 : Elaboration d'un schéma optimal de la fermeture des puits de pétrole et de gaz de Sonatrach

Les Activités de Recherche et d'exploitation peuvent avoir des impacts négatifs sur l'environnement si les outils nécessaires ne sont pas réunis pour répondre aux exigences réglementaires et aux engagements Santé Sécurité Environnement (HSE) du Groupe Sonatrach. Ainsi, la fonction (HSE) au sein de notre entreprise doit veiller à l'application de la réglementation en vigueur et se prémunir contre tout risque.

En effet, l'un des défis majeurs auxquels doit faire face l'entreprise aujourd'hui est la mise en conformité des puits abandonnés.

1.3.2.2.1. Définitions et généralités

1.3.2.2.1.1. Abandon définitif de puits [09]

Le but principal de tout abandon de puits consiste à isoler de manière permanente toutes les formations souterraines traversées par le puits. Bien que l'obturation des réservoirs épuisés soit d'un intérêt certain en termes de procédures, les opérations idéales d'abandon s'effectuent en isolant simultanément les réservoirs productifs et les autres formations liquides.

Une isolation complète permet d'éviter la migration de gaz de pétrole ou d'eau vers la surface. L'opération de fermeture définitive d'un puits est indispensable puisque il est inutilisable pour différentes raisons

A) Raison de sécurité

- ✓ Venue de gaz et d'eau ; Eruption interne ;
- ✓ Ou pour d'autres raisons sécuritaires pouvant surgir

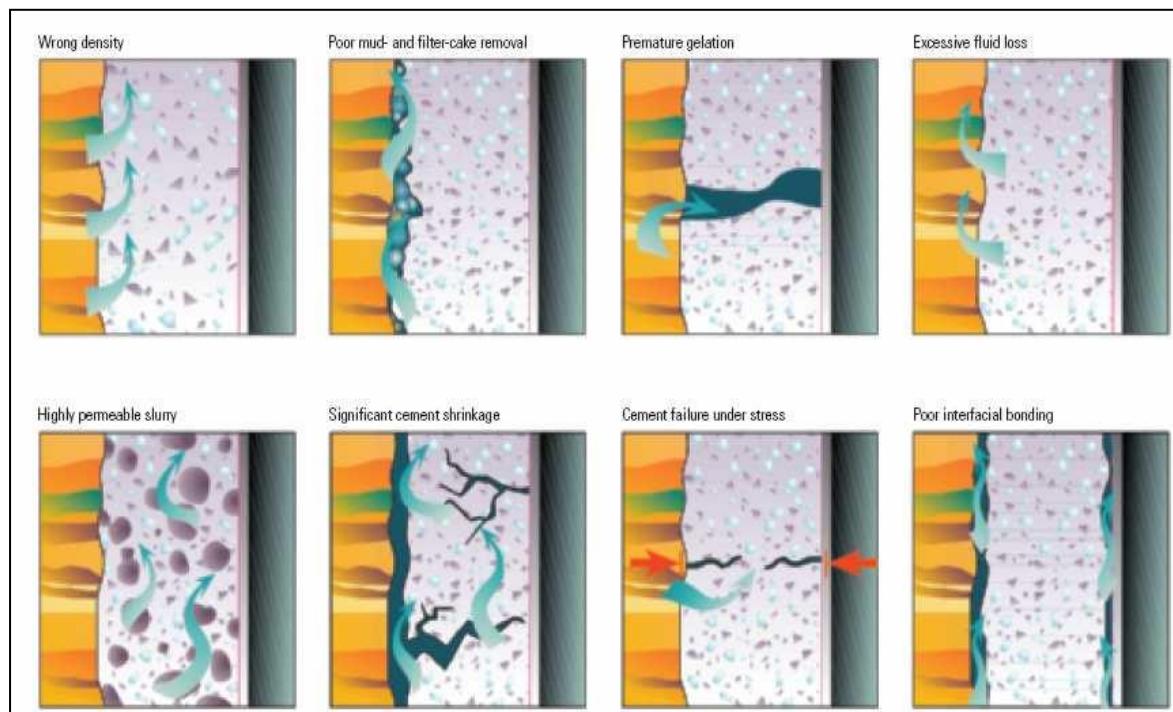


Fig.1.13 Schéma explicatif des raisons d'abandon

B) Raison technique

La réglementation cite quatre (04) catégories de puits pouvant être abandonnés pour des raisons techniques :

1- La première catégorie comprend :

- ✓ Les puits d'exploration ou d'extension qui ont atteint leurs objectifs mais qui se sont avérés secs ou producteurs d'eau ;
- ✓ Les puits qui n'ont pas atteint leurs objectifs mais dont le forage est arrêté pour absence d'intérêt géologique ;
- ✓ Les puits qui ont atteint la profondeur prévue sans avoir ouvert l'ensemble du niveau à cause du programme de forage adopté.

2- La deuxième catégorie comprend les puits ayant rencontré de mauvaises caractéristiques de réservoir.

3- La troisième catégorie comprend les puits abandonnés pour des raisons imputables à la mauvaise qualité des forages.

4-La quatrième catégorie comprend les puits qui doivent être abandonnés pour les raisons suivantes :

- ✓ Invasion par l'eau du réservoir ou par l'eau injectée ;
- ✓ État de déplétion très avancée pour lequel le débit est inférieur au seuil de rentabilité
- ✓ Absence d'intérêt d'utilisation ultérieure pour l'injection ou l'observation pour des raisons techniques et géologiques.

1.3.2.2.1.2. Pourquoi doit-on fermer les puits abandonnés

Les procédures d'abandon de puits et de champ font aujourd'hui l'objet d'un intérêt majeur dans l'Amont pétrolier pour plusieurs raisons ; mais la raison majeure est la suivante :

- ♦ La protection de l'environnement à long terme, surtout lorsque les réglementations deviennent plus strictes et plus complexes (la loi 05-07 Amendée).

Pour envisager une opération d'abandon, l'exploitant doit se conformer à plusieurs procédures.

1.3.2.2.1.3. L'appareil Rig « Plate-forme de forage » [09]

La figure (1.12) montre la description de l'appareil de forage utilisé pour la fermeture d'un puits.

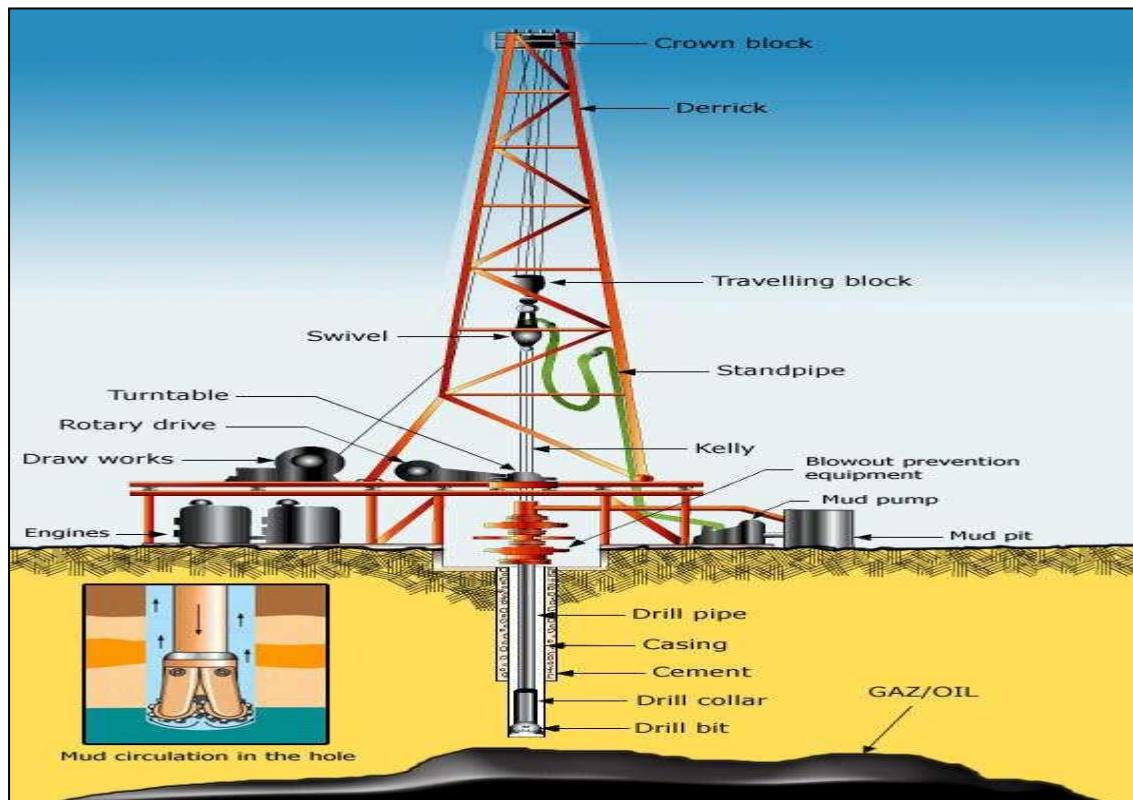


Fig.1.14 Rig « Plate-forme de forage »

1.3.2.2. Problématique

Position du problème

Le problème posé par l'activité Amont de SONATRACH consiste en la fermeture des puits abandonnés et ce pour les raisons suivantes :

- ✓ La loi sur l'environnement à long terme (loi 05-07 Amendée) qui est devenu plus stricte et qui oblige la conformité de la fermeture d'un puits ;
- ✓ De nombreuses implantations en Algérie sont concernées, car de nombreux champs atteignent aujourd'hui leurs limites économiques et de production ;
- ✓ La capture et le stockage du dioxyde de carbone sous la surface peuvent être l'une des solutions pour réduire la concentration de dioxyde de carbone dans l'atmosphère. Le secteur du pétrole et du gaz peut jouer un rôle important dans ce domaine comme le CO₂ peut être stocké dans des réservoirs épuisés, des aquifères salins profonds ou des veines d'un gisement de charbon.

La problématique proposée par Direction Etudes Economiques et Synthèses (DES) peut traiter plusieurs aspects, à savoir :

- ❖ Optimisation du planning d'utilisation d'appareils « Rig »;
- ❖ La sécurité, le respect des normes ainsi que les aspects économiques.

1.3.2.2.3. Objectifs

Les objectifs principaux de notre étude sont :

- ❖ La fermeture de tout les puits abandonnés ;
- ❖ Minimiser le coût d'affectation des machines « Rig » aux puits abandonnés qui revient à minimiser les distances de parcours.

1.4. Conclusion :

Ce chapitre a été consacré à la présentation des problématiques et aux principaux objectifs assignés à nos études. La modélisation des problèmes qui fera l'objet du chapitre suivant.

Chapitre 2

Modélisation mathématique

2.1. Introduction

Chaque problème qu'un être humain rencontre ne peut être suffisamment simple pour être compris sans avoir recours à un concept, idée ou notion. L'idée consiste à remplacer le phénomène qu'on étudie par un modèle de structure semblable, mais un peu moins complexe. Par conséquent les modèles sont une nécessité fondamentale de la démarche scientifique. Cependant dans la pratique trouvée un modèle mathématique qui convienne à un problème donné n'est pas une tâche des plus faciles. L'idéal est de pouvoir prendre en charge les contraintes de problème réel et pouvoir les exprimer sous forme mathématique d'une manière adéquate et simple.

2.2. Modélisation mathématique du problème 1

Avant de dresser une modélisation du système, nous allons présenter les hypothèses retenues sur le système étudié, sans que cela ne remette en question la représentativité du modèle.

2.2.1. Les hypothèses

Le but de toute modélisation est de représenter de la manière la plus fidèle possible la réalité ; pour cela la disponibilité de données fiables et suffisantes est indispensable. La complexité du système étudié nécessite la formulation d'un certain nombre d'hypothèses permettant de mieux cerner les aspects importants et influents du système sans altérer son comportement réel ; c'est ainsi que nous avons défini l'ensemble des hypothèses suivantes :

➤ Au niveau des postes de chargement

- dans notre étude on considère que les ports d'Arzew et de Betioua comme ports d'exportation, car les volumes exportés occasionnellement à partir du port de Skikda sont insignifiants ;
- ces deux ports sont considérés comme étant un seul port de chargement, vu leur proximité ;
- Le port considéré dans notre étude, comprend donc cinq postes de chargements (S1, S2, S3, D1, M6) ;
- On suppose que le produit sur le port de chargement est tout le temps disponible (en réalité l'épuisement du stock arrive très rarement) ;

- On suppose que le temps de chargement est d'une journée pour toutes les classes de navire (car en réalité il est à-peu-près d'une journée pour toutes les classes) ;
- le travail se fait dans des conditions normales (pas de grève des employés, ou tout autre organisme dont dépend le bon fonctionnement de la procédure de chargement) ;
- Les conditions climatiques sont favorables au chargement des navires ;
- Dès son débarquement sur le quai, le chargement du navire commence ;
- Le matériel de chargement est dans un état de fonctionnement normal.

➤ **Au niveau des ports de déchargement**

- Chaque port de déchargement possède une demande mensuelle ;
- Le client est tenu responsable du déchargement dans les délais des quantités transportées, du moment que ces quantités sont demandées ;
- Le client est tenu responsable du stockage de la quantité déchargée, du moment qu'elle se situe dans la marge de flexibilité mensuelle.

➤ **Au niveau de la flotte**

La flotte qu'on est chargé de déterminer, sera constituée des navires de Sonatrach, plus les navires affrétés. Pour que tous les navires de cette flotte auront le même statut, on considère que les navires de Sonatrach, sont des navires affrétés avec un coût nul.

➤ **Sur la période d'étude**

La période d'étude sera soit une saison, soit une année ; de toutes les façons elle sera partitionnée en mois, qui seront partitionnés à leurs tours en jours, ces jours seront numéroté d'une façon continue, pour faciliter l'écriture des contraintes (si la période d'étude sera une année, les jours seront numérotés de 1 à 365).

➤ **Aux niveaux des demandes**

On considère les deux gaz GPL transportés comme étant un seul gaz, car chaque client demande une quantité mensuelle de chaque gaz, donc on doit tout simplement assurer le transport de la somme des deux quantités en tonnes, puis charger les navires avec les portions correspondante

2.2.2. Les contraintes

La minimisation des coûts de transport des exportations est régie par plusieurs contraintes dont les principales sont :

- Contraintes portuaires ;
- Contraintes de la demande ;
- Contraintes de rotation des navires.

Contraintes Portuaires

1. Aux ports de chargement

Un poste de chargement ne peut recevoir qu'un seul navire le même jour, car le temps de chargement de tous les navires est à peu près de 24h.

2. Aux ports de déchargement

Chaque navire présente des caractéristiques techniques spécifiques qui doivent être conformes aux restrictions portuaires de chaque port de livraison, qui sont :

- ✓ La capacité de chargement ;
- ✓ Le tirant d'eau ;
- ✓ La longueur.

Contraintes de demandes

Chaque client possède une demande mensuelle .Cette demande doit être livré avec une marge de flexibilité.

Contraintes de rotations des navires

- un navire ne peut pas servir plus d'un client en même temps ;
- un navire ne peut pas effectuer un chargement pour un client, alors qu'il n'est pas rentré d'une rotation précédente éventuelle.
- Tous les navires doivent démarrer leurs rotations éventuelles, avant la fin de la période d'étude.

2.2.3. Modèle mathématique

2.2.3.1. Nomenclature

➤ Indice

- i : Indice navire; j : Indice jour ;
- k : Indice client (port de déchargement); l : Indice quai; m : Indice mois.

➤ Données

- NN : Nombre de navires (ce nombre comprend les navires de Sonatrach plus les navires susceptibles d'être affrété, en d'autres termes, c'est la liste de tous les navires candidats pour constituer notre flotte) ;
- NJ : nombre de jours sur toute la période d'étude ;
- NC : Nombre de clients (destinations) ;
- NQ : Nombre de quais ;
- NM : Nombre de mois sur la durée d'étude ;
- D_m : Date du début du mois m ;
- F_m : Date de la fin du mois m ;
- Q_i : Capacité du navire i ;
- DM_{km} : Demande du client (port) le mois m ;
- km : La marge de flexibilité de la demande du client k le mois m ;
- DR_{ik} : durées de rotation du navire i entre le port de chargement et le client (destination) k ;
- $a_{ik} = \begin{cases} 1 & \text{Si le client } k \text{ peut recevoir le navire } i \\ 0 & \text{Sinon} \end{cases}$
- CA_i : Le coût d'affrètement du navire i sur la période d'étude ;
- CE_{ik} : Coût de d'exploitation entre le port de chargement et le client k du navire i .

2.2.3.2. Définition des variables

$$X_{ijkl} = \begin{cases} 1 & \text{Si le navire } i \text{ effectue un chargement à jour } j \text{ pour le client } k \text{ dans la quai } l \\ 0 & \text{Sinon} \end{cases}$$

$$i = \overline{1; NN} \quad j = \overline{1; NJ} \quad k = \overline{1; NC} \quad l = \overline{1; NQ}$$

2.2.3.3. Définition de la fonction objective

Notre objectif est de minimiser le coût de transport, qui comprend le coût d'exploitation et le coût d'affrètement des navires. La fonction "objectif" sera donc :

$$\text{Min } Z = \sum_i^{NN} \sum_j^{NJ} \sum_k^{NC} \sum_l^{NQ} CE_{ik} X_{ijkl} + \sum_i^{NN} CA_i \left[\frac{\sum_j^{NJ} \sum_k^{NC} \sum_l^{NQ} X_{ijkl}}{NJ \times NC \times NQ} \right]$$

2.2.3.4. Définition des contraintes

1- Contrainte de demande

Ces contraintes garantissent que la demande mensuelle du client 1 sera satisfaite avec une marge flexibilité.

$$DM_{km} - km \leq \sum_i^{NN} \sum_l^{NQ} \sum_{j=Dm}^{Fm} Q_i X_{ijkl} \leq DM_{km} + km$$

$$m = \overline{1; NM} \quad k = \overline{1; NC}$$

2- Contrainte de portuaire

- **Au port de chargement**

Ces contraintes expriment qu'un quai ne peut accueillir qu'un navire dans le même jour.

$$\sum_{i=1}^{NN} \sum_{k=1}^{NC} X_{ijkl} \leq 1 \quad j = \overline{1; NJ} \quad l = \overline{1; NQ}$$

- **Aux ports de déchargements**

Ce type de contraintes exprime qu'un client ne peut recevoir que certains types de navires.

$$X_{ijkl} \leq a_{ik} \quad i = \overline{1;NN} \quad j = \overline{1;NJ} \quad k = \overline{1;NC} \quad l = \overline{1;NQ}$$

3- Contraintes de rotations

Ce type de contraintes exprime à la fois :

- un navire ne peut pas servir plus d'un client en même temps ;
- un navire ne peut pas effectuer un chargement après la fin de l'année.

$$\min\{J+DR_{ik}, NJ\} \sum_{s=J}^{NC} \sum_{t=1}^{NQ} X_{ijst} \leq 1 \quad i = \overline{1;NN} \quad j = \overline{1;NJ} \quad k = \overline{1;NC} \quad l = \overline{1;NQ}$$

2.2.3.5. Résumé du modèle

$$\left\{ \begin{array}{l} \text{Min } Z = \sum_i^{\overline{1;NN}} \sum_j^{\overline{1;NJ}} \sum_k^{\overline{1;NC}} \sum_l^{\overline{1;NQ}} CE_{ik} X_{ijkl} + \sum_i^{\overline{1;NN}} CA_i \left[\frac{\sum_j^{\overline{1;NJ}} \sum_k^{\overline{1;NC}} \sum_l^{\overline{1;NQ}} X_{ijkl}}{NJ \times NC \times NQ} \right] \\ DM_{km} - km \leq \sum_i^{\overline{1;NN}} \sum_l^{\overline{1;NQ}} \sum_{j=D_m}^{F_m} Q_l X_{ijkl} \leq DM_{km} + km \\ \sum_{i=1}^{\overline{1;NN}} \sum_{k=1}^{\overline{1;NC}} X_{ijkl} \leq 1 \quad j = \overline{1;NJ} \quad l = \overline{1;NQ} \\ X_{ijkl} \leq a_{ik} \quad i = \overline{1;NN} \quad j = \overline{1;NJ} \quad k = \overline{1;NC} \quad l = \overline{1;NQ} \\ \sum_{s=J}^{\min\{J+DR_{ik}, NJ\}} \sum_{t=1}^{NQ} X_{ijst} \leq 1 \quad i = \overline{1;NN} \quad j = \overline{1;NJ} \quad k = \overline{1;NC} \quad l = \overline{1;NQ} \\ X_{ijkl} \in \{0;1\} \quad i = \overline{1;NN} \quad j = \overline{1;NJ} \quad k = \overline{1;NC} \quad l = \overline{1;NQ} \end{array} \right.$$

2.2.4. Identification du modèle

La modélisation du problème a permis d'obtenir un programme linéaire en variable

bivalentes. Si ce problème pourrait se formuler en un nombre restreint de variables, on aurait pu peut-être utiliser une méthode de séparation et évaluation par exemple, mais vu le nombre important de variables, et compte tenu que ces variables soient bivalentes, on était contraint à se diriger vers les métaheuristiques.

2.3. Modélisation mathématique du problème 2

2.3.1. Modélisation mathématique du problème 2.1

2.3.1.1. Modèle mathématique

2.3.1.1.1. Définitions des données

- F : La couverture exigée par le client, I : Facteur d'inflation,
- O_{IN} : "In-line offset" exigé, O_{CR} : "Cross-line offset" exigé,
- O_{MAX} : "Maximum offset" exigé, O_{MIN} : "Minimum offset" exigé,
- C_{MAX} : Le nombre maximum de canaux de réception disponibles,
- S_{MAX} : Le nombre maximum de points vibrés par jour,
- R_{MAX} : Le nombre maximum de canaux de réception déplacés par jour,
- C_{SSV} : Le coût du levé topographique par point de tir,
- C_{RSV} : Le coût du levé topographique par point de récepteur,
- C_{SCC} : Le coût de nettoyage par le bulldozer par km pour les lignes de tir,
- C_{CR} : Coût quotidien moyen de l'équipage,
- C_{EQ} : Coût de l'équipement / récepteur canal par jour,
- C_{PCT} : Le coût de traitement par trace.

2.3.1.1.2. Définitions des variables

- X_1 La distance entre deux lignes de tirs consécutives.
- X_2 La distance entre deux lignes de réception consécutives.

- X_3 La distance entre deux points de tirs consécutifs sur une même ligne.
- X_4 La distance entre deux points de réception consécutives sur une même ligne.
- X_5 Nombre de canaux de récepteurs actifs dans le "template"
- X_6 Nombre de lignes de récepteurs actifs dans le "template".
- X_7 Nombre de point vibré (VP) par kilomètre carré.
- X_8 Nombre de point vibré (VP) par jour.
- X_9 Le nombre de canaux déplacés par jour pour être posés à l'avant du profil.
- X_{10} Nombre de points de tirs entre chaque ligne réceptrice.
- X_{11} Nombre de récepteurs entre chaque ligne tirs.
- X_{12} Nombre de canaux de récepteur actif par ligne réceptrice active.
- X_{13} La moitié du nombre de lignes de récepteur.

Où X_1, X_2, X_3, X_4 sont des réels positifs. Alors que $X_5, X_6, X_7, X_8, X_9, X_{10}, X_{11}, X_{12}, X_{13}$ représentent des entiers positifs.

2.3.1.1.3. Les contraintes

A- Contraintes géophysiques

Il s'agit d'exigences géophysiques. Elles concernent l'ordre de couverture (F), "maximum offset" (OMAX) et " maximum minimum offset" (OMIN)

- 1. Contrainte relative à la couverture " Fold ":** afin de satisfaire son client Sonatrach doit réaliser une couverture supérieure ou égale à celle exigée par ce dernier :

$$X_5 \times X_7 \times \frac{X_3}{2} \times \frac{X_4}{2} \geq F$$

2. "In line offset" doit rester supérieur à une valeur donnée O_{IN}

$$\left(\frac{X_5}{2X_6} \times X_4 \right) - \frac{X_4}{2} \geq O_{IN}$$

3. "Cross line offset" doit rester supérieur à une valeur donnée O_{CR}

$$\left(\frac{X_6}{2} \times X_2 \right) - \frac{X_3}{2} \geq O_{CR}$$

4. "Maximum offset" dans toute étude doit être supérieur à une valeur donnée O_{MAX}

$$\left(\left(\frac{X_5}{2X_6} \times X_4 \right) - \frac{X_4}{2} \right)^2 + \left(\left(\frac{X_6}{2} \times X_2 \right) - \frac{X_3}{2} \right)^2 \geq O_{MAX}^2$$

5. "Maximum minimum offset" dans toute étude doit être inférieur à une valeur

Donnée O_{MIN}

$$X_1^2 + X_2^2 \leq O_{MIN}^2$$

B- Contraintes opérationnelles :

Comme le nombre de points vibrer par jour qui dépend de la nature géologique du terrain et la disponibilité des canaux de réceptions.

6. Contrainte relative au nombre de point vibrer par jour : ce dernier ne peut pas dépasser une valeur fixée comme limite (SMAX).

$$X_8 \leq S_{MAX}$$

7. Le nombre de canaux de réception déplacés par jour ne peut pas dépasser le nombre de Canaux de réception qui peut être déplacé en une journée (R_{MAX}).

$$X_9 \leq R_{MAX}$$

8. L'équivalence entre le nombre de tirs par jour et le nombre de canaux de réception

déplacé par jour. Le but est d'avoir au début de chaque journée de travail assez de récepteurs en position pour pouvoir lancer l'acquisition vue que la disponibilité des ces derniers est limité.

$$X_8 \times X_1 \times X_3 = X_9 \times X_4 \times X_2$$

9. Contrainte relative à la disponibilité du nombre de canaux de réception (C_{MAX})

$$X_5 \times (1 + I) \leq C_{MAX}$$

10. Contrainte relative à la densité des points de tirs.

$$X_7 \times X_1 \times X_3 = 1$$

11. Contrainte relative à la distance entre deux lignes réceptrices.

$$X_2 - (X_{10} \times X_3) = 0$$

12. Contrainte relative à la distance entre deux lignes de tirs.

$$X_1 - (X_{11} \times X_4) = 0$$

13. Contrainte relative à la cohérence entre des canaux de récepteurs actif par les lignes de récepteur actif.

$$X_5 - (X_{12} \times X_6) = 0$$

14. Condition de la scission-propagation de conception.

$$X_6 - (2 \times X_{13}) = 0$$

2.3.1.1.4. La fonction de l'objectif :

La minimisation des déplacements des camions vibrateurs grâce à l'optimisation des paramètres sismique engendre une minimisation des coûts

Les coûts peuvent être décomposés et formulés comme suit :

[(Coût de levé topographique/Km2) + (Coût du layonage/Km2) + (Coût d'émission réception/Km2) + (Coût de traitement/Km2)]

- **Coût de levé topographique :** est le coût produit par l'équipe topographie lors de l'implantation des points d'émission et de réception .Il est formulé comme suit :

$$C_{SSV} \times X_7 + C_{RSV} \times \left(\frac{1}{X_2 \times X_4} \right)$$

- **Coût du layonage :** est un coût généré par l'équipe bulldozers ayant pour mission de faciliter l'accès aux camions vibrateurs en éliminant les obstacles. Il est formulé comme suit :

$$C_{SCC} \times \left(\frac{1}{X_1} \right)$$

- **Coût d'émission réception :** est le coût généré par les camions vibrateurs. Il est formulé comme suit :

$$\left((C_{CR} + C_{EQ} \times (X_5 \times (1+I))) \times \left(\frac{X_7}{X_8} \right) \right)$$

- **Coût de traitement :** est le coût généré par l'équipe contrôle de qualité de l'image. Il est formulé comme suit :

$$C_{PCT} \times X_7 \times X_5$$

Il s'agira donc de minimiser la fonction suivante :

$$\begin{aligned} Min(Z) = & C_{SSV} \times X_7 + C_{RSV} \times \left(\frac{1}{X_2 \times X_4} \right) + C_{SCC} \times \left(\frac{1}{X_1} \right) + \\ & \left((C_{CR} + C_{EQ} \times (X_5 \times (1+I))) \times \left(\frac{X_7}{X_8} \right) \right) + C_{PCT} \times X_7 \times X_5 \end{aligned}$$

2.3.1.1.5. Résumé du modèle :

Nous obtenons le programme mathématique suivant:

$$\left\{
 \begin{array}{l}
 \text{Min}(Z) = C_{SSV} \times X_7 + C_{RSV} \times \left(\frac{1}{X_2 \times X_4} \right) + C_{SCC} \times \left(\frac{1}{X_1} \right) + \\
 \quad \left((C_{CR} + C_{EQ} \times (X_5 \times (1+I))) \times \left(\frac{X_7}{X_8} \right) \right) + C_{PCT} \times X_7 \times X_5 \\
 X_5 \times X_7 \times \frac{X_3}{2} \times \frac{X_4}{2} \geq F \\
 \left(\frac{X_5}{2X_6} \times X_4 \right) - \frac{X_4}{2} \geq O_{IN} \\
 \left(\frac{X_6}{2} \times X_2 \right) - \frac{X_3}{2} \geq O_{CR} \\
 \left(\left(\frac{X_5}{2X_6} \times X_4 \right) - \frac{X_4}{2} \right)^2 + \left(\left(\frac{X_6}{2} \times X_2 \right) - \frac{X_3}{2} \right)^2 \geq O_{MAX}^2 \\
 X_1^2 + X_2^2 \leq O_{MIN}^2 \\
 X_8 \leq S_{MAX} \\
 X_9 \leq R_{MAX} \\
 X_8 \times X_1 \times X_3 = X_9 \times X_4 \times X_2 \\
 X_5 \times (1+I) \leq C_{MAX} \\
 X_7 \times X_1 \times X_3 = 1 \\
 X_2 - (X_{10} \times X_3) = 0 \\
 X_1 - (X_{11} \times X_4) = 0 \\
 X_5 - (X_{12} \times X_6) = 0 \\
 X_6 - (2 \times X_{13}) = 0 \\
 X_i \geq 0 \quad i = 1, \dots, 4 \\
 X_i \geq 0, \text{et } i \in \mathbb{N} \quad i = 5, \dots, 13
 \end{array}
 \right.$$

2.3.1.2. Identification du modèle :

Le problème ainsi modélisé a donné lieu à un programme non-linéaire, à variables mixtes.

La non linéarité d'une partie des contraintes du problème est due au respect des paramètres géophysiques.

2.3.2. Modélisation mathématique du problème 2.2

2.3.2.1. Définitions des variables et des données :

Application au champ du hassi berkine :

Dans cette étape les puits à fermer nécessitent l'intervention d'un appareil « Rig » d'où l'assimilation de nos donnés au problème du VRP qui se traduit de la manière suivante :

- Un client dans le problème du VRP représente un puits dans notre problème ;
- Un véhicule dans le VRP est assimilé à un appareil dans notre problème.

Ici on dispose d'un seul appareil, le tableau suivant indique les distances séparant les différents puits à fermer dans cette région :

Puits	P1	P2	P3	P4	P5	P6	P7
P1	-	886	649	757	431	659	350
P2	886	-	768	1079	693	324	1005
P3	649	868	-	416	707	572	567
P4	757	1079	416	-	990	869	500
P5	431	693	707	990	-	486	659
P6	659	324	572	869	486	-	659
P7	350	1005	567	500	659	781	-

Tableau 2.1. Distances séparant les différents puits (en m)

Définissons les différentes variables nécessaires pour effectuer la formulation mathématique du notre problème problème .

- d_{ij} = Distance entre le puits i et le puits j ; et n = Nombre des puits ;
- Q : la durée maximale de travail du l'appareil ;
- q_i : la durée de fermeture du puits i .

X_{ij} = Variable binaire qui prend la valeur 1 si le puits i est visitée immédiatement avant le puits j . Sinon, cette variable prend la valeur 0

2.3.2.2. Le modèle mathématique

Le modèle général s'écrit sous la forme :

$$\left\{ \begin{array}{l} \text{Min } z = \sum_{i=1}^n \sum_{j=1}^n d_{ij} X_{ij} \end{array} \right. \quad (1)$$

les contraintes sont les suivantes :

$$\sum_{j=1, j \neq i}^n X_{ij} = 1 \quad i = 1, \dots, n \quad (2)$$

$$\sum_{i=1, i \neq j}^n X_{ij} = 1 \quad j = 1, \dots, n \quad (3)$$

$$\sum_{i \in S} \sum_{j \in S} X_{ij} \leq |S| - 1 \quad \text{pour toute } S \in \mathbb{N} \quad (4)$$

$$\sum_{i=1}^n q_i \leq Q \quad i = 1, \dots, n \quad (5)$$

$$\forall i = 1 \dots n, \forall j = 1 \dots n : X_{ij} \in \{0,1\} \quad (6)$$

La contrainte (2) s'assure qu'on ne sort qu'une seule fois de chacun des puits tandis que la contrainte (3) vérifie que l'on entre seulement une fois à chaque puits. Ces contraintes permettent donc de visiter tous les puits une et une seule fois. Par contre, si l'on n'ajoute pas la contrainte (4) des sous tours se formeront. La quatrième contrainte élimine donc tous les sous-tours possibles. Dans cette contrainte $|S|$ dénote la cardinalité de l'ensemble S . Cette contrainte doit être décrite pour tous les sous-ensembles S de \mathbb{N} . La contrainte (5) permet de respecter la capacité de travail d'un appareil.

La modélisation du problème a permis d'obtenir un programme linéaire en variable bivalente

2.4. Conclusion :

Dans ce chapitre nous avons présenté des formulations mathématiques de nos problèmes et par suite nous avons obtenu des représentations de la réalité qui permettra de mettre en évidence les différents paramètres du chaque problème.

La modélisation du problème 1 a permis d'obtenir un programme linéaire en variable bivalentes. L'analyse du deuxième modèle mathématique, montre qu'on est en présence d'un problème non linéaire, à variable mixte, par contre le troisième modèle est un problème linéaire dont les variables bivalentes.

Après l'étape modélisation du problème, vient l'étape choix de la méthode de résolution, c'est le but du chapitre suivant.



Chapitre 3

Optimisation combinatoire et méthodes d'optimisations métaheuristiques : Théories et méthodes de résolution du problème 1



3.1. Introduction :

Dans l'analyse d'un système, il est important de bien faire la distinction entre le fond des techniques utilisées et le cadre méthodologique de leur emploi, et de ne pas être tenté de vouloir réduire ce qui est un esprit et une méthode à un simple arsenal de techniques plus ou moins sophistiquées. Et d'ailleurs, ces techniques s'affinent ou évoluent, le cadre méthodologique, lui-même, ne changera guère.

En essayant d'être aussi rationnel que possible, nous avons pu identifier notre travail à la logique schématique que nous traçons ci-dessous et qui représente la méthodologie générale de l'analyse des systèmes.

C'est la théorie de complexité des algorithmes qui, en donnant un sens précis aux termes d'algorithmes « efficaces » a permis d'asseoir l'optimisation combinatoire sur une base théorique solide. C'est ainsi que la complexité d'un problème, d'un système peut être mesurée par :

- Le nombre de ses composants.
- La nature du système : les systèmes indécidables sont plus complexes que les systèmes déterministes.
- On peut, d'autre part, prendre en compte la quantité d'informations internes au Système, ce qui conduit à considérer comme plus simple, un système ayant plusieurs composants identiques, par rapport à un système ayant un même nombre de composants mais tous différents.
- On peut tenir compte également de la complexité géométrique du système c'est ce que l'on fait, par exemple, en distinguant entre un système discret et un système continu.

Notant que certains systèmes, complexes lorsque leur description est exacte, peuvent devenir moins complexes à certain degré d'approximation.

La notion de complexité des problèmes est donc très importante, car si un problème est identifié comme facile, on connaît un algorithme fini et efficace pour le résoudre, par contre s'il est identifié comme étant un problème complexe il sera difficile de trouver un algorithme efficace pour le résoudre, il est alors justifié de se contenter d'exigences plus limitées : résolution approchée du problème posé, résolution d'un problème voisin plus simple.

L'exécution des méthodes dites exactes (programmation dynamique, séparation et Evaluation) Pour la résolution des problèmes NP-Difficiles risque de prendre un temps de calcul considérable, notamment si la taille du problème est très grande.

Afin d'éviter ce genre de situation on se contente souvent d'une solution dite approchée nous utilisons certaines méthodes appelées "méthodes approchées" ou "heuristiques", une solution dont la valeur de la fonction objectif correspondant se rapproche de celle de la solution exacte.

3.2. Optimisation combinatoire

3.2.1. Problème d'optimisation combinatoire

Un problème d'optimisation combinatoire est défini par un ensemble d'instances. A chaque instance du problème est associé un ensemble discret de solutions S , un sous-ensemble X de S représentant les solutions admissibles (réalisables) et une fonction de coût f (ou fonction objectif) qui assigne à chaque solution $x \in X$ le nombre réel (ou entier) $f(x)$. Résoudre un tel problème (plus précisément une telle instance du problème) consiste à trouver une solution $x^* \in X$ optimisant la valeur de la fonction de coût f . Une telle solution x^* s'appelle une *solution optimale* ou un *optimum global*. Nous avons donc la définition suivante :

Définition 3.1 [11] Une *instance* I d'un problème de minimisation est un couple (X, f) où $X \subseteq S$ est un ensemble fini de solutions admissibles, et f une fonction de coût (ou objectif) à minimiser $f : X \rightarrow R$. Le problème est de trouver x^* tel que : $f(x^*) \leq f(x)$ pour tout élément $x \in X$.

Notons que d'une manière similaire, on peut également définir les problèmes de maximisation en remplaçant simplement \leq par \geq .

La résolution des problèmes combinatoires est assez délicate puisque le nombre fini de solutions réalisables croît généralement avec la taille du problème, ainsi que sa complexité.

Avant d'aborder les différentes méthodes de résolution des problèmes d'optimisation combinatoire nous introduisons quelques définitions et notions sur la complexité.

3.2.2. La théorie de la complexité

Pour tout problème « p » et un algorithme A le résolvant, il se pose pour ce dernier les problèmes suivants : [12]

- ❖ **La validité de A :** C'est prouver que pour toute donnée de « p », A est fini : soit A s'arrêtera au bout d'un nombre fini d'étapes et fournira une solution de p .
- ❖ **Complexité temporelle – complexité spatiale :** C'est évaluer le temps d'exécution de A, et la place mémoire utilisée par ce dernier.

Dans la suite, nous intéresserons à la complexité temporelle des algorithmes.

Nous définissons les trois classes de problèmes :

1- La classe P

On dira qu'un problème $p \in P$ s'il existe un algorithme A tel que $T_A(n) \in O(\Pi(n))$, où Π est un polynôme en n , et $T_A(n)$, la taille de la plus grande donnée traitée de p .

P est la classe des problèmes polynomiaux.

2-La classe NP

Un problème de décision : est un problème dont l'ensemble des réponses se trouve dans { Vrai , Faux }.

Un problème de décision $p \in NP$ si et seulement si on peut vérifier qu'une donnée est à réponse positive en temps polynomial.

Remarque 3.1. La notation NP est souvent mal interprété, NP ne signifie pas Non Polynomial. (En anglais : Non Determinist polynomial).

Transformation polynomiale : Soient p_1 et p_2 deux problèmes de décision. On notera $D(p_i)$ l'ensemble des données de p_i .

On dira que p_1 se transforme polynomialement en p_2 s'il existe :

$f : D(p_1) \rightarrow D(p_2)$ telle que :

1. $\forall d \in D(p_1)$, le temps de calcul de $f(d)$ est borné par un polynôme.
2. d est une donnée à réponse positive de p_1 si et seulement si $f(d)$ est une donnée à réponse positive de p_2 . On notera alors : $p_1 \alpha p_2$.

3-La classe NPC

Soit p un problème de décision, on dira que $p \in NPC$ ou que p est un problème NP-complet si :

- $p \in NP$
- $\forall q \in NP, q \alpha p ; q$ problème de décision.

Remarques 3.2.

1. NPC est une classe d'équivalence dans NP.

2. P est une classe des problèmes les plus faciles de NP ; les plus difficiles sont ceux de NPC.

En général, dans les algorithmes se rapportant aux graphes, on montre qu'un problème est NP-complet par comparaison avec un autre problème connu NP- complet, en utilisant la transformation décrite ci- dessus.

On peut, alors, représenter les différentes classes comme suit :

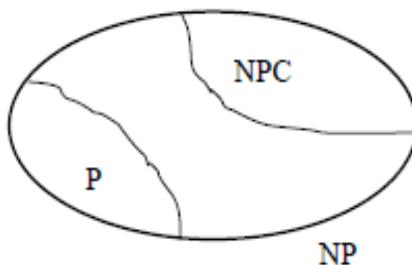


Fig.3.1. les différentes classes de la complexité

3.2.2. Les méthodes d'optimisation combinatoire :

Les problèmes d'optimisation combinatoire, sont des problèmes NP difficiles et donc ne possèdent pas à ce jour un algorithme efficace, c'est à dire ; de complexité polynomiale, valable pour toutes les données. Ceci a motivé les chercheurs à développer de nombreuses méthodes de résolution en recherche opérationnelle et en intelligence artificielle. Ces méthodes appartiennent à deux grandes familles : les méthodes exactes et les méthodes approchées [15].

- **Les Méthode exacte :** Le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des combinaisons de l'espace de recherche. Parmi les méthodes exactes, nous trouvons les méthodes de séparation et évaluation, dites méthodes de Branch and Bound, et la programmation dynamique
- **Méthode approchée (approximative) :** explore une sous-partie de l'espace de recherche. Les méthodes approchées sont subdivisées en deux types:

1-Heuristiques : règles empiriques simples qui ne sont pas basées sur l'analyse scientifique. Elles sont basées sur l'expérience et les résultats déjà obtenus et sur l'analogie pour optimiser les recherches suivantes. Généralement, on n'obtient pas la solution optimale mais une solution approchée.

2-Meta-heuristiques : algorithmes d'optimisation (généralement de type stochastique) combinant plusieurs approches heuristiques.

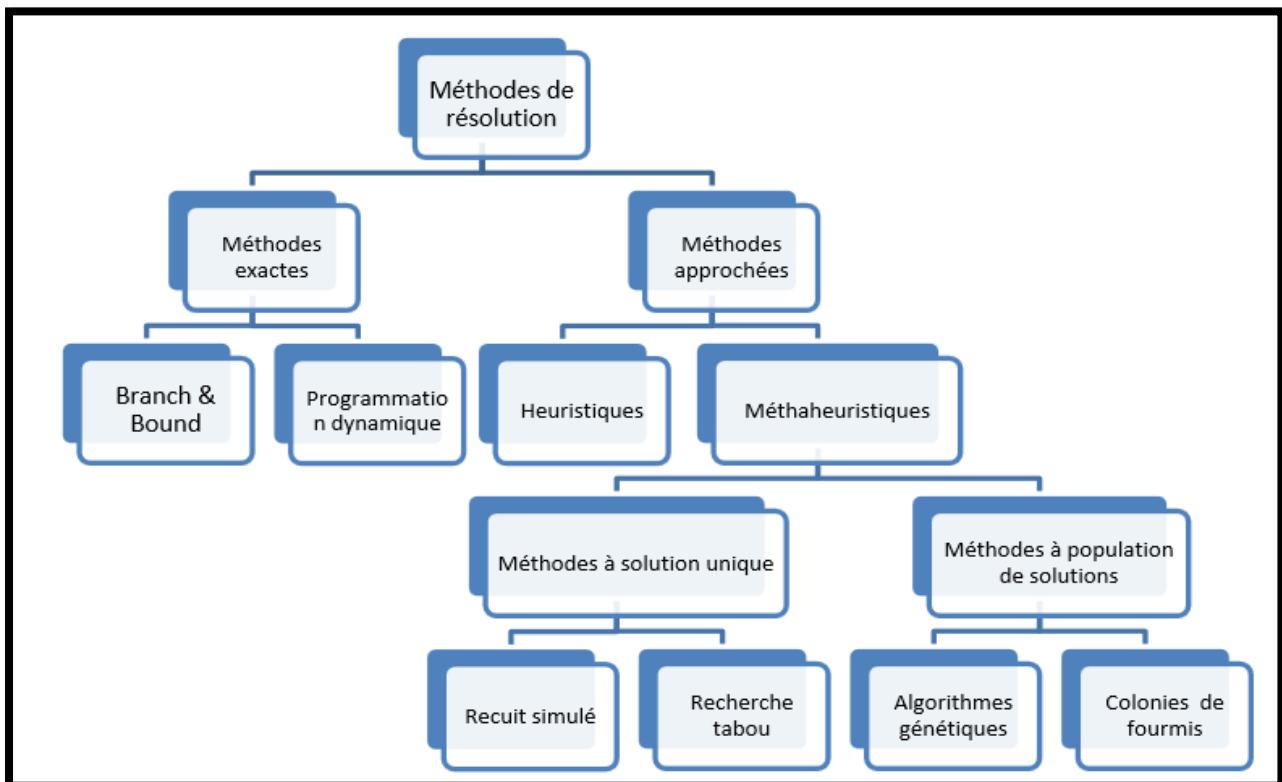


Fig.3.2 Classification des méthodes d'optimisation combinatoire

Remarque 3.3 Les méthodes exactes ne sont efficaces que pour les instances de problèmes de petite taille. On ne s'intéressera ici qu'aux méthodes approchées.

3.3. Intérêt et domaines d'applications des méthodes heuristiques

Soit le problème d'optimisation de la forme

$$\min_{x \in X} f(x)$$

$\left\{ \begin{array}{l} \text{sous des contraintes:} \\ g(x) \leq b \end{array} \right.$

La fonction f peut-être vectorielle (optimisation multi-objectifs) et les contraintes g sont généralement vectorielles. De façon générale, la fonction objective f et les contraintes g sont non linéaires.

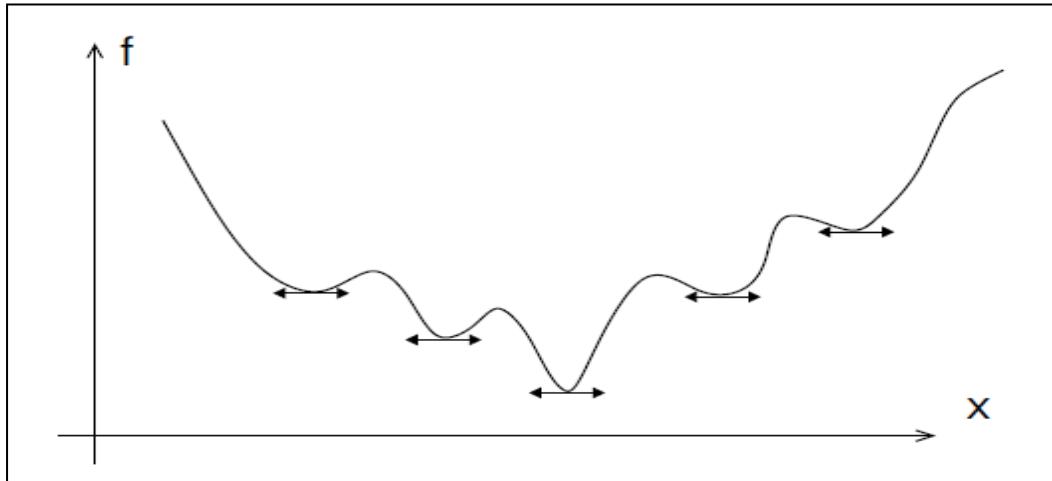


Fig.3.3 Présentation du problème d'optimisation (plusieurs minima locaux)

Difficultés :

Plusieurs minima locaux possibles (Figure 3.3) : Les méthodes classiques d'optimisation non-linéaire (basées sur le calcul différentiel) sont parfois coûteuses et incapables de capturer la solution globale [13].

Méthodes meta-heuristiques : capacité à s'extraire d'un minimum local pour aller vers un minimum global.

3.4. Méthode de Recherche Tabou

La méthode Tabou est une technique itérative générale pour l'optimisation combinatoire, introduite par FRED CLOVER en 1977 et développée plus tard. La recherche Tabou est considérée comme étant une puissante procédure d'optimisation capable d'organiser et de diriger les opérations d'une méthode subordonnée.

La Recherche Tabou a remporté un immense succès dans de nombreux problèmes d'optimisation combinatoire à savoir les problèmes d'affectations quadratiques ainsi que l'ordonnancement et autres.

3.4.1. Principe de la méthode de Recherche Tabou

La procédure de la Recherche Tabou est destinée à trouver un minimum global d'une fonction F définie sur un ensemble de solutions réalisables X . Pour chaque solution S de X on définit un voisinage $N(S)$ constitué de toutes les solutions réalisables qui peuvent être obtenues par l'application d'une simple modification m sur S .

La procédure commence par une solution réalisable initiale et tente d'atteindre un optimum global du problème par un déplacement à chaque étape, dès qu'une solution réalisable est obtenue, on génère un sous-ensemble V^* de $N(S)$ et on se déplace vers la meilleure solution S^* dans V^* . Si l'ensemble $N(S)$ n'est pas très large, il est possible de prendre $V^* = N(S)$.

L'utilité du critère du meilleur déplacement dans la Recherche Tabou est basée sur la supposition que des placements avec beaucoup d'évaluations ont de plus grandes probabilités d'atteindre une solution optimale (ou proche de l'optimal) en peu d'étapes.

Afin d'échapper aux minimums locaux, le déplacement vers S^* est effectué même si S^* est plus mauvaise que S ($F(S^*) \geq F(S)$) [14], [15].

❖ Liste tabou :

La stratégie de la Recherche Tabou peut clairement induire le cycle de l'algorithme. Afin de prévenir cela, on introduit une liste notée T . Cette liste contient les modifications effectuées dans un passé de T étapes de l'algorithme.

Lors de la construction, on introduit la génération des solutions qui sont obtenues de S par application inverse mémorisée dans la liste Tabou.

Le principe réduit le risque de cyclage dès lors que nous ne garantissons pas un retour pour un nombre donné d'irritations à une solution déjà visitée au par avant.

❖ Critère d'aspiration :

Malheureusement, la liste Tabou pourrait jouer le rôle qui consiste à interdire de se déplacer vers des nouvelles régions susceptibles de contenir de meilleures solutions. Dans le but d'avoir une plus grande liberté dans la génération d'un sous-ensemble V il devrait être possible de perdre le statut Tabou d'une modification quand il semble raisonnable de le faire. C'est pourquoi on introduit pour chaque valeur possible Z de la fonction objectif une valeur d'aspiration $A(Z)$. Une solution S dans $N(S)$ qui veut devenir Tabou à cause de la liste Tabou peut quand même être prise en compte si $F(S) \leq A(F(S))$, la fonction A est appelée fonction d'aspiration.

❖ Critère d'arrêt :

La règle pouvant être définie pour interrompre le déroulement du processus de la Recherche Tabou est d'arrêter dès que $imax$ itérations sont effectuées sans diminuer la valeur de la fonction objective qui correspond à la solution courante atteint une valeur F_{mn} de F qui est connue dans certains problèmes.

Remarque 3.4. La notation $S_i = S \oplus m_i$ veut dire S_i est obtenue à partir de S par application de la modification m_i .

3.4.2. Algorithme général de la méthode Recherche Tabou

1. Générer une solution réalisable S dans X

$S() := S$; $S()$ la meilleure solution rencontrée

$n_{iter} := 0$; Compteur d'itérations

$best_{iter} := 0$; L'itération à laquelle une meilleure solution a été trouvée.

$T := \emptyset$

Initialiser la fonction d'aspiration

2. Tant que ($F(S) \geq F_{mn}$ et $n_{iter} - best_{iter} \leq imax$) faire

Générer un ensemble V^* de solutions $S_i = S \oplus m_i$ dans $N(S)$ tel que :

$m_i \oplus T$ ou $F(S_i) \geq A(F(S))$;

Choisir la meilleure solution S^* dans V^*

Mettre à jour la liste Tabou T et la fonction d'aspiration

Si $F(S^*) \leq F(S())$ alors

$S() := S^*$;

$best_{iter} := n_{iter}$;

$S := S^*$;

Fin ;

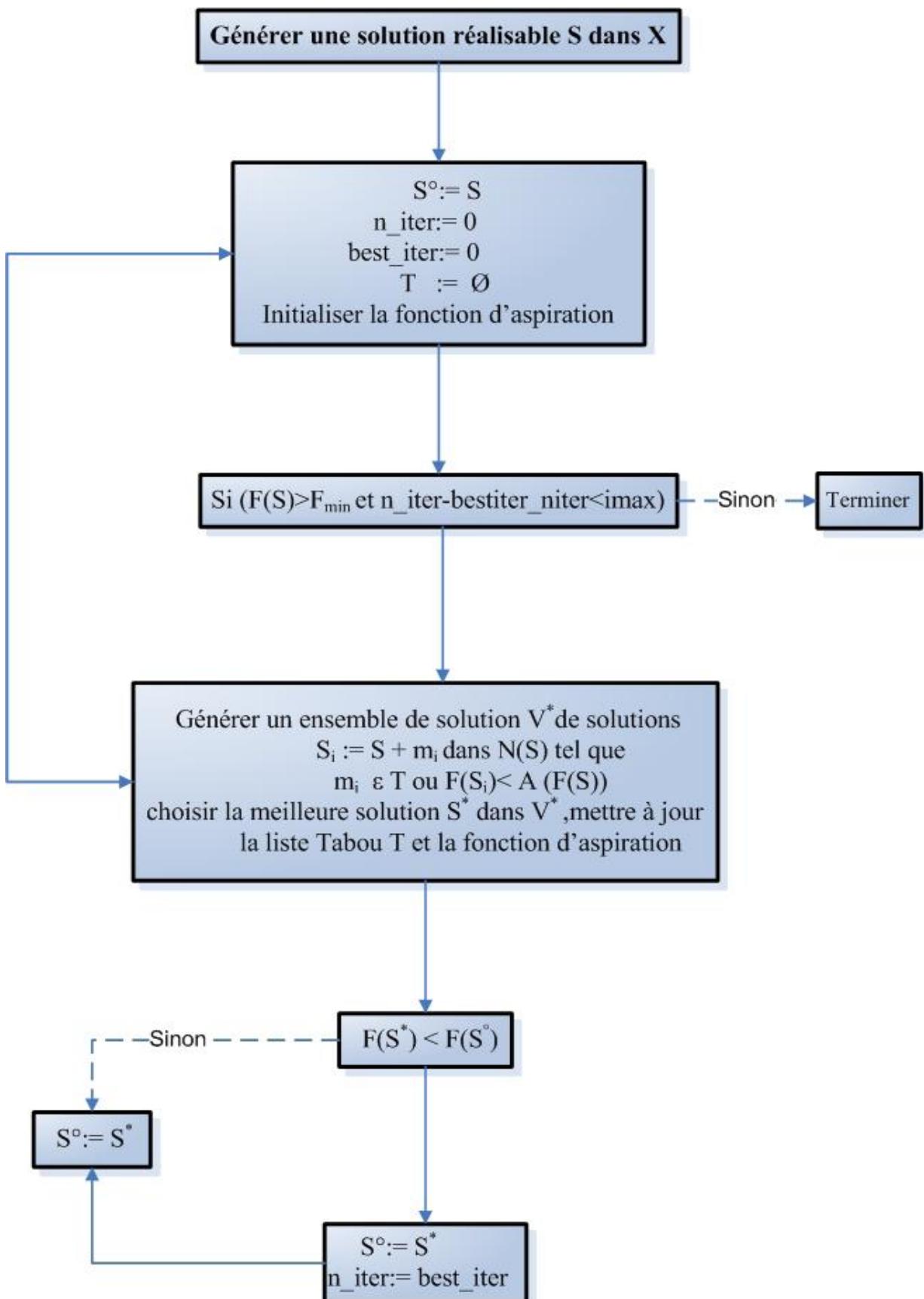


Fig.3.4 Organigramme de la méthode de Recherche Tabou

3.5. Les algorithmes génétiques :

L'algorithme génétique est une métaheuristique qui permet de résoudre des problèmes n'ayant pas de méthode de résolution décrite de manière précise, ou dont la solution exacte est connue, et trop compliquée pour être déterminée en un temps raisonnable . C'est notamment le cas quand des contraintes multiples et complexes et parfois même en partie contradictoires doivent être satisfaite simultanément [17].

3.5.1. Principe des algorithmes génétiques :

Les algorithmes génétiques (AGs) sont des algorithmes d'optimisation stochastiques fondés sur les mécanismes de la sélection naturelle de la génétique. Leur fonctionnement est extrêmement simple.

On part d'une population de solutions potentielles (chromosomes) initiales arbitrairement choisies puis on évalue leurs performances (fitness) relatives. Sur la base de ces performances on crée une nouvelle population de solutions potentielles en utilisant des opérateurs évolutionnaires simples à savoir la sélection, le croisement et la mutation. On recommence ce cycle jusqu'à ce que l'on trouve une solution satisfaisante.

3.5.2. Les composantes d'un algorithme génétique :

Un algorithme génétique est défini par :

- ❖ Individu/chromosome/séquence : une solution potentielle du problème,
 - ❖ Population : un ensemble de chromosomes ou de points
 - ❖ de l'espace de recherche,
 - ❖ Environnement : l'espace de recherche,
 - ❖ Fonction de fitness : la fonction-positive-que nous cherchons à maximiser,
 - ❖ Les opérateurs : qui permettent la construction d'une population nouvelle à partir de 1
- Avant d'aller plus loin il est nécessaire de définir quelques termes importants généralement définies sous l'hypothèse du codage binaire.

Définition 3.2

Nous appelons une *séquence* (chromosome, individu) A de longueur l une suite

$$A = \{a_1, \dots, a_l\} \quad \text{avec } i \in [1, l], \quad a_i \in \{0, 1\}$$

Un chromosome est donc une suite de bits en codage binaire, appelé aussi chaîne binaire.

1	0	0	1	0	0
---	---	---	---	---	---

Définition 3.3 :

La fitness (efficacité) est donnée par une fonction à valeurs positives réelles. Dans le cas d'un codage binaire, on utilise souvent une fonction de décodage d à une valeur réelle

$d : \{0,1\}^l \rightarrow \mathbb{R}$ (ou l est la longueur de la chaîne). La fonction de fitness est alors choisie de façon à ce qu'elle transforme cette valeur positive.

Soit $f : d(\{0,1\}) \rightarrow \mathbb{R}_+$. Le but d'un algorithme génétique est alors simplement de trouver la chaîne qui maximise cette fonction f . Bien évidemment, chaque problème particulier nécessitera ses propres fonctions d et f

3.5.3. Description détaillée des éléments de l'algorithme génétique:

3.5.3.1. Codage des données :

La première étape est de définir et de coder convenablement le problème. A chaque variable d'optimisation x_i , nous faisons correspondre un gène. Nous appelons chromosome un ensemble de gènes. Chaque solution est représentée par un individu doté d'un génotype constitué d'un ou plusieurs chromosomes. Nous appelons population un ensemble de N individus que nous allons faire évoluer.

On aboutit à une structure présentant cinq niveaux d'organisation (Figure 3.4) :

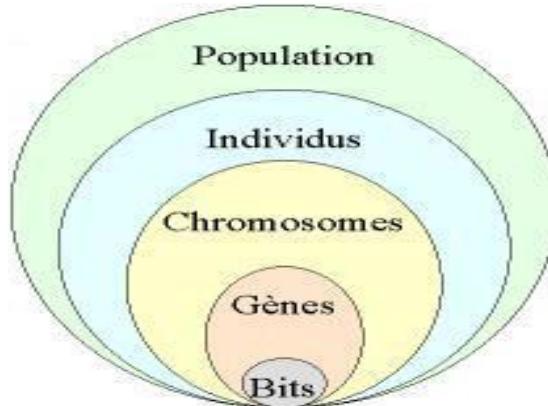


Fig. 3.5 Les cinq niveaux d'organisation de l'AG

Historiquement, le codage utilisé par les algorithmes génétiques était représenté sous forme de chaînes de bits contenant toute l'information nécessaire à la description d'un point dans l'espace d'état. Ce type de codage a pour intérêt de permettre de créer des opérateurs de croisement et de mutation simples.

Pour des problèmes d'optimisation dans des espaces de grande dimension, le codage binaire peut rapidement devenir mauvais. Généralement, chaque variable est représentée par une partie de la chaîne de bits et la structure du problème n'est pas bien reflétée, l'ordre des variables ayant une importance dans la structure du chromosome, alors qu'il n'en a pas forcément dans la structure du problème.

Les algorithmes génétiques utilisant des vecteurs réels évitent ce problème en conservant les variables du problème dans le codage de l'élément de population, sans passer par le codage binaire intermédiaire.

3.5.3.2. Un mécanisme de génération de la population initiale

Ce mécanisme doit être capable de produire une population d'individus non homogène qui servira de base pour les générations futures.

Le choix de la population initiale est important car il peut rendre plus ou moins rapide la convergence vers l'optimum global.

Dans le cas où l'on ne connaît rien du problème à résoudre, il est essentiel que la population initiale soit répartie sur tout le domaine de recherche.

3.5.3.3. Les opérateurs de l'algorithme génétique

1- Opérateur de sélection :

Cet opérateur détermine la capacité de chaque individu à persister dans la population et à se diffuser. En règle générale, la probabilité de survie d'un individu sera directement reliée à sa performance relative au sein de la population. Cela traduit bien l'idée de la sélection naturelle : les gènes les plus performants ont tendance à se diffuser dans la population tandis que ceux qui ont une performance relative plus faible ont tendance à disparaître.

Il existe plusieurs méthodes pour représenter la sélection. La méthode la plus connue et la plus utilisée est sans nul doute, la sélection par roulette biaisée (*roulette wheel*) [17].

Selon cette méthode, chaque chromosome sera dupliqué dans une nouvelle population proportionnellement à sa performance.

On effectue, en quelque sorte, autant de tirages avec remise qu'il y a d'éléments dans la population. Ainsi, dans le cas d'un codage binaire, la performance d'un chromosome particulier, A_i , étant $f(d(A_i))$, la probabilité avec laquelle il sera réintroduit dans la nouvelle population de taille N est :

$$P[A_i] = \frac{f(d(A_i))}{\sum_{j=1}^N f(d(A_j))}$$

Plus la performance d'un individu est élevée par rapport à celle des autres individus de la même population, plus il a une chance d'être reproduit dans la population. Les individus ayant une grande performance relative ont donc plus de chance d'être sélectionnés. On parle alors de sélection proportionnelle. Le nombre n_i espéré de chaque individu dans la nouvelle population qui va résulter de la sélection est alors égal à :

$$n_i = N \times P[A_i]$$

2-Opérateur de croisement :

L'opérateur de croisement permet la création de nouveaux individus selon un processus fort simple. Il permet donc l'échange d'information entre les chromosomes (individus) par le biais de leur combinaison. La population qui résulte de la sélection est divisée en deux sous populations de taille ($n/2$) et chaque couple formé par un membre provenant de chaque sous-population participe à un croisement avec une probabilité donnée (la probabilité de croisement, p_c , souvent supérieure à 60%).

Si le croisement a lieu, sa localisation entre la position 1 et la position l , dans le cas du codage binaire, est tirée selon une loi uniforme et les deux individus échangent leurs gènes des deux cotés de cette localisation.

3-Opérateur de mutation :

Le rôle de cet opérateur est de modifier aléatoirement, avec une certaine probabilité, la valeur d'un composant de l'individu. Dans le cas du codage binaire, chaque bit $a_i \in \{0,1\}$ est remplacé selon une probabilité p_m par son complémentaire $\tilde{a}_i = 1 - a_i$.

La mutation est traditionnellement considérée comme un opérateur intervenant à la marge dans la mesure où sa probabilité est en général fixée assez faible (de l'ordre de 1%). Mais elle confère aux algorithmes génétiques une propriété très importante : l'ergodicité (i.e. tous les points de l'espace de recherche peuvent être atteints). Cet opérateur est donc d'une grande importance et il est loin d'être marginal. Il a de ce fait un double rôle : celui d'effectuer une recherche locale et/ou de sortir d'une trappe (optimums locaux).

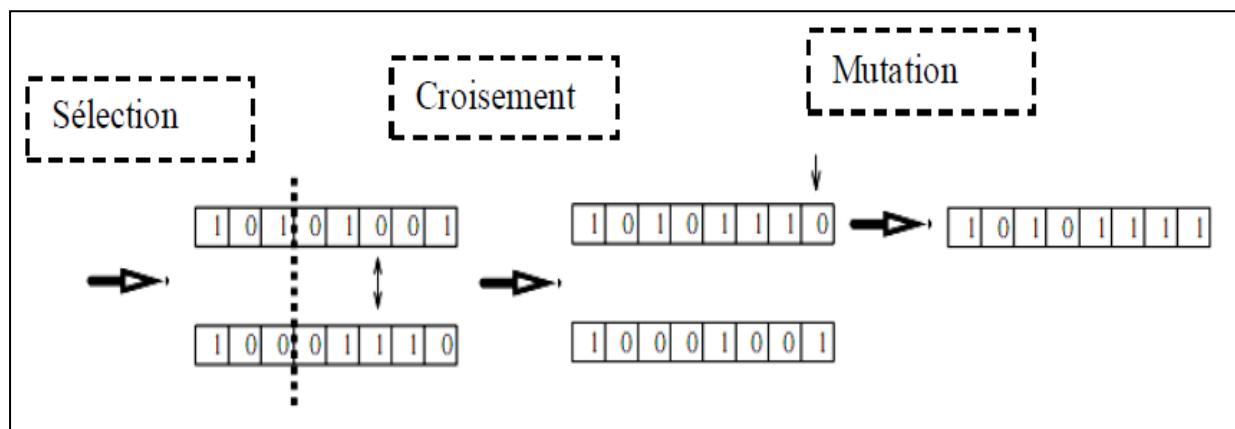


Fig.3.6. Exemple schématisé des opérateurs de sélection, croisement et mutation

3.5.4. Critère d'arrêt d'un algorithme génétique

L'arrêt de l'évolution d'un algorithme génétique est l'une des difficultés majeures car il est souvent difficile de savoir si l'on a trouvé l'optimum.

Actuellement les critères les plus utilisés sont les suivants :

- arrêt de l'algorithme après un certain nombre de générations, fixé au départ ; c'est ce que l'on est tenté de faire lorsque l'on doit trouver une solution dans un temps limité ;
- L'algorithme peut être arrêté lorsque la population n'évolue plus ou plus suffisamment rapidement, c.-à-d les meilleurs individus ne s'améliorent plus depuis un certain nombre de générations.

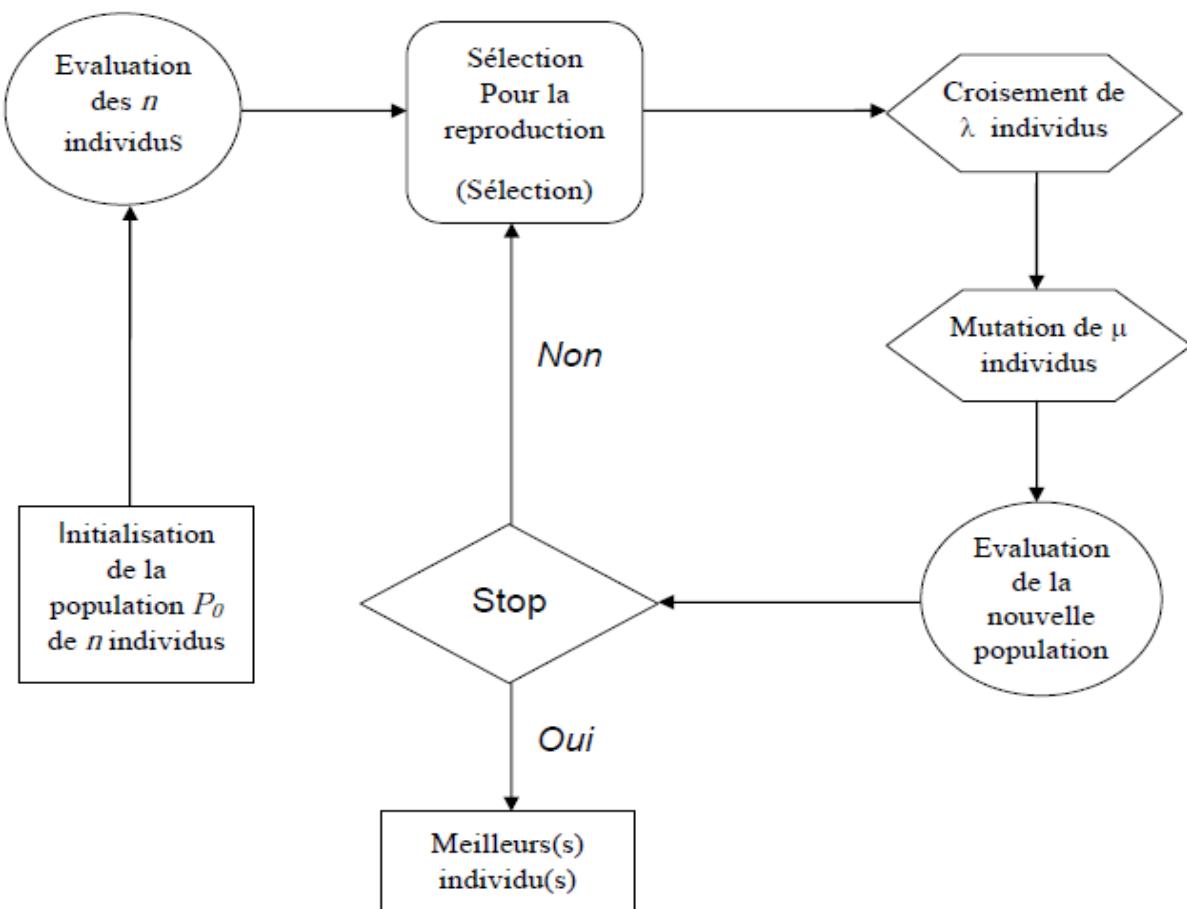


Fig.3.7 .Organigramme d'un Algorithme Génétique standard

3.5. Conclusion :

L'optimisation combinatoire occupe une place très importante en recherche opérationnelle, en mathématiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire. Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart appartiennent à la classe des problèmes NP-difficiles et ne possèdent donc pas à ce jour de solution algorithmique efficace valable pour toutes les données.

Etant donnée l'importance de ces problèmes, de nombreuses méthodes de résolution ont été développées en recherche opérationnelle. Elles peuvent être classées sommairement en deux grandes catégories : les méthodes exactes (complètes) qui garantissent la complétude de la résolution et les méthodes approchées (incomplètes) qui perdent la complétude pour gagner en efficacité.

Chapitre 4

*Programmation mathématique non linéaire
avec contraintes et problème de routage de
véhicules V.R.P: théories et méthodes de
résolution du problème 2*

4.1. Introduction :

Dans ce chapitre nous présentons les différentes approches de résolution utilisées pour le problème étudié. On les a séparées en deux parties :

- La première partie est consacrée à la théorie de programmation non linéaire avec contraintes. Cette partie tentera aussi de faire la lumière sur les méthodes les plus connues et utilisées pour la résolution de ce type de problème.
- La deuxième partie mettra en exergue quelques aspects concernant les problèmes de tournées de véhicules ainsi que leurs principales généralisations.

4.2. Programmation mathématique non linéaire avec contraintes : théories et méthodes de résolution du problème 2.1

4.2. 1. Notions fondamentales

4.2.1.1. Eléments de topologie

Définitions 4.1 (Norme euclidienne) : On appelle l'application $\|\cdot\|: \mathbb{R}^n \rightarrow \mathbb{R}$ donnée par

$$\|x\| = \left(\sum_{i=1}^n x_i^2 \right)^{\frac{1}{2}} \quad (x \in \mathbb{R}^n), \quad \text{la norme euclidienne sur } \mathbb{R}^n$$

Définitions 4.2 (Boule, voisinage) : Etant donné un point $x_0 \in \mathbb{R}^n$ et $r > 0$, on appelle l'ensemble $B(x_0, r) = \{x \in \mathbb{R}^n / \|x - x_0\| < r\}$ la boule (euclidienne) de centre x_0 et de rayon r . On note que la définition de boule dépend de la norme choisie.

On dit qu'un ensemble $U \subseteq \mathbb{R}^n$ est un voisinage d'un point $x_0 \in \mathbb{R}^n$ s'il existe un rayon $r > 0$ tel que $B(x_0, r) \subseteq U$.

Exemple 4.1. L'ensemble $U = \{x = (x_1, x_2) \in \mathbb{R}^n / x_1 \geq 0, x_2 \geq 0\}$

est un voisinage du point $(1, 1)$.

Définitions 4.3 (ensemble ouvert, fermé, borné)

- On dit qu'un ensemble $U \subseteq \mathbb{R}^n$ est ouvert si tout $x_0 \in U$, il existe un rayon $r > 0$ tel que $B(x_0, r) \subseteq U$. Autrement dit, un ensemble $U \subseteq \mathbb{R}^n$ est ouvert si et seulement si c'est un voisinage de chacun de ses éléments.
- On dit qu'un ensemble $F \subseteq \mathbb{R}^n$ est fermé si son complémentaire $F^C = \mathbb{R}^n \setminus F$ est ouvert.
- On dit qu'un ensemble $F \subseteq \mathbb{R}^n$ est borné s'il existe un rayon $r > 0$ tel que $B \subseteq B(0, r)$. Autrement dit, B est borné si et seulement si il existe une constante $r > 0$ telle que pour tout $x \in B$ on a $\|x\| \leq r$.

4.2.1.2. Convergence

Définition 4.4. Une suite $(x_k) \subseteq \mathbb{R}^n$ converge vers $x^* \in \mathbb{R}^n$ si :

$$\forall \varepsilon > 0, \exists n_0 \in \mathbb{N}, k \geq n_0 \Rightarrow \|x_k - x^*\| < \varepsilon.$$

On note $\lim_{k \rightarrow \infty} x_k = x^*$ si (x_k) converge vers x^* . On dit que (x_k) est une suite de Cauchy si : $\forall \varepsilon > 0, \exists n_0 \in \mathbb{N}, \forall k, m \geq n_0 \Rightarrow \|x_k - x_m\| < \varepsilon$.

- Finalement, on dit que, (x_k) est une suite bornée s'il existe $r > 0$ tel que pour tout k on a $\|x_k\| \leq r$.

Théorème 4.1. [18] Soit suite $(x_k) \subseteq \mathbb{R}^n$ converge vers $x^* \in \mathbb{R}^n \Rightarrow (x_k)$ est une suite de Cauchy $\Rightarrow (x_k)$ est une suite bornée.

4.2.1.3. Rappels de calcul différentiel

Définition 4.5 (Continuité) : Soit $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$, on dit que f est continue au point $x_0 \in \mathbb{R}^n$ si pour tout réel $\varepsilon > 0$ il existe $\eta > 0$ tel que pour tout $x \in \mathbb{R}^n$ on a

$$\|x - x_0\| < \eta \Rightarrow \|f(x) - f(x_0)\| < \varepsilon.$$

Définition 4.6. (Différentiabilité) Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, On dit que f est différentiable (ou dérivable au sens de FRECHET) en un point $x_0 \in \mathbb{R}^n$ s'il existe une application linéaire de \mathbb{R}^n dans \mathbb{R}^n notée $Df(x_0)$ ou $f'(x_0)$ telle que :

$$\lim_{h \rightarrow 0} \frac{|f(x_0 + h) - f(x_0) - Df(x_0)(h)|}{\|h\|} = 0$$

$Df(x_0)$ est l'application différentielle (dérivée) de f au point x_0 .

L'application dérivée $Df(x_0)$ associe à $x_0 \in \mathbb{R}^n$ un vecteur de \mathbb{R}^n appelée aussi gradient de f au point x_0 est noté $\nabla f(x_0)$.

Définition 4.7. (Le gradient et la matrice hessienne) : Le gradient d'une fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continûment différentiable évalué au point $x \in \mathbb{R}^n$ s'écrit :

$$\nabla f(x^*) = \begin{pmatrix} \frac{\partial f(x)}{x_1} & \frac{\partial f(x)}{x_2} & \frac{\partial f(x)}{x_3} & \dots & \frac{\partial f(x)}{x_n} \end{pmatrix}^T$$

et l'élément de la i^{eme} ligne et la j^{eme} colonne de la matrice Hessienne s'écrit :

$$[\nabla^2 f(x)]_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}.$$

Définition 4.8. L'application $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est dite deux fois différentiable en un point a s'il existe une matrice symétrique $\nabla^2 f(x)$ tel que

$$f(a+h) = f(a) + \nabla f(a)^T h + h^T \nabla^2 f(a) h + \|h\| \varepsilon(h)$$

Où $\nabla^2 f(x)$ est la matrice hessienne

Définition 4.9 (le sous-gradient) : on appelle sous gradient de f au point x^* tout vecteur $y \in \mathbb{R}^n$ vérifiant $f(x) \geq f(x^*) + y^T(x - x^*)$

L'ensemble de tous les sous-gradients de f en x^* , noté $\partial f(x^*)$ est appelé sous différentiel de f au point x^*

4.2.1.4. Rappels de quelques outils d'analyse convexes

Définition 4.10 (Ensemble convexe) : Soit l'ensemble $D \subseteq \mathbb{R}^n$. D est dit *convexe* si et seulement si :

$$\forall x, y \in D, \forall \lambda \in [0,1], \text{ alors } \lambda x + (1-\lambda)y \in D.$$

Exemples 4.2.

- Le polytope $p = \left\{ x \in \mathbb{R}^n : x = \sum_i \lambda_i x_i, \sum_i \lambda_i = 1, \lambda_i \geq 0 \right\}$ est un ensemble convexe

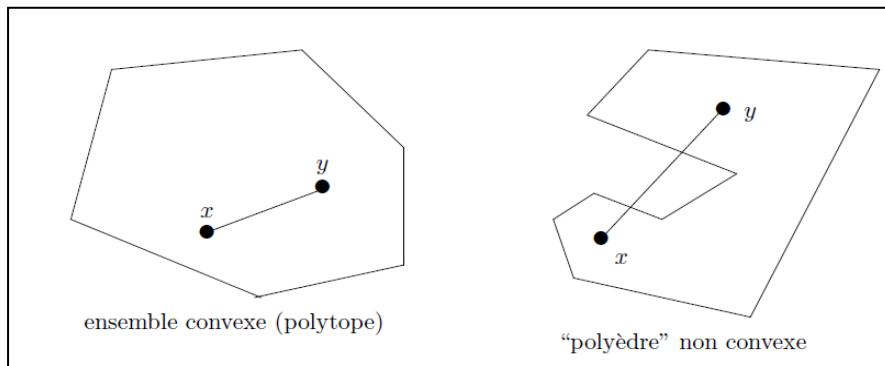


Fig.4.1. Exemple d'ensembles convexes et non convexes

Définition 4.11 (Fonction convexe) : Soit l'ensemble convexe $D \subseteq \mathbb{R}^n$. Une fonction $f : D \rightarrow \mathbb{R}$ est convexe si et seulement si

$$\forall x, y \in D, \forall \lambda \in [0,1], \text{ alors } f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y).$$

Définition 4.12 (la fonction conjuguée) : soit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction convexe. On appelle fonction conjuguée de f , notée f^* , la fonction définie par :

$$\forall y \in \mathbb{R}^n, f^*(y) = \sup_{x \in \mathbb{R}^n} \{ y^t x - f(x) \}$$

La fonction conjuguée de f^* notée f^{**} , est appelée fonction bi-conjuguée de f .

On dit que f est une fonction convexe fermée si et seulement si $f^{**} = f$

Définition 4.13 (le sous-différentiel) : pour une fonction convexe et différentiable, le

gradient $\nabla f(x^*)$ en tout point x^* vérifie $f(x) \geq f(x^*) + \nabla f^t(x^*)(x - x^*)$

Théorème 4.2 [19]: Soit $f : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ continûment différentiable sur un domaine Convexe D . f est une fonction convexe sur D si et seulement si la matrice hessienne est semi- définie positive, c'est-à-dire : pour tout $x \in D$, $y^T \nabla^2 f(x) y \geq 0$, $\forall y \in \mathbb{R}^n$. de même si la matrice hessienne est définie positive c'est-à-dire : pour tout $x \in D$, $y^T \nabla^2 f(x) y > 0$, $\forall y \neq 0 \in \mathbb{R}^n$ alors f est une fonction strictement convexe sur D

Remarque 4.1. Une matrice A est semi-définie (définie) positive si et seulement si les déterminants des mineurs principaux de A sont (strictement) positifs

4.2.2. Programmation mathématique

4.2.2.1. Formulation générale de programmation mathématique

Un programme mathématique (PM) est un problème d'optimisation de la forme :

$$(PM) \left\{ \begin{array}{l} \min_{x \in \mathbb{R}^n} f(x) \\ s.c \\ D = \{x \in \Omega \subseteq \mathbb{R}^n / g_i(x) \leq 0, i=1, \dots, m \text{ et } h_j(x) = 0, j=1, \dots, p\} \end{array} \right.$$

Où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction continûment différentiable appelée fonction objectif et D est appelé ensemble des solutions réalisables, avec g et h sont des fonctions définies de \mathbb{R}^n dans \mathbb{R}

La formulation mathématique (PM) signifie que l'on cherche à trouver la solution $x^* \in D$ dont la valeur de la fonction objectif est la plus petite.

Définitions 4.14

- Une solution réalisable qui minimise l'objectif sur D est dite solution optimale globale de (PM). On note par $\arg \min_D f(x)$ l'ensemble des solutions optimales globales

- Un point $x^* \in D$ est une solution optimale locale de (PM) si : $\exists V$ (voisinage) de x^* tel que : $f(x^*) \leq f(x), \forall x \in V$ et on note par $\text{loc min}_D f(x)$ l'ensemble des solutions optimales locales de (PM).

Remarque 4.2 : Nous avons toujours $\arg \min_D f(x) \subseteq \text{loc} \min_D f(x)$ et si (PM) est convexe les deux ensembles sont égaux.

Remarque 4.3 : Notons que le minimum global n'est pas nécessairement unique, mais la valeur optimale l'est. Par exemple, $\min_{x \in \mathbb{R}} \sin x$ possède une infinité de minima globaux.

4.2.2.2. Classification d'un programme mathématique

On classifie le problème (PM) à partir de deux propriétés fondamentales à savoir la convexité et la différentiabilité des fonctions du problème. A ce propos, (PM) est dit convexe si f et g_i sont convexes et h_j affines. Si ces dernières sont toutes différentiables on dit que (PM) est différentiable. La classe de programmes mathématiques convexes différentiables est le modèle le mieux élaboré, les programmes non convexes ou non différentiables sont difficiles à traiter. On peut aussi classer les problèmes d'optimisation en fonction des caractéristiques mathématiques de la fonction objective, des contraintes et des variables d'optimisation (Tableau 4.1).

Caractéristiques	Propriétés	Classification
Nombre de variables	Une seule variable plus d'une variable	Monovariable Multivariable
Type de variables	Réelles Entières Réelles et entières Entières avec permutation	Continue Discrète Mixte Combinatoire
Type de fonction objectif	Linéaire en fonction des variables Quadratique en fonction des variables Non linéaire en fonction des variables	Linéaire Quadratique Non linéaire
Formulation du problème	Soumis à des limitations Pas de limitations	Avec contraintes Sans contraintes

Tab.4.1. Classifications des problèmes de programmation mathématique

4.2.2.3. Qualification des contraintes

- ❖ Si D est un polyèdre convexe (i.e. : toutes les fonctions contraintes sont affines), alors par définition les contraintes sont qualifiées en tout point réalisable.
- ❖ Si D est convexe (i.e. : g_i convexe et h_j affines) et $\text{int } D \neq \emptyset$ les contraintes sont qualifiées partout. C'est la condition de Slater.
- ❖ Une contrainte d'inégalité $g_i \leq 0$ est dite saturée (ou active) en $x^* \in D$ et $g_i(x^*) = 0$

De ce fait, une contrainte d'égalité $h_j = 0$ est par définition saturée en tout point $x^* \in D$. les contraintes sont qualifiées en $x^* \in D$ si les gradients de toutes les fonctions des contraintes saturées en x^* sont linéairement indépendants [20].

4.2.2.4. Résolution d'un programme mathématique

La résolution complète de (PM) traite dans l'ordre les points suivants:

- L'existence (et éventuellement l'unicité) d'une solution optimale.
- La caractérisation de la solution (il s'agit des conditions d'optimalité).
- L'élaboration d'algorithmes pour calculer cette solution.

4.2.2.4.1. Principaux résultats d'existence

Théorème 4.3 (Weierstrass) [21] : Soit D compact non vide de \mathbb{R}^n , si f est continue sur D alors (PM) admet au moins une solution optimale globale $x^* \in D$.

Théorème 4.4 [21] : Soit D non vide et fermé de \mathbb{R}^n , f est continue et coercive sur D (c'est à dire $\lim_{\|x\| \rightarrow \infty} f(x) = +\infty$), alors (PM) admet au moins une solution optimale globale.

Remarque 4.4 : L'unicité d'une solution éventuelle est en général une conséquence de la convexité de D et de la stricte convexité de f .

4.2.2.4.2. Conditions d'optimalité

4.2.2.4.2.1. Cas sans contraintes

Conditions nécessaires

Théorème 4.5 (condition nécessaire du premier ordre) [22]: Soient $f : \mathbb{R}^n \rightarrow \mathbb{R}$ continuellement différentiable et $x^* \in \mathbb{R}^n$. Si x^* est un minimum local de f , alors $\nabla f(x^*) = 0$.

Remarque 4.5 : Si f est convexe, la condition nécessaire du premier ordre est également suffisante pour que x^* soit un minimum global.

Théorème 4.6 (condition nécessaire du second ordre) [22]: soient $f : \mathbb{R}^n \rightarrow \mathbb{R}$ deux fois continuellement différentiable et $x^* \in \mathbb{R}^n$. Si x^* est un minimum local de f , alors $\nabla f(x^*) = 0$ et $\nabla^2 f(x^*) = 0$.

Conditions suffisantes

Théorème 4.7 (condition suffisante du second ordre) [22]: soient $f : \mathbb{R}^n \rightarrow \mathbb{R}$ deux fois continuellement différentiable et $x^* \in \mathbb{R}^n$. Si $\nabla f(x^*) = 0$ et $\nabla^2 f(x^*)$ est définie positive, alors x^* est un minimum local de f .

4.2.2.4.2. Cas avec contraintes

Pourquoi avons-nous besoin de conditions d'optimalité ?

Afin d'analyser ou de résoudre d'une manière efficace un problème d'optimisation, il est fondamental de pouvoir disposer des conditions d'optimalité. En effet, celles-ci nous servent non seulement à vérifier la validité des solutions obtenues, mais souvent l'étude de ces conditions aboutit au développement des algorithmes de résolution eux-mêmes.

L'approche considérée ici pour l'obtention de conditions est basée sur les notions de descente et de direction admissible.

On considère le programme mathématique suivant :

$$(PM) \left\{ \begin{array}{l} \min f(x) \\ s.c \\ D = \{x \in \Omega \subseteq \mathbb{R}^n / g_i(x) \leq 0, i=1, \dots, m \text{ et } h_j(x) = 0, j=1, \dots, p\} \end{array} \right.$$

Où f , g_i et h_j sont des fonctions définies de \mathbb{R}^n dans \mathbb{R} et sont continûment différentiables.

Définitions 4.15 :

- une direction d est dite admissible en x_0 si et seulement si :

$$\exists \alpha^* > 0, \forall \alpha \in [0, \alpha^*[, x = x_0 + \alpha.d \in D$$

- Un point admissible x_0 est régulier ssi les gradients en x_0 des contraintes inégalités saturées en x_0 et les gradients en ce même point des contraintes égalités forment une famille libre.

Problème avec contraintes d'égalités

On va tout d'abord s'intéresser au problème suivant, dit problème d'optimisation avec contraintes d'égalité seulement (PMCE):

$$(PMCE) \left\{ \begin{array}{l} \min_{x \in D \subseteq \mathbb{R}^n} f(x) \\ s.c \\ h_j(x) = 0, j=1, \dots, p \end{array} \right.$$

Théorème 4.8 [23] : Si $x^* \in D$ est un minimum local de f sur D et si les vecteurs $\nabla h_1(x^*), \dots, \nabla h_p(x^*)$ sont linéairement indépendants (i.e. x^* est un point régulier), il existe alors $\lambda^* = (\lambda_1^*, \dots, \lambda_p^*) \in \mathbb{R}^p$ tel que :

$$\nabla f(x^*) + \sum_{j=1}^p \lambda_j^* \nabla h_j(x^*) = 0$$

Les $(\lambda_j^*)_{1 \leq j \leq p}$ sont appelés multiplicateurs de Lagrange.

Soit $(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^p$. On peut formuler une autre condition nécessaire en utilisant la fonction de Lagrange (ou Lagrangien) définie par :

$$L(x, \lambda) = f(x) + \sum_{j=1}^p \lambda_j h_j(x).$$

Théorème 4.9 [23] : Si $x^* \in D$ est un minimum local de f sur D et si les vecteurs $\nabla h_1(x^*), \dots, \nabla h_p(x^*)$ sont linéairement indépendants (i.e. x^* est un point régulier), il existe alors $\lambda^* = (\lambda_1^*, \dots, \lambda_p^*) \in \mathbb{R}^p$ tel que :

$$\nabla L(x^*, \lambda^*) = 0.$$

Problème avec contraintes d'égalités et d'inégalités

Nous reprenons le problème général initial (PM) avec contraintes d'égalités et d'inégalités :

Théorème 4.10 (*Karush-Kuhn-Türck, KKT*) [21]:

On suppose que l'hypothèse de qualification des contraintes (QC) est vérifiée en x^* solution de (PM). Une condition nécessaire pour que x^* soit un optimum local de (PM) est qu'il existe des multiplicateurs $\lambda^* \in \mathbb{R}_+^m$ et $\mu^* \in \mathbb{R}^p$ tel que :

$$\begin{cases} \nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^p \mu_j^* \nabla h_j(x^*) = 0 & \text{(condition d'optimalité)} \\ \lambda_i^* g_i(x^*) = 0, \quad i = 1, \dots, m & \text{(condition de complémentarité)} \\ h_j(x^*) = 0, \quad j = 1, \dots, p \end{cases}$$

Deux des principales conditions suffisantes connues pour que (QC) soit vérifiée sont [23] :

- 1) les fonctions g_i sont convexes, les fonctions h_j sont linéaires et il existe $x \in D$ tel que $g_i(x) < 0, \forall i$ et $h_j(x) = 0, \forall j$.
- 2) En $x^* \in D$, les gradients $\nabla g_i(x^*)$ et $\nabla h_j(x^*)$ sont linéairement indépendants.

Comme à la section précédente, on peut définir une fonction de Lagrange en posant :

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{j=1}^p \mu_j h_j(x)$$

Remarque 4.6

- ❖ Si les contraintes ne sont pas qualifiées en x^* , les conditions de KKT ne s'appliquent pas (x^* peut être optimal sans vérifier ces conditions).
- ❖ Si (PM) est convexe, les conditions de KKT sont à la fois nécessaires et suffisantes pour que x^* soit un minimum global.

4.2.3. Aperçu sur les méthodes de programmation mathématique avec Contraintes

Nous reprenons le problème général initial (PM) avec contraintes d'égalités et d'inégalités :

$$(PM) \left\{ \begin{array}{l} \min f(x) \\ s.c \\ D = \{x \in \Omega \subseteq \mathbb{R}^n / g_i(x) \leq 0, i=1, \dots, m \text{ et } h_j(x) = 0, j=1, \dots, p\} \end{array} \right.$$

Nous décrivons les principales méthodes itératives implémentées pour ce problème. Il s'agira des classes de méthodes suivantes :

1. Les méthodes de directions admissibles
2. Les méthodes de plans sécants
3. Les méthodes de pénalité intérieure (barrière)
4. Les méthodes de pénalité extérieure

Remarque 4.7 : toute contrainte sous forme d'égalité peut être exprimée de manière équivalente par deux inégalités :

$$h_j(x) \text{ peut être exprimée sous la forme } \begin{cases} h_j(x) \leq 0 \\ -h_j(x) \leq 0 \end{cases}$$

Au travers des sections suivantes, nous admettrons donc que D n'est défini que par une collection de m inégalité $g_i(x) \leq 0$, puisque cela n'induit aucune perte de généralité. Nous agirons de la sorte pour toutes les méthodes sauf la méthode de barrière, qui nous verrons pourquoi, n'est pas en mesure de traiter de contraintes sous forme d'égalités.

4.2.3.1. Méthode des directions admissibles

Dans cette méthode, nous nous intéressons au problème suivant :

$$\left\{ \begin{array}{l} \min_{x \in D \subseteq \mathbb{R}^n} f(x) \\ S.C \\ g_i(x) \leq 0, i = 1, \dots, m \end{array} \right.$$

Principe de la méthode :

Cette classe de méthodes résout un problème de minimisation non linéaire en se déplaçant d'un point de D vers un autre de ses points au coût inférieur. Elles fonctionnent selon le principe suivant : étant donné un élément x^k de D , une direction d^k est générée telle que pour un $\alpha^k > 0$ suffisamment petit, les propriétés suivantes sont assurées :

- 1- $x^k + \alpha^k d^k$ appartient toujours à D ;
- 2- $f(x^k + \alpha^k d^k) < f(x^k)$.

Une fois d^k déterminée, α^k s'obtient par minimisation monodimensionnelle pour que le déplacement dans la direction d^k soit optimale, mais cette fois-ci il est nécessaire d'imposer une borne supérieure sur la valeur de α^k afin de ne pas sortir de D . Cela définit le nouveau point x^{k+1} et le processus est recommandé.

Il existe plusieurs stratégies pour la résolution du sous-problème consistant à déterminer d^k . Comme nous allons le voir, il peut être exprimé sous forme d'un programme linéaire. Parmi ces méthodes on peut citer par exemple : la méthode de Frank et Wolfe [22].

4.2.3.1.1. La méthode de Frank et Wolfe :

Description de la méthode

Cette méthode s'applique si les contraintes sont linéaires et si D est borné. Une manière simple de générer une direction d^k satisfaisant la condition de descente $\nabla f(x^k) \cdot d^k < 0$, si l'on

pose $d^k = \bar{x}^k - x^k$, est de minimiser la dérivée directionnelle de f dans la direction d^k , comme c'est le cas pour la méthode du gradient ; mais il faut en plus prendre garde de pas sortir de l'ensemble des solutions admissibles (le point \bar{x}^k généré doit lui-même appartenir à D , de sorte que le point $x^k + d^k$ soit lui-même admissible). Le sous-problème de recherche de d^k peut être formulé ainsi :

$$\begin{cases} \text{Min } \nabla f(x^k) \cdot (x - x^k) \\ \text{sous contraintes } x \in D \subseteq \mathbb{R}^n \end{cases}$$

et nous pouvons obtenir \bar{x}^k comme la solution optimale de ce sous-problème. Celui-ci est un programme linéaire (les contraintes du problème de départ le sont elles-mêmes) et peut être résolu par l'algorithme du simplexe. Selon un théorème connu, sa solution optimale \bar{x}^k se trouvera alors en l'un des points extrêmes du domaine D , de sorte que l'optimisation monodimensionnelle le long de d^k devra être faite sous la restriction $0 \leq \alpha^k \leq 1$. La méthode se termine si la direction générée est égale au vecteur nul.

Convergence de la méthode

Théorème 4.11 [22]. Considérons le problème de minimisation de $f(x)$ sous contraintes $x \in D \subseteq \mathbb{R}^n$ où D est un polytope borné. Alors la méthode de Frank et Wolfe démarrée depuis un point initial admissible converge vers un point satisfaisant la condition de Kuhn-Tucker.

4.2.3.2. Les méthodes de pénalité intérieure (ou méthodes de barrière)

Le principe de ces méthodes réside dans la transformation d'un problème contraint en une séquence de problèmes sans contraintes, en ajoutant au coût une pénalité en cas de violation de celles-ci. Un tel sous-problème est résolu à chaque itération d'une méthode de pénalité.

L'appellation "pénalité intérieure" est employée car le minimum est approché depuis l'intérieur de D ; les méthodes de pénalité extérieure, elles, approchent le minimum depuis l'extérieur de cet ensemble.

Les méthodes de barrière s'appliquent aux problèmes dont l'ensemble admissible D est défini uniquement par une collection d'inégalités :

$$\begin{cases} \underset{x \in D \subseteq \mathbb{R}^n}{\text{Min}} f(x) \\ \text{s.c } g_i(x) \leq 0, i = 1, \dots, m \end{cases}$$

Où f et g_i sont des fonctions de \mathbb{R}^n dans \mathbb{R} et sont continues. En effet, ces méthodes utilisent des fonctions dites de barrière, définies uniquement à l'intérieur de D . Si des contraintes sous forme d'égalités étaient introduites, l'intérieur de cet ensemble, c'est-à-dire son sous-ensemble tel qu'en chacun de ses points aucune contrainte n'est active, serait clairement vide. C'est donc le domaine de définition de la fonction de barrière qui serait vide rendant l'utilisation de la méthode impossible.

L'intérieur de l'ensemble D défini par les g_i est le suivant :

$$D_I = \left\{ x \mid g_i < 0, \forall i \in \{1, \dots, m\} \right\}.$$

La fonction de barrière, notée $B(x)$, est ajoutée au coût $f(x)$; elle est continue sur D_I et sa valeur tend vers l'infini lorsque la frontière de D est approchée par l'intérieur, c'est-à-dire lorsque l'un des $g_i(x)$ approche zéro par les valeurs négatives. Une itération de la méthode consiste ensuite minimiser la fonction $f(x) + \lambda B(x)$ (où λ est un paramètre réel strictement positif) à l'aide d'algorithme de minimisation directe, une fonction $B(x)$ et un λ convenablement choisis assurant que cette minimisation ne puisse nous mener à des points situés hors de D_I . La suite du processus consiste à réduire progressivement λ afin de diminuer la pénalité et autoriser les algorithmes de minimisation directe à se rapprocher peu à peu de la frontière de D .

Les fonctions de barrière les plus répandues sont les suivantes :

La logarithmique : $B(x) = -\sum_{i=1}^m \ln(-g_i(x))$.

L'inverse : $B(x) = -\sum_{i=1}^m \frac{1}{g_i(x)}$.

Il est important de noter que, si tous les g_i sont convexes, ces deux fonctions de barrière le sont également.

Remarquons qu'une nécessité, pour pouvoir appliquer une telle méthode, est de disposer d'un point initial situé à l'intérieur de D . La méthode de barrière est définie en introduisant la séquence de paramètres $\{\lambda^k\}, k = 0, 1, \dots$ avec $0 < \lambda^{k+1} < \lambda^k$ et $\lambda^k \rightarrow 0$ lorsque $k \rightarrow \infty$.

Une itération de celle-ci consiste à déterminer $x^k = \arg \min_{x \in D_I} \{f(x) + \lambda^k B(x)\}$.

Le fait que $B(x)$ ne soit défini que dans l'intérieur D_I et tende vers l'infini au fur et à mesure que l'on se rapproche des bords de D assure que, même avec un algorithme de minimisation directe, le point obtenu à chaque itération appartient lui aussi à D_I . Le fait que λ^k tende vers zéro implique que le terme $f(x) + \lambda^k B(x)$ tende vers $f(x)$ lorsque k tend vers l'infini.

Si l'on admet que la définition de x^k est celle énoncée ci-dessus, il est possible de démontrer le théorème suivant :

Théorème 4.12. [22] Tout point limite d'une séquence $\{x^k\}$ générée par une méthode de barrière est un minimum global du problème contraint original.

Algorithme de pénalité intérieur [24]

1- Initialisation ;

Fixer $\varepsilon > 0$; trouver un point strictement admissible x^0 .

Choisir $\lambda^1 > 0, (\lambda^1 \simeq 10)$, et $\eta < 0$.

2- Itération; $k=1,2,\dots$

Trouver $x^k = \arg \min_{x \in D_I} \{f(x) + \lambda^k B(x)\}$.

3- Si $\|x^{k+1} - x^k\| < \varepsilon$ stop x^{k+1} est optimal

Sinon poser $k = k + 1$, $\lambda^{k+1} = \eta \lambda^k$ et aller en 2

4.2.3.3. Les méthodes de pénalité extérieure

Les méthodes de pénalité extérieures cherchent à approcher le minimum depuis l'extérieur de D . Au coût $f(x)$ une fonction de pénalité extérieure $p(x)$ dont la valeur est égale à zéro si x est admissible et supérieure à zéro s'il ne l'est pas. L'ajout de cette fonction a pour seul but de pénaliser le coût en cas de violation d'une ou de plusieurs contraintes.

La forme la plus classique des fonctions de pénalités est : $p(x) = \frac{1}{2} \sum_{i=1}^m \left[\max\{0, g_i(x)\} \right]^2$

Et nous introduisons un paramètre $\alpha > 0$ qui permet d'amplifier ou de diminuer sa valeur et, à chaque itération, le sous-problème à résoudre à l'aide d'une méthode de minimisation directe sera de la forme suivante :

$$\begin{aligned} & \min f(x) + \alpha p(x) \\ & \text{avec } x \in \mathbb{R}^n \end{aligned}$$

Lorsque α est grand, une plus grande importance est attachée à l'admissibilité (au sens du problème constraint original), ce qui suggère que la méthode doit commencer avec un α qui n'ait pas une valeur trop élevée (afin d'éviter une terminaison prématurée en un point certes admissible mais qui pourrait se trouver loin de l'optimum), puis augmenter celui-ci progressivement pour s'approcher de l'ensemble admissible du problème original. Ainsi la séquence $\{\alpha^k\}$ utilisée devra être telle que $0 < \alpha^k < \alpha^{k+1}$ et $\alpha^k \rightarrow \infty$ lorsque $k \rightarrow \infty$. Un

paramètre additionnel $\beta > 0$ est utilisé pour générer $\alpha^{k+1} = \beta \alpha^k$. A chaque itération, la méthode obtient donc x^k de la manière suivante :

$$x^k = \arg \min_{x \in D_I} \{f(x) + \alpha^k p(x)\}.$$

Proposition [22],[24] : la suite $\{x^k\}$ converge vers la solution optimale du problème initial

Algorithme de pénalité extérieur [24]

4- Initialisation ;

Fixer $\varepsilon > 0$; trouver un point non admissible x^0 .

Choisir $\alpha^1 > 0, (\alpha^1 \approx 1)$, et $\beta > 1$ (*fixe*).

5- Itération; k=1,2,...

Trouver $x^k = \arg \min_{x \in D \subset \mathbb{R}^n} \{f(x) + \alpha^k p(x)\}$.

6- Si $\|x^{k+1} - x^k\| < \varepsilon$ stop x^{k+1} est optimal

Sinon poser $k = k + 1$, $\alpha^{k+1} = \beta \alpha^k$ et aller en 2

4.2.3.4. Un simple exemple d'application

$$\begin{cases} \min(x_1 - 3)^2 + (x_2 - 2)^2 \\ x_1 + x_2 \leq 1 \\ x_1, x_2 \in \mathbb{R} \end{cases}$$

1- Par la méthode de KKT

$$L(x, \lambda) = (x_1 - 3)^2 + (x_2 - 2)^2 + \lambda(x_1 + x_2 - 1)$$

$$\begin{cases} \nabla f(x) + \lambda \nabla g(x) = 0 \\ \lambda g(x) = 0 \end{cases} \Leftrightarrow \begin{cases} 2(x_1 - 3) + \lambda = 0 \\ 2(x_2 - 2) + \lambda = 0 \\ \lambda(x_1 + x_2 - 1) = 0 \end{cases}$$

Cas 1 : $\lambda = 0$, on trouve $X^1 = (3, 2)$, n'est pas admissible

Cas 2 : $\lambda \neq 0$, on trouve $X^2 = (1, 0)$, admissible

Dans le cas convexe, on a la C.N.S, donc $X^2 = (1, 0)$ est optimale

2- Par la méthode de pénalité extérieure

Soit $X^0 = (1, 1)$, $\alpha^1 = 0.5$, $\eta = 2$

$$p(x, \alpha) = f(x) + \alpha \sum_{i=1}^m \left(\max\{0, g_i(x)\} \right)^2 = (x_1 - 3)^2 + (x_2 - 2)^2 + 0.5(x_1 + x_2 - 1)^2$$

On cherche $x^k = \arg \min_{x \in D \subset \mathbb{R}^n} \{f(x) + \alpha^k p(x)\}$.

Pour k=1

CNS (cas convexe) : $\nabla p(x, \alpha) = 0 \Leftrightarrow \begin{cases} 2(x_1 - 3) + (x_1 + x_2 - 1) = 0 \\ 2(x_2 - 2) + (x_1 + x_2 - 1) = 0 \end{cases}$

On trouve $= X^1 = (2, 1)$, $\|X^1 - X^0\| > \varepsilon$

A la fin, pour k=4 : On trouve la solution optimale $X^4 = (1, 0)$,

3- Par la méthode de barrière

$$B(x, \lambda) = (x_1 - 3)^2 + (x_2 - 2)^2 + \lambda \left(-\frac{1}{x_1 + x_2 - 1} \right)$$

C.N.S (cas convexe) : $\nabla B(x, \lambda) = 0 \Leftrightarrow \begin{cases} 2(x_1 - 3) + \frac{\lambda}{(x_1 + x_2 - 1)^2} = 0 & (*) \\ 2(x_2 - 2) + \frac{\lambda}{(x_1 + x_2 - 1)^2} = 0 & (** \end{cases}$

On a clairement $x_1 = x_2 + 1$, on remplace dans $(**)$, on doit avoir : $8x_2^2(x_2 - 2) = -\lambda$

$$\lambda \rightarrow 0 \Rightarrow 8x_2^2(x_2 - 2) = 0 \Leftrightarrow \begin{cases} x_2 = 0 \Rightarrow x_1 = 1 \Rightarrow X^1 = (1, 0) \\ x_2 = 2 \Rightarrow x_1 = 3 \Rightarrow X^2 = (3, 2) \end{cases}$$

X^1 est admissible et X^2 n'est pas admissible, donc X^1 est optimale

4.3. Problème de VRP : Théories de résolution du problème 2.2

La recherche sur le transport a beaucoup évolué au cours des dernières années. Les entreprises ont découvert qu'une meilleure planification des tournées permettait d'épargner des coûts importants. Les problèmes étudiés sont variés. Ce chapitre mettra en exergue quelques aspects concernant les problèmes de tournées de véhicules ainsi que leurs principales généralisations.

Avant de commencer la présentation de problème de V.R.P, il est intéressant d'apporter au lecteur quelques éléments et définitions sur la théorie des graphes et la programmation mathématique en nombres entiers.

4.3.1. Aperçu sur la Programmation linéaire en nombres entiers :

Un problème de programmation linéaire en nombres entiers (PLNE) est un programme linéaire, sous des contraintes linéaires, dans lequel il y a la contrainte supplémentaire que les variables sont entières.

4.3.1.1. Formulation mathématique d'un programme linéaire en nombres entiers :

$$(PLNE) \left\{ \begin{array}{l} \min Z = \sum_{j=1}^n c_j x_j \\ s.c \quad \sum_{j=1}^n a_{ij} x_{ij} \geq b_i \quad i = 1, \dots, m \\ \quad x_j \geq 0 \quad j = 1, \dots, n \\ \quad x_j \text{ entier} \quad j = 1, \dots, n \end{array} \right.$$

Où A est une matrice m x n et ses coefficients a_{ij} sont entiers, C est un vecteur de longueur n et les c_i sont entiers, b est un vecteur de longueur m et les b_i sont entiers. Dans le cas particulier où les variables $x_j \in \mathbb{N}$ sont remplacées par $x_j \in \{0,1\}$ on dit qu'on a un «*Programme Linéaire en 0, 1*» ou «*Programme linéaire en variables bivalentes*» [25],[26].

Remarques 4.8

- ❖ On considère que le polytope $\Omega = \{X / X \in \mathbb{R}^n, A.X \geq b, X \geq 0\}$ est borné (et non vide), ce qui implique que chaque x_j est borné.
- ❖ Comme x_j est en plus à valeur entière, on en déduit que x_j ne peut prendre qu'un nombre fini de valeurs.

4.3.1.2. Les méthodes de séparation et évaluation (Branch-and-Bound)

4.3.1.2.1. Principes généraux

Les méthodes de séparation et évaluation (les méthodes arborescentes) sont des méthodes exactes d'optimisation qui pratiquent une énumération intelligente de l'espace des solutions. Il s'agit en quelque sorte d'énumérations complètes améliorées. Elles partagent l'espace des solutions en sous-ensembles de plus en plus petits, la plupart étant éliminés par des calculs de bornes avant d'être construits explicitement. D'où le nom synonyme de **méthodes d'énumération implicite**.

Soit un problème d'optimisation combinatoire (POC) défini par un ensemble S de solutions et une fonction économique f de S dans \mathbb{R} . Comme pour les heuristiques, on peut inventer pour un tel problème plusieurs méthodes arborescentes. Toutes auront cependant trois composantes communes [33] :

- une **règle de séparation** (de partitionnement) des solutions ;
- une **fonction d'évaluation** (de bornage) des solutions ;

- une stratégie d'exploration.

Exemple numérique : Problème de sac à dos

Rappelons que ce problème consiste à calculer un sous-ensemble de valeur maximale parmi n objets, sous une contrainte de poids total limité à b . Les valeurs forment un n -vecteur c , et les poids un n -vecteur a . Si on définit des variables binaires x_i (valant 1 si l'objet i est pris), on peut formuler le problème comme un programme linéaire en 0-1 à une seule contrainte

$$\begin{cases} \max z = \sum_{i=1}^n c_i \cdot x_i \\ \sum_{i=1}^n a_i \cdot x_i \leq b \\ x_i \in \{0,1\} \end{cases}$$

La difficulté vient de l'interdiction de fractionner les objets. Nous allons le résoudre par une méthode arborescente. Considérons la donnée suivante :

$$n = 5, a = (18, 12, 15, 16, 13), c = (27, 9, 30, 16, 6.5), b = 45.$$

Définition des nœuds et règle de séparation

En un nœud S , certains objets vont être pris, d'autres interdits, et d'autres auront un statut non encore déterminé, on dira qu'ils sont libres. On sépare un nœud S en choisissant une variable x_i libre et en posant $x_i = 1$ (l'objet i est pris) ou bien 0 (i est non pris).

Evaluation en un noeud S

Comme on a affaire à un problème de maximisation, l'évaluation d'un noeud S va être une **estimation par excès** (majorant) du coût maximal des solutions contenues dans ce noeud.

L'estimation se base sur le programme linéaire relaxé, c'est-à-dire que les x_i peuvent prendre toute valeur entre 0 et 1. Cette situation correspond au cas idéal où les objets peuvent être fractionnés, elle se résout très facilement. On trie pour cela les objets par *utilité* (valeur relative) décroissante c_i / a_i , comme dans le tableau 4.2.

Rang dans le tri	Indice objet	c_i	a_i	Utilité c_i/a_i
1	3	30	15	2
2	1	27	18	1.5
3	4	16	16	1
4	2	9	12	0.75
5	5	6.5	13	0.5

Tab.4.2. Tri des objets par utilité décroissante

On prend les objets par utilités décroissantes, en fractionnant éventuellement le dernier objet pris, de manière à remplir le sac. Ici, on prend les objets 3 puis 1, pour un coût de 57 et un poids de 33. On ne peut prendre ensuite que $12/16 = 3/4$ de l'objet 4 et on atteint ainsi le poids-limite de 45, pour une valeur totale de 69. Ce calcul nous donne l'évaluation à la racine. La solution maximale du problème si les fractionnements sont interdits coûtera donc au plus 69. En un noeud interne, on résout le problème résiduel avec les objets encore libres et la capacité encore disponible, et on ajoute la valeur trouvée à celle des objets déjà pris.

Choix du noeud à séparer et de la décision de séparation sur ce sommet

La décision de séparation se fait sur l'objet libre d'utilité maximale. L'arborescence obtenue est donnée par la figure 4.2. Les noeuds numérotés par ordre de création contiennent l'évaluation. Remarquons qu'un nœud comme S_6 , qui viole la contrainte de capacité, peut être tué immédiatement. S_9 donne la première solution, de coût 66 : les objets 1, 2 et 3 sont pris, et le sac est plein. C'est la fin de la recherche, car toutes les feuilles en attente (S_1, S_3 et S_7) sont d'évaluation inférieure et peuvent donc être tuées.

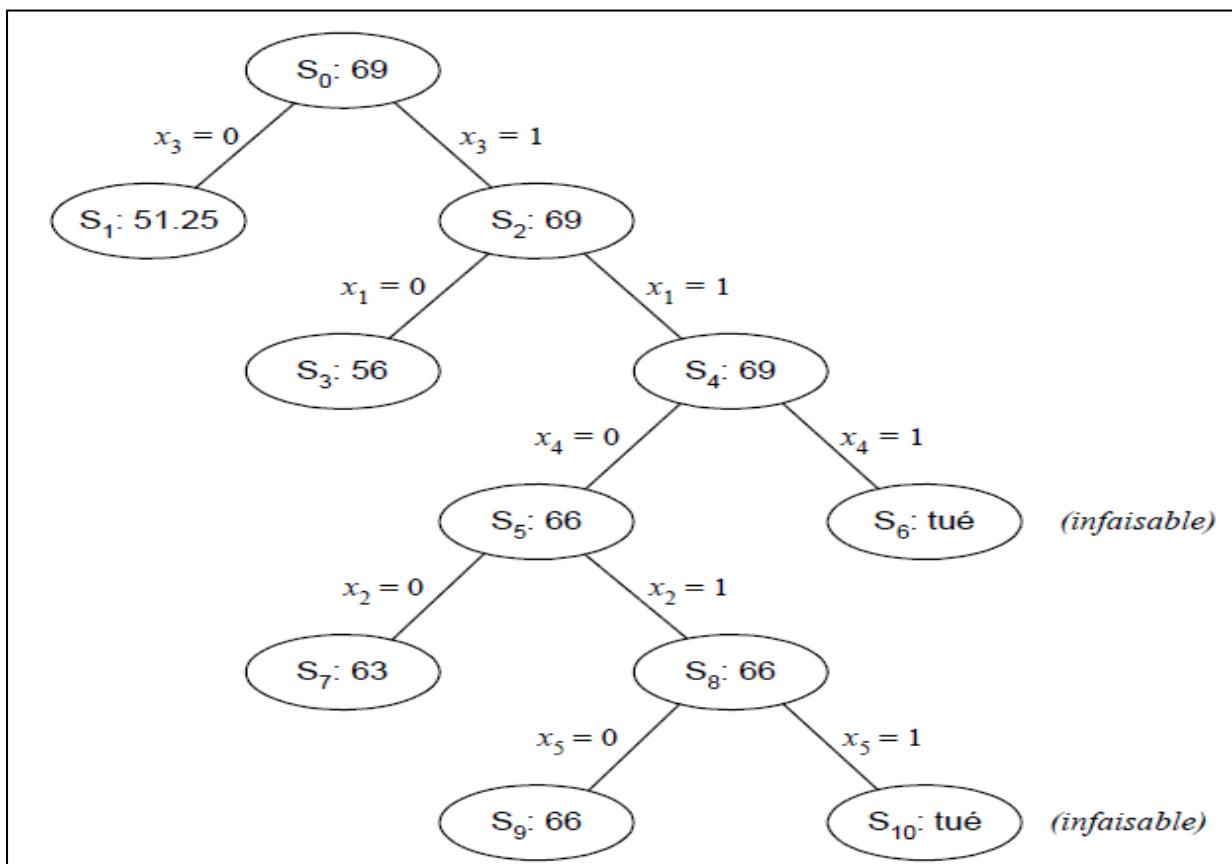


Fig.4.2. Arborescente construite pour l'exemple du Sac à Dos

Stratégies de parcours de l'arborescence

- Dans la stratégie «meilleur d'abord», on se base sur l'idée que la valeur de la borne inférieure est une estimation de l'optimum dans le sous-arbre.
- Dans la stratégie en largeur d'abord, on construit l'arbre niveau par niveau.
- La stratégie en profondeur d'abord fournit très vite (en n étapes) une solution, et utilise peu de mémoire. Mais elle peut s'égarer dans une région peu intéressante de l'arbre.

4.3.2. Rappels sur la théorie des graphes

Avant de commencer la présentation de problème de V.R.P, il est intéressant d'apporter au lecteur quelques éléments et définitions sur la théorie des graphes.

4.3.2.1. Un bref historique de la théorie des graphes

La théorie des graphes est née en 1736 quand Euler démontra qu'il était impossible de traverser chacun des sept ponts de la ville russe de Königsberg (aujourd'hui Kaliningrad) une fois exactement et de revenir au point de départ. Les ponts enjambent les bras de la Pregel qui coulent de part et d'autre de l'île de Kneiphof. Dans la figure suivante, les nœuds représentent les rives [28].

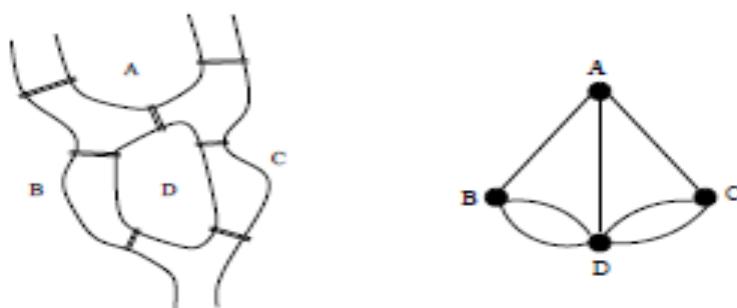


Fig.4.3 Les sept ponts de Königsberg

La théorie des graphes est une tentative de visualisation concrète des faits. C'est par conséquent, un essai de formalisation de situations concrètes de notre vie quotidienne (vie sociale, économique, politique,...) au moyen de dessin permettant d'exprimer commodément le problème posé et parfois même d'en suivre aisément la solution par un algorithme approprié.

4.3.2.2. Définition d'un graphe

Définitions 4.16 :

Un graphe $G = (X, U)$ est déterminé par :

- Un ensemble $X = \{x_1, x_2, \dots, x_n\}$ dont les éléments sont appelés sommets ou noeuds (ce dernier terme est plutôt dans le contexte des réseaux) ;
- Un ensemble $U = \{u_1, u_2, \dots, u_m\}$ du produit cartésien $X \times X$ dont les éléments sont appelés arcs.
- Pour un arc $u = (x_i, x_j)$, x_i est l'extrémité initiale, x_j l'extrémité finale (ou bien origine et destination). L'arc u part de x_i et arrive à x_j .
- Un arc (x_i, x_i) est appelé une boucle.
- Si les lignes de U sont orientées, on les appelle alors des arcs et G prend le nom de "graphe orienté" (Fig.4.5). Par contre, si elles ne sont pas orientées, on obtient alors des arrêtes et G devient "un graphe non orienté" (Fig.4.6).

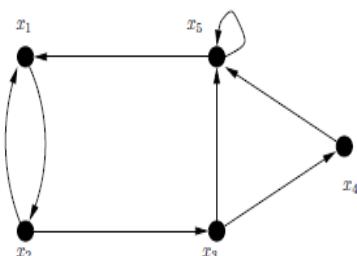


Fig.4.5. Graphe orienté

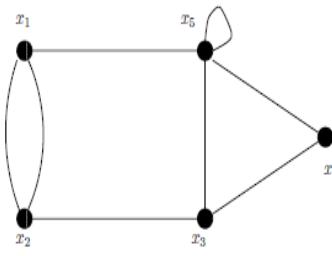


Fig.4.6. Graphe non orienté

A tout graphe orienté, on peut associer un graphe non orienté, supprimant l'orientation des arcs. De même, à tout graphe non orienté, on peut associer un graphe orienté en remplaçant chacune arrête par deux arcs en sens inverses.

Remarque 4.9 : Dans le cas non-orienté, au lieu de noter $G = (X, U)$ et $u = (x_i, x_j)$, on préfère souvent $G = (X, E)$ et $e = (x_i, x_j)$.

Définition 4.17 :

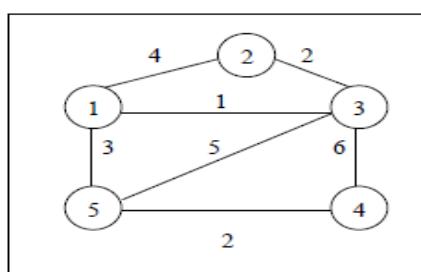
- Un multigraphie $G = (X, E)$ est un graphe pour lequel il peut exister plusieurs arêtes entre deux sommets.

- Un multigraphé $G = (X, E)$ sera appelé un graphe simple si l'on a les deux conditions :
 - 1- Entre deux sommets, il n'y a jamais plus d'une arrête ;
 - 2- Il n'a pas de boucles.

4.3.2.3. Représentation d'un graphe

Un graphe peut être représenté par plusieurs structures, les plus utilisées sont, la structure matricielle et la structure de liste linéaire chaînée [29].

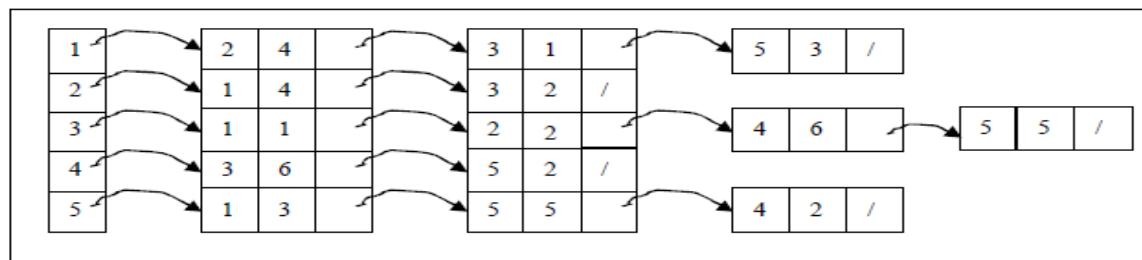
Exemple :



Représentation graphique

	1	2	3	4	5
1	0	4	1	0	3
2	4	0	2	0	0
3	1	2	0	6	5
4	0	0	6	0	2
5	3	0	5	2	0

Représentation matricielle



Représentation chaînée

4.3.2.4. Quelques concepts fondamentaux

4.3.2.4. 1. Principales définitions

- Deux sommets sont adjacents (ou voisins) s'ils sont joints par un arc.
- Deux arcs sont adjacents s'ils ont au moins une extrémité commune.
- L'ensemble des successeurs d'un sommet x est noté $\Gamma^+(x)$ et l'ensemble des ses prédecesseurs est noté $\Gamma^-(x)$

- On appelle degré sortant ou demi-degré extérieur d'un sommet x le nombre d'arcs de la forme $a = (x, y)$ avec $y \neq x$, c'est-a-dire le nombre d'éléments de $\Gamma^+(x) / \{x\}$.

On note $d_s(x)$ de ce degré.

- On appelle degré entrant ou demi-degré intérieur d'un sommet x le nombre d'arcs de la forme $a = (y, x)$ avec $y \neq x$, c'est-a-dire le nombre d'éléments de $\Gamma^-(x) / \{x\}$.

On note $d_e(x)$ de ce degré.

On appelle degré de x la somme du degré entrant et du degré sortant.

Chaine (chemin), cycle (circuit) :

- Dans un graphe simple $G = (X, U)$, on appelle une **chaîne** C une séquence finie de sommets, x_1, x_2, \dots, x_n , telle que $\forall 1 < i < n, (x_i, x_{i+1}) \in U$. On écrit $C = [x_1, x_2, \dots, x_n]$. L'entier n s'appelle la longueur de la chaîne et on écrit $n = L(C)$. Les sommets x_1 et x_n sont appelés les extrémités respectivement initiale et finale de C .
- On appelle un **chemin**, toute chaîne dont les arcs sont orientés dans le même sens.
- On appelle **cycle** de G toute chaîne fermée de G dont l'extrémité initiale coïncide avec l'extrémité finale.
- On appelle **circuit** de G tout chemin fermé de G .
- Une chaîne (chemin, cycle, circuit) est dite **élémentaire** si lors de son parcourt, on ne rencontre pas deux fois le même sommet.
- Alternativement, une chaîne est dite élémentaire si chaque sommet de cette chaîne est adjacent à deux arcs au plus de cette chaîne.

4.3.2.4.2. Connexité dans un graphe :

- Un graphe est dit connexe si pour chaque couple de sommets x et y de G , il existe une chaîne les reliant.
- On dit qu'un graphe est fortement connexe si entre chaque couple de sommets x et y , il existe un chemin de x à y et un autre chemin de y à x . Autrement dit, il passe entre chaque couple de sommets de G , au moins un circuit

4.3.2.4.3. Arbre et arborescence :

- **Arbre** : On appelle arbre tout graphe connexe qui ne possède pas de cycle
- **Racine** : Soit $G = (X, U)$ un graphe connexe et a un sommet de X . On dira que le sommet a est une racine de G si :
$$\forall x \in X ; \exists \text{ un chemin reliant le sommet } a \text{ au sommet } x. x \neq a.$$
- On appelle **arborescence** un arbre menu d'une racine

4.3.2.4. 4. Parcours eulériens et hamiltoniens

Parcours eulériens

Soit $G = (X; U)$ un graphe connexe d'ordre n

Un parcours eulérien d'un graphe G est un parcours passant une et une seule fois sur chaque arête de G .

Quand le parcours est fermé, on parle de tour eulérien.

Théorème 4.13 [30] Un graphe connexe admet un parcours eulérien s'il possède 0 ou 2 sommets de degré impair.

Théorème 4.14 (Euler 1736)[30] Un graphe connexe est eulérienssi chacun de ses sommets a un degré pair.

Parcours hamiltoniens

On appelle parcours hamiltonien un parcours passant une et une seule fois sur chaque sommet du graphe G .

Un parcours hamiltonien fermé ne passant qu'une fois sur chaque sommet est appelé cycle hamiltonien.

Théorème 4.15 (Dirac 1962) [30] Tout graphe G d'ordre $n \geq 3$ et de degré minimal

$$\delta(G) \geq \frac{1}{2} n$$
 possède un cycle hamiltonien.

4.3.3. Problème de voyageur de commerce (PVC)

4.3.3.1. Définition

Le problème du voyageur de commerce, mieux connu sous le nom de Traveling Salesman Problem (TSP), est un des problèmes les plus étudiés en recherche opérationnelle. Effectivement, il est à la base de tous les problèmes de tournées et a une multitude d'applications réelles. Ce problème consiste à trouver le chemin le plus court qui visite toutes les villes une et une seule fois tout en revenant au point de départ. Le problème du voyageur de commerce étant NP-Dur, le temps pour trouver la solution optimale augmente de façon exponentielle en fonction de la taille du problème [31]. Il se définit mathématiquement comme ceci: Soit $G = (N, A)$ où G est un graphe, $N = \{v_0, v_1, \dots, v_n\}$ représente l'ensemble des nœuds (villes). Par ailleurs, $A = \{(v_i, v_j) \mid v_i, v_j \in N, i \leq j\}$ représente les arrêtes reliant ces nœuds si les distances sont symétriques et $A = \{(v_i, v_j) \mid v_i, v_j \in N, i \neq j\}$ représente les arcs lorsqu'elles sont asymétriques. Dans le cas où le problème est symétrique c'est-à-dire où le graphe est non orienté, on parlera d'un cycle et dans le cas contraire on parlera d'un circuit.

4.3.3.2. Formulation mathématique

Définissons les différentes variables nécessaires pour effectuer la formulation mathématique du problème du voyageur de commerce.

- d_{ij} = Distance entre les villes i et j ;
- n = Nombre de villes j ;
- X_{ij} = Variable binaire qui prend la valeur 1 si la ville i est visitée immédiatement avant la ville j . Sinon, cette variable prend la valeur 0.

Le problème consiste à minimiser la longueur du **cycle Hamiltonien**.

$$\left\{ \begin{array}{l} \text{Min } z = \sum_{i=1}^n \sum_{j=1}^n d_{ij} X_{ij} \\ \text{les contraintes sont les suivantes :} \end{array} \right. \quad (1)$$

$$\sum_{j=1}^n X_{ij} = 1 \quad i = 1, \dots, n \quad (2)$$

$$\sum_{i=1}^n X_{ij} = 1 \quad j = 1, \dots, n \quad (3)$$

$$\sum_{i \in S} \sum_{j \in S} X_{ij} \leq |S| - 1 \quad \text{pour toute } S \in \mathcal{N} \quad (4)$$

La contrainte (2) s'assure qu'on ne sort qu'une seule fois de chacun des points tandis que la contrainte (3) vérifie que l'on entre seulement une fois à chaque point. Ces contraintes permettent donc de visiter tous les points une et une seule fois. Par contre, si l'on n'ajoute pas la contrainte (4) des sous tours se formeront. La quatrième contrainte élimine donc tous les sous-tours possibles. Dans cette contrainte $|S|$ dénote la cardinalité de l'ensemble S . Cette contrainte doit être décrite pour tous les sous-ensembles S de \mathcal{N} .

4.3.4. Le problème de tournées de véhicules (VRP) et ces extensions :

Le problème de tournées de véhicules et ses généralisations a été largement étudié au cours des dernières années. Le premier papier traitant du problème de tournée de véhicules a été publié vers la fin des années 1950 par Dantzig et Ramser (1959). Ce problème, plus souvent appelé Vehicle Routing Problem (VRP) a ensuite attiré un grand nombre de chercheurs car il est théoriquement très intéressant. De plus, les applications du VRP sont nombreuses. Ainsi, la plupart des entreprises qui doivent livrer un produit à plusieurs clients font face à ce problème. La littérature du problème VRP est par conséquent très volumineuse.

4.3.4.1. VRP de base :

Le problème de tournées de véhicules peut être défini comme un problème où de nombreux clients doivent être desservis à partir d'un unique dépôt avec des demandes connues mathématiquement. Le VRP se définit sur un graphe $G = (V, A)$ où $V = \{0, \dots, n\}$ représentent l'ensemble des sommets, c'est -à - dire l'ensemble des clients à visiter et $A = \{(i, j) \mid i, j \in V \text{ et } i \neq j\}$ ensemble des arcs représentant l'ensemble des chemins possibles.

Le point « 0 » représente le dépôt qui est le point de départ et d'arrivée de toutes les tournées.

Les véhicules sont indiqués par $k=1,\dots,m$. Un client « i » a une demande q_i . Le coût de transport entre les clients i, j est de C_{ij} . la capacité du véhicule k est Q_k [32].

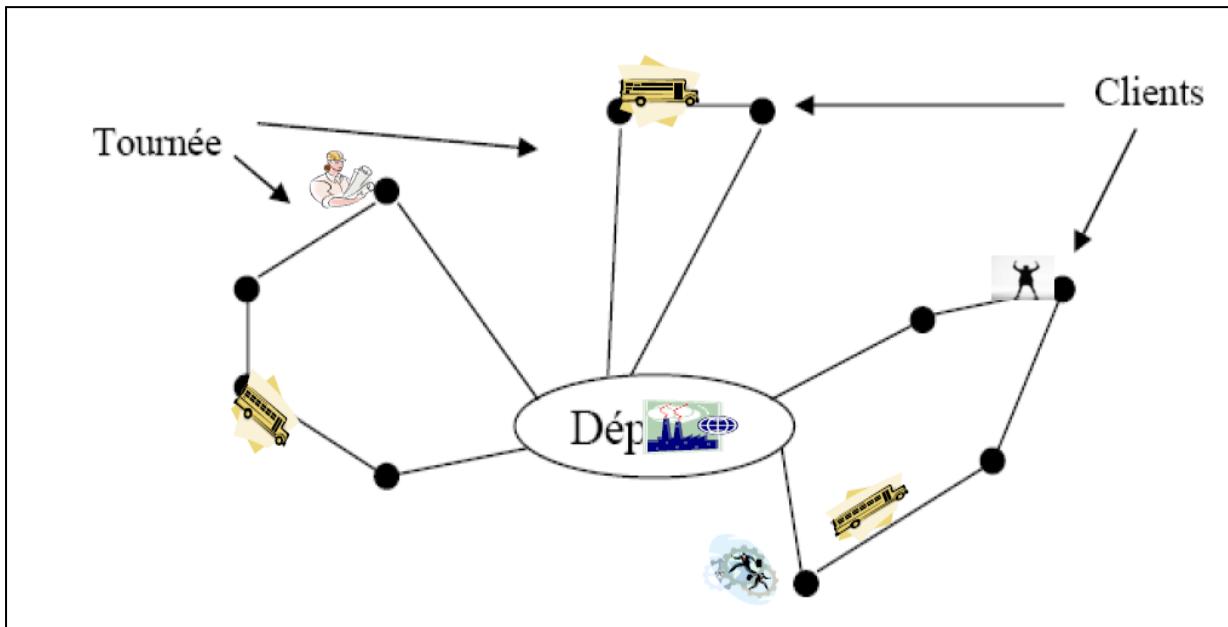


Fig.4.7 Schéma d'une solution du VRP de base

Le VRP de base consiste à affecter un itinéraire à chaque véhicule, démarrant et finissant au dépôt, de sorte que les demandes de tous les clients soient satisfaites, le coût total de transport devant être réduit au minimum. La forme de la solution d'un VRP est représentée dans la figure (Fig.4.7).

Formulation du VRP de base

Il existe de nombreuses formulations du problème de tournées de véhicules [39]. La formulation suivante est tirée de Fischer et Jaickumur [36]. Définissons tout d'abord l'ensemble des variables nécessaires pour effectuer la formulation mathématique.

❖ Paramètre :

m : Nombre de Camions (Véhicules) disponibles ; n : Nombre de clients à visiter.

Les clients sont numérotés de 1 à n et l'entrepôt a le numéro 0.

Q_k : Capacité du camion k ; q_i : Demande du client i .

C_{ij} : Le coût de transport entre les clients i, j

❖ Variables de décision :

$$X_{ijk} = \begin{cases} 1 & \text{si le camion } k \text{ visite le client } j \text{ immédiatement après le client } i \\ 0 & \text{Sinon} \end{cases}$$

$$Y_{ik} = \begin{cases} 1 & \text{si le client } i \text{ visité par le camion } k \\ 0 & \text{Sinon} \end{cases}$$

Le VRP de base doit alors réduire au minimum.

$$\left\{ \begin{array}{l} \text{Min } Z = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^m C_{ijk} X_{ijk} \end{array} \right. \quad (1)$$

$$\left\{ \begin{array}{l} \sum_{k=1}^m Y_{ik} = \begin{cases} 1 & i = 2..n \\ m & i = 1 \end{cases} \end{array} \right. \quad (2)$$

$$\left\{ \begin{array}{l} \sum_{i=1}^n q_i Y_{ik} \leq Q_k \quad k = 1..m \end{array} \right. \quad (3)$$

$$\left\{ \begin{array}{l} \sum_{i=1}^n X_{ijk} = Y_{jk} \quad j = 1..n, k = 1..m \end{array} \right. \quad (4)$$

$$\left\{ \begin{array}{l} \sum_{j=1}^n X_{ijk} = Y_{ik} \quad i = 1..n, k = 1..m \end{array} \right. \quad (5)$$

$$\left\{ \begin{array}{l} Y_{ik} \in \{0,1\} \quad i = 1..n, k = 1..m \end{array} \right. \quad (6)$$

$$\left\{ \begin{array}{l} X_{ijk} \in \{0,1\} \quad i = 1..n, j = 1..n, k = 1..m \end{array} \right. \quad (7)$$

- La contrainte (2) assure que chaque client est approvisionné par un seul véhicule (excepté le dépôt qui est visité par tous les véhicules).
- La contrainte (3) permet de respecter les capacités des véhicules.
- La contrainte (4) et (5) vérifie qu'un véhicule qui visite un client doit le quitter.

Extensions du VRP de base :

Le VRP ignore un grand nombre de contraintes rencontrées dans des problèmes concrets.

Certaines de ces contraintes et extensions sont [34] :

- (i) Chaque véhicule peut être affecté à plus d'un itinéraire dans la mesure où son temps de travail T le permet ; si le temps nécessaire au parcours de ces itinéraires est inférieur à T . Notons qu'une telle contrainte, ainsi que beaucoup de celles qui nécessitent la connaissance des temps de parcours (t_{ij}) entre chaque paire de clients peuvent compliquer le modèle.
- (ii) Chaque client doit être visité une seule fois, pendant des intervalles de temps spécifiés durant la période donnée.
- (iii) Outre les conditions de la contrainte (ii) ci-dessus, les véhicules peuvent également avoir des fenêtres de temps de travail pendant la période (time Windows) [35]. Chaque véhicule ne fonctionne que durant les fenêtres de temps indiquées.

4.3.4.2. Quelques aspects pratiques du VRP :

❖ Composition de la flotte :

La composition de la flotte : le problème général de tournées suppose que les véhicules soient homogènes [34]. Pourtant, il est rare de trouver un tel cas en pratique, car il existe souvent des distinctions entre les véhicules, donc le coût de transport, la rapidité de la livraison, la capacité de chargement varient de l'un à l'autre.

❖ Produits multiples :

Dans certains VRP, les véhicules sont compartimentés de telle sorte que différents produits puissent y être stockés séparément, chaque client peut avoir besoin de quantités de chaque type de produit. De tels problèmes se présentent dans la distribution des produits alimentaires, carburants..., etc.

❖ Fenêtre de temps :

Le problème de tournée de véhicules devient plus complexe si on ajoute des fenêtres de temps [34]. L'ajout de la contrainte de fenêtre de temps pour un client qui sait ainsi à quelle heure sa demande lui sera livrer et qui pourra donc prévoir cette réception dans la planification des tâches prévue à l'horaire de la journée.

❖ Multi Dépôts :

Dans certains VRP, les dépôts fonctionnent de manière dépendante et les véhicules partant d'un dépôt peuvent, après avoir livré des clients, retourner à un autre dépôt, éventuellement pour un autre chargement et assurer la livraison d'un autre client, dans ce cas, les dépôts ne peuvent pas être traités individuellement [37].

❖ Différents objectifs :

Dans le problème de tournées de véhicules contrairement au problème de voyageur de commerce l'objectif n'est pas seulement de minimiser la distance totale parcourue, mais peut être aussi :

1. Minimiser le nombre de véhicules utilisés.
2. Minimiser le nombre de clients non desservis.
3. Minimiser le temps de parcours.

4.3.4.3. Présentation de quelques méthodes de résolution du VRP de base :

4.3.4.3.1. Les méthodes exactes :

Les méthodes exactes se basent sur une stratégie qui consiste à chercher toutes les configurations possibles, évaluer leurs coûts et retenir le minimum. Parmi les algorithmes exacts, il y'a :

- Les méthodes par séparation et évaluation, appelées aussi méthodes arborescentes ou *BRANCH AND BOUND*
- La programmation dynamique.

La complexité de l'algorithme Branch-and-Bound

L'algorithme de séparation et évaluation (Branch-and-Bound) est essentiellement une méthode d'exploration exhaustive, permettant d'examiner toutes les solutions possibles même dans les cas les plus défavorables.

Considérons un graphe orienté, symétrique et ayant n sommets, associé aux arêtes et carrefours d'une ville. Dans ce cas, il y'aura éventuellement $(n-1)$ cycles Hamiltoniens distincts ; la complexité de la méthode de séparation et d'évaluation ne dépassera pas le seuil de $O(n !)$ même dans les situations les plus critiques.

La difficulté de résolution du VRP (problème NP-complet) avec des méthodes exactes à inciter les chercheurs à mettre en oeuvre des méthodes heuristiques (approchées) simples et rapides

4.3.4.3.2. Les méthodes approchées

Les approches heuristiques se répartissent en deux classes principales :

- Procédures de construction des tournées
- Procédures d'affinage des tournées.

1-Procédure de construction des tournées

On définit un graphe $G = (X, U)$, où X est l'ensemble des clients et U est un arc du graphe, c'est-à-dire $(i, j) \in U$, s'il représente le plus court chemin reliant directement le client i au client j .

Les plus connus de ces algorithmes sont :

Méthode du plus proche voisin

Principe

Le principe de cette méthode est de choisir arbitrairement un sommet, ce dernier c'est le premier sommet d'une chaîne Hamiltonienne qu'on va construire. Le sommet suivant est le sommet le plus proche du dernier choisi, qui n'appartient pas déjà à la chaîne. Joindre le sommet d'extrémité finale de la chaîne à celui d'extrémité initiale [41], [42].

Exemple :

Soit le problème de voyageur de commerce (PVC) présenté sous la forme matricielle suivante

	1	2	3	4	5
1	-	5	8	4	2
2	5	-	8	3	3
3	8	8	-	5	2
4	4	3	5	-	3
5	2	3	2	3	-

Matrice des coûts

Soit par exemple on démarre à partir du sommet 1. On applique la méthode du plus proche voisin, on aura :

- ✚ Le plus proche voisin de 1 est 5. 1----5 de valeur 2.
- ✚ Le plus proche voisin de 5 est 3. 1----5---3 de valeur 4 (coût de la chaîne {1, 5, 3}).

- ⊕ Le plus proche voisin de 3 est 4. 1---5---3---4 de valeur 9.
- ⊕ Le plus proche voisin de 4 est 2. 1---5---3 ---4---2 de valeur 12.
- ⊕ Tous les sommets sont parcourus, on va joindre le sommet d'extrémité finale 2 à celui d'extrémité initiale 1, on obtient donc la chaîne hamiltonienne suivante : 1--- 5 ---- 3 -- 4 ---- 2 ----1 de valeur de coût égale à 17.

Cette méthode peut être adaptée au problème étudié avec de petites modifications, dont l'algorithme est le suivant :

Algorithme de plus proche voisin adapté au problème de VRP

Début

- a) Choisir un véhicule de la liste des véhicules disponibles ;
- b) Commencer la tournée à partir d'un sommet quelconque non visité (dans notre cas c'est le dépôt).

c) Répéter

- Trouver le sommet le plus proche du dernier sommet de la tournée et l'ajouter celle-ci en vérifiant la contrainte de capacité du véhicule et d'autres contraintes spécifiques au problème étudié ;

Jusqu'à aucune liaison ne soit possible

- d) Si La liste des clients non visités n'est pas vide alors

 Si La liste des véhicules est non épuisée alors

 Aller en a) .

 Sinon afficher « Manque de véhicules »

 Sinon Fin.

Fin

2-Procédures d'affinage des tournées

Il est d'usage d'essayer d'améliorer une solution réalisable générée pour tout problème d'optimisation combinatoire. Aussi, certaines méthodes d'amélioration des tournées du TSP (problème de voyageur de commerce) ont été élaborées. Puisque une solution réalisable d'un

VRP peut être vue comme un ensemble de tournées (plusieurs TSP), ces procédures peuvent être adaptées au VRP, nous présentons quelques-unes des plus utilisées.

Méthode de 2-Optimalité

La méthode 2-Optimalité consiste à échanger (permuter) deux branches ou arcs dans une tournée, jusqu'à aboutir à une solution que l'on ne peut plus raffiner. Le principe de cette dernière se présente dans l'algorithme suivant : [41], [43]

Algorithm

Début

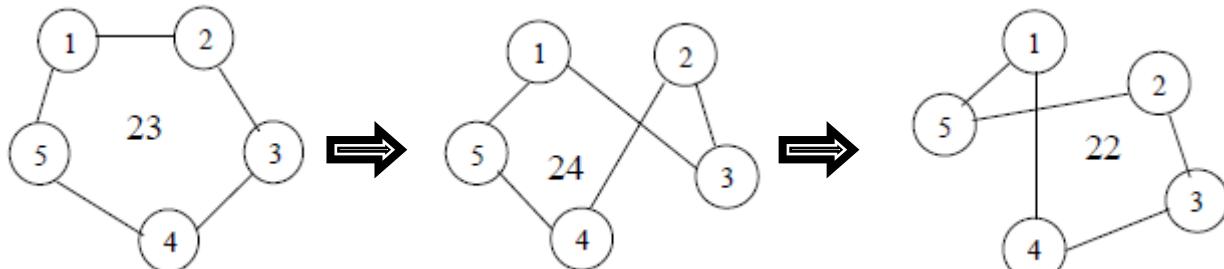
- * Etape 1 : trouver une tournée initiale (généralement de manière aléatoire).
- * Etape 2 : améliorer la tournée en substituant à un ensemble de deux arcs, un autre ensemble de même nombre d'arcs, c'est-à-dire supprimer deux arcs (i, j) et (k, l) non adjacents d'une tournée, puis de reconnecter les deux chaînes obtenues par les deux nouveaux arcs (i, l) et (j, k) .
- * Etape 3 : reprendre l'étape (2) jusqu'à ce qu'il n'y ait plus de changement (la solution est alors dite 2-Optimale).

Fin

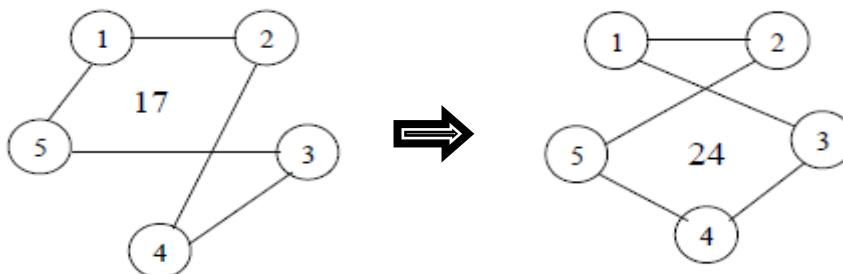
Exemple

Retenant l'exemple du voyageur de commerce précédent, et soit la tournée initiale suivante 1-2-3-4-5-1 :

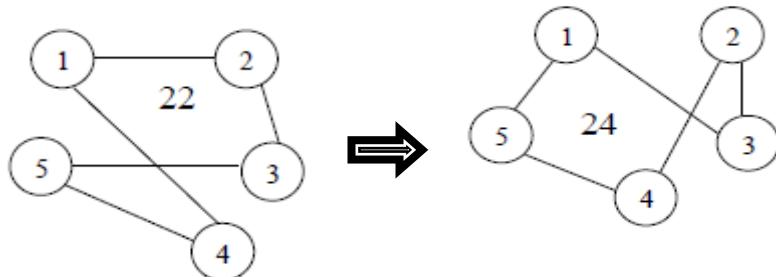
1 : On choisit l'arc $(1,2)$ donc, on le permute avec les arcs $(3,4)$ et $(4,5)$ respectivement. On aura :



2- On choisit l'arc $(2,3)$:



3- On choisit l'arc (3,4) :



Donc le cycle de départ est améliorer à : 1-2-4-3-5-1 de valeur 17. En général pour une tournée donnée comprenant n clients le nombre de combinaisons à considérer est de l'ordre n^2 , puisqu'il y a C_n^2 façons d'enlever deux arcs et deux manières de reconnecter les chaînes.

Algorithme de 2-Optimalité adapté au problème de VRP

Début

- a- Ordonner la liste des véhicules par ordre décroissant des capacités,
 - b- Choisir un véhicule de la liste des véhicules disponibles,
 - c- Appliquer la méthode de 2-Optimalité sur le problème posé,
 - d- Affecter au véhicule choisi les clients suivant la tournée trouvée dans c- en vérifiant la capacité du véhicule,
 - e- Supprimer les clients visités dans la liste des clients,
 - f- Si la liste des clients non visités $\neq \emptyset$ alors
 - Si la liste des véhicules est non épuisée alors
 - Aller en b-,
 - Sinon
 - « Il y a manque de véhicules »
- Finsi
Finsi

Fin

4.4. Conclusion

Dans ce chapitre nous avons détaillé quelques méthodes qui résolvent les deux problèmes proposés à savoir le problème de programmation mathématique non linéaire avec contraintes et problème de tournée de véhicule. La résolution des problèmes qui fera l'objet du chapitre suivant.

Chapitre 5

*Résolution de problème 1 et réalisation
d'un logiciel d'application*

5.1. Introduction

Ce chapitre porte sur la description et la mise en œuvre de deux méthodes heuristiques pour résoudre le problème 1, entre autre choisir deux méthodes adéquates, les adapter au problème, puis comparer les résultats et les performances des deux méthodes.

Lors de la réalisation de cette étude, nous avons été amenés à concevoir un logiciel dans le but d'appliquer les méthodes de résolution au problème posé. En effet il serait déraisonnable d'essayer de trouver une solution au problème sans l'aide d'une machine, étant donné la complexité du problème et la méthode utilisée.

Notre logiciel est une application MDI qui a été conçue avec Borland Delphi 7 Entreprise.

5.2. Choix des méthodes

Notre choix s'est porté sur la méthode de la recherche tabou et la méthode génétique. La résolution peut être envisagée en deux étapes, la première consiste à trouver une solution réalisable de départ pour la méthode recherche tabou, et un ensemble de solutions réalisables pour la méthode génétique. La seconde étape consiste à améliorer la solution obtenue dans la première étape.

5.3. Justifications du choix

Rappelons que notre problème est un problème linéaire en variables bivalentes. Si ce problème pourrait se formuler en un nombre restreint de variables, on aurait pu peut-être utiliser une méthode de séparation et évaluation par exemple, mais vu le nombre important de variables, et compte tenu que ces variables soient bivalentes, on était contraint à se diriger vers les métahéuristiques.

➤ Méthode 1 :

Vu la complexité du problème, et qu'il n'existe pas de méthodes exactes pour la résolution de ce type de problème, nous avons décidé de résoudre notre problème avec une métahéuristique Génétique ;Ce choix est justifié par la facilité d'adaptation des algorithmes génétiques aux différents problèmes, et par le temps de calcul de ces algorithmes qui est acceptable, et aussi de vouloir comparer deux approches fondamentalement différentes, la méthode génétique qui est une méthode évolutive, et la méthode de recherche tabou qui est une méthode séquentielle.

➤ Méthode 2 :

L'application de la méthode " Recherche Tabou " sur des problèmes d'affectations de grande dimension, a généralement abouti à des résultats très satisfaisants, et envisageables dans la pratique si cette méthode est bien exploitée, dans le sens ou l'efficacité de cette méthode générale, est flexible et dépendante particulièrement de la façon de son adaptations au problème, c'est pour cela qu'on a choisi la méthode de la recherche tabou comme méthode de résolution à notre problème.

5.4. Adaptation de la méthode génétique

L'adaptation de la méthode génétique à un problème, consiste d'abord à établir un bon codage de la solution, puis à l'adaptation de tous les opérateurs de cette méthode à ce problème, car en seins de ces opérateurs que l'adaptation sera la plus aigue, et bien sûr l'adaptation de l'algorithme général, en choisissant les probabilités de reproduction, et de mutation.

5.4.1. Codage de la solution et des données

Avant de procéder au codage de la solution, on doit établir un codage précis des données.

➤ Codage des données :

Pour le codage des données, on définit les variables suivantes

1. **NbNavire** : représente le nombre de navires de Sonatrach plus les navires susceptibles d'être affrétés ;
2. **NbPort** : représente le nombre de ports de déchargement ;
3. **NbMois** : représente le nombre de mois de la période d'étude ;
4. **NbMaxRotation** : représente le nombre maximum de rotation que le navire le plus rapide pourra effectuer ;

Les données des navires et des ports de déchargement, seront respectivement stockés dans les enregistrements **CelluleNavire** et **CellulePort**, détaillé dans le tableau suivant :

CelluleNavire=Enregistrement

Num : entier Positif ; *{Numéro du navire}*
Cap : entier Positif ; *{Capacité du navire}*
Vit : entier Positif ; *{Vitesse du navire}*
Deb : entier Positif ; *{Débit de déchargement du navire}*

Fin ;

CellulePort=Enregistrement

Num : entier Positif ; *{Numéro du port}*
CapMin : entier Positif ; *{Capacité d'accueil minimum}*
CapMax : entier Positif ; *{Capacité d'accueil maximum}*
FraisP : entier Positif ; *{Frais portuaires}*
Deb : entier Positif ; *{Débit de déchargement du port}*
DMin : Tableau[1..nbPort,1..nbMois]d'entier Positif ; *{Demandes mensuelles minimum}*
DMax : Tableau[1..nbPort,1..nbMois]d'entier Positif ; *{Demandes mensuelles maximum}*
Fin ;

Tab.5.1. les enregistrements CelluleNavire et CellulePort

➤ **Codage de la solution :**

Afin d'économiser l'espace mémoire, et de faciliter la programmation de ces méthodes de résolutions, on a opté pour le codage suivant.

CelulleRotation =Enregistrement

J :entier positif ; *{Jour de chargement de la rotation}*
D :entier positif ; *{Destination de chargement de la rotation }*
Q :entier positif ; *{Quai de chargement de la rotation}*

Fin ;

Sol =Tableau[1..nbNavire,1..nbRotation] de CelulleRotation ;

Tab.5.2. Codage de la solution (enregistrements CelluleRotation)

Exemple 5.1.

$\text{Sol}[i,r].J$: {Le jour de chargement de la rotation r effectuée par le navire i } ;

$\text{Sol}[i,r].D$: {Le numéro de la destination de la rotation r effectuée par le navire i } ;

$\text{Sol}[i,r].Q$: {Le numéro du quai de chargement de la rotation r effectuée par le navire i } ;

➤ **Explication du codage de la solution :**

La solution du problème sera représentée par une matrice, **Sol**, de taille,

NbNavire×**NbMaxRotation**, entre autre la matrice **Sol** comporte **NbNavire** lignes, qui comporte chacune d'elles **NbMaxRotation** de cellule **CelleleRotation**.

Une cellule **CelluleRotation**, représente les informations relatives à une seule rotation, elle comporte trois entiers positifs qui représentent respectivement, le jour de chargement de la rotation, la destination de la rotation, et le poste de chargement de la rotation.

Notons que le navire qui effectue la rotation n'est pas représenté par un entier, il est en fait donné par la ligne de la cellule **CelluleRotation**, car chaque ligne représente les rotations effectuées par un navire tout au long de l'année.

Enfin, L'ensemble de la population (des solutions) d'une génération sera représenté par le tableau suivant.

TSol = Tableau [1..100] de Sol ;

➤ **Illustration :**

Le codage d'une seule solution est illustré dans l'exemple suivant.

Pour cet exemple le nombre maximum de rotation sera égal à 3, et le nombre de navires sera aussi égal à 3.

	Rotation 1			Rotation 2			Rotation 3		
	Jour	Destination	Quai	Jour	Destination	Quai	Jour	Destination	Quai
Niveau 1	5	6	2	14	2	3	0	0	0
Niveau 2	4	3	5	30	5	5	49	1	5
Niveau 3	0	0	0	0	0	0	0	0	0

Tab.5.3. Exemple de codage d'une seule solution

Dans l'exemple précédent, le navire 1 effectuera :

- Le chargement pour sa première rotation le jour 5 vers le port 6, à partir du poste de chargement 2 ;
- Le chargement pour la deuxième rotation le jour 14 vers le port 2, à partir du poste de chargement 3, et avec cette rotation le navire 1 accomplira son programme. ;

Par contre le navire 3, n'effectuera aucune rotation, ce navire ne sera pas affrété.

5.4.2. Procédure Recherche de solutions réalisables

La procédure suivante génère une solution réalisable pour notre problème :

Procédure Sol-réal(E/S Sol :Solution) ;

Var

i : entier positif ; {indice navire}

j : entier positif ; {indice port}

m : entier positif ; { indice mois}

DM :Tableau[1..nbMois] d'entier positif ;

FM :Tableau[1..nbMois] d'entier positif ;

{Les tableaux précédents représentent les dates de début, et de la fin des mois de la durée d'étude, Exemple : FM[m] : date de la fin du mois[m]}

QT :réel ; {QT est utilisé par l'algorithme pour calculer la somme de la quantité transportée chaque mois vers le client}

JourDisCour[i] : Tableau[1..Nb-Navire] de entier positif ;

{Numéro du jour de disponibilité courant du navire[i]}

RotCour [i] : Tableau[1..NbNavire] de entier positif ;

{Numéro de la Rotation courante du navire[i]}

Début

Pour m :=1 à Nb-Mois **faire** {pour chaque mois faire} ;

Tri-Aléa-Port(TabPort,NbPort) ; {Cette procédure trie aléatoirement le tableau TabPort.

Le tri des ports selon l'ordre décroissant des quantités mensuelles nous donnera une solution plus proche de l'optimum que celle qui sera trouvée par le tri aléatoire, mais on doit introduire l'aléatoire pour garantir de trouver un nombre important de solutions réalisables

différentes en un temps acceptable.}

Pour i :=1 à Nb-Navire **faire** {pour chaque navire faire}

Si jourDis[i] <DM[m] **alors** JourDis[i] :=DM[m] ;

Finsi ;

Fait ;

{Les demandes des mois précédents au mois[m] supposées satisfaites, la boucle précédente affecte aux jours de disponibilité des navires qui pourront être libre(disponible) avant le début du mois en cour ,la date de début de ce dernier, pour garantir que les navires ne fassent pas de rotations avant le début du mois en cour }

Pour j :=1 à Nb-Port **faire** {pour chaque port[j] faire }

Tri-Navire(TabNavire,nb-Navire) ; {Cette procédure trie le tableau TabNavire constitué de CelluleNavire par ordre croissant du rapport (coût de Rotation du navire[i] au port[j]/quantité transportée par le navire[i]au port[j]). De cette manière, on favorise les navires les plus rentables, et on fait un pas vers l'optimum}

I :=1 ; {Prendre le premier navire de la liste ListeNavireAcc}

QT :=0 ; {ce compteur compte la quantité qui peut être transportée, par les rotations affectées jusqu'à la, au port[j] le mois en cour [m]}

Tant que (QT<port[j].DMin[m]) et (i<Nb-Navire-Acc) **faire** {si quantité QT ne satisfait pas la demande minimale mensuelle du port[j] le mois[m], qui est représentée par Port[j].DMin[m], et les navires de la liste ListeNavireAcc ne sont pas tous explorés, alors pour chaque navire[i] de la liste ListeNavireAcc faire}

Si (JourDis[i] <FM[m]) et (QT+Navire[i] .cap <= Port[j].DMax) **alors**

{Si le navire[i] de la Liste ListeNavireAcc est disponible avant la fin du mois en cour, et la quantité QT plus la quantité qui peut être transporté par le navire[i] ne dépasse pas la quantité maximale demandée par le Port[j] le mois en cour [m] alors}

Début Quai :=A_etQuai(TabNavire[i], JourDisNavire[i]) ;

Si quai<>0 **alors** {si un quai compatible est disponible le jour de disponibilité de navire alors}

Début

Sol[i,RotCour[i]] .J := JourDisNavire[i] ;

Sol[i,RotCour [i]] .D :=j ;

Sol[i,RotCour [i]] .Q :=Quai ;

{Affectation d'une rotation au navire TabNavire[i]}

```
Qt := Qt + TabNavire[i].cap ; {mise à jour du compteur QT}
RotCour [i] := RotCour [i]+1 ;
Fin ;
{incrémenter la Cellule rotation en cour}
JourDisNav[i] := JourDisNav[i]+DureeRotation[I,j] ;
{mise à jour de la date de disponibilité du navire i}
Fin ;
Sinon JourDisNav[i] := JourDisNav[i] +1 ;
{si aucun quai compatible n'est disponible le jour JourDisNav[i], alors le navire est considéré comme indisponible ce jour là, donc le jour de disponibilité du navire est incrémenté par un jour}
Finsi ;
Sinon i := i+1 ;
{Si le jour de disponibilité du navire i, dépasse la fin du mois en cours, on passe au navire suivant dans le tableau TabNavire }
Finsi ;
Fait ;
Fait ;
Fin.
```

Algorithme 5.1. Procédure Recherche de solutions réalisables

Remarque 5.1 : On a préféré donner l'algorithme détaillé de la procédure **SolReal**, car cette procédure est très importante dans la résolution, et permet à l'algorithme de résolution de faire un grand pas vers l'optimum.

Remarque 5.2 : L'organigramme de cette procédure est donné par la figure 5.1.

5.4.3. Justification de l'algorithme SolReal

- Le nombre de ports de déchargement est fini. A chaque fois qu'on satisfait la demande d'un port, on passe à un autre port de déchargement suivant l'ordre aléatoire des

demandes. Les ports sont triés aléatoirement à chaque mois pour générer un grand nombre de solutions réalisables différentes. Le processus s'arrêtera dès qu'on parcourt tous les ports. Et à la fin de l'algorithme tous les ports seront satisfaits.

- Le nombre de navires disponibles et compatibles avec un port de déchargement est fini et non nul, car les navires disponibles dans la liste des navires candidats à l'affrètement, qui comporte les navires de Sonatrach plus les navires susceptibles d'être affrétés sont largement suffisants pour satisfaire la demande de tous les ports de déchargement.
- L'algorithme explore les navires compatibles selon l'ordre du tri établi par rapport à chaque port, ce tri se fait par l'ordre croissant du rapport (coût de la rotation du navire i vers le port j) / (la quantité transportée vers ce port par cette rotation), de cette manière, on favorise les navires de meilleur rendement, donc on effectue un pas vers l'optimum.
- Une rotation n'est affectée à un navire que dans le cas où le navire est disponible, plus exactement si il n'effectue aucune rotation, et on dispose d'un quai de chargement le jour de chargement de la rotation. La disponibilité du navire i est exprimée dans l'algorithme par la variable " JourDisNav[i] ".
- La disponibilité d'un quai de chargement un jour donné est vérifiée par la fonction "AffectQuai ", cette fonction renvoie le numéro du quai si ce dernier est disponible, et renvoi zéro si aucun quai n'est disponible. Si aucun quai n'est disponible le jour JourDisNav[i], ce dernière sera incrémenté d'un jour, et le navire sera considéré comme indisponible jusqu'au prochain jour, car la durée de chargement de tous les navires est à peu près de 24h, contrairement aux durées de déchargements qui sont très différentes d'un navire à un autre.
- Les quantités transportées par les rotations affectées le mois en cours, jusqu'à le jour en cours, sont comptées par la variable QT, si la somme de cette quantité avec la quantité éventuellement transportée par le navire en cours de traitement par l'algorithme se situe dans la marge de flexibilité de la demande mensuelle du port, et un quai compatible et disponible, et le navire concerné est aussi disponible, une rotation sera affectée à ce navire, qui aura comme jour de chargement le premier jour de disponibilité du navire, de cette manière on minimise le temps de repos des navires.

- Lorsqu'on affecte une rotation à un navire il faut préciser la destination de la rotation, qui est tout simplement le port en cours de traitement par l'algorithme, et il faut préciser aussi le quai de chargement qui sera déterminé par la fonction " AffectQuai ".
- A chaque fois qu'une rotation est affectée à un navire, le jour de disponibilité de ce navire sera incrémenté par le nombre de jour qu'il met pour faire cette rotation.

5.4.4. Paramètres de l'algorithme Génétique (Population initiale)

Pour l'application de l'algorithme, nous avons choisi de prendre une population initiale de 100 individus, tous réalisables. La génération d'une population non-homogène se fait de manière aléatoire avec la procédure "**SolRéal**".

5.4.5. Adaptation des opérateurs :

1- L'opérateur de sélection :

L'opérateur de sélection sélectionne = 50 pour la reproduction, par la méthode la roulette, détaillée dans le chapitre précédent. Rappelons que cette méthode favorise les individus les plus adaptés proportionnellement à leurs fonctions d'adaptation.

2- L'opérateur de croisement :

L'opérateur de croisement permet de créer des nouvelles combinaisons des paramètres des composants. Nous avons choisi dans notre cas d'utiliser le croisement à découpage de chromosomes à un point. Pour effectuer ce type de croisement sur des chromosomes, on tire aléatoirement une position de croisement identique dans chacun des parents. On échange ensuite les deux sous-chaînes terminales de chacun des deux chromosomes, ce qui produit deux enfants.

Pour ce qui est du taux de croisement nous avons choisi, **Pc = 0;6**.

Exemple 5.2. : Soient deux individus X, Y tirés aléatoirement de la population courante.

Dans cet exemple, on croise à la 6ième position.

$X = \{2; 2; 1; 2; 3 ; 5; 1; 1; 3; 4; 5; 4; 0; 0; 0; 0; 0; 0\}$



$Y = \{1; 1; 2; 3; 4 ; 4; 2; 1; 5; 2; 3; 2; 4; 2; 2; 2; 3; 1\}$

Après croisement on obtient :

$X' = \{2; 2; 1; 2; 3 ; 4; 2; 1; 5; 2; 3; 2; 4; 2; 2; 2; 3; 1\}$

$Y' = \{1; 1; 2; 3; 4 ; 5; 1; 1; 3; 4; 5; 4; 0; 0; 0; 0; 0; 0\}$

La procédure de croisement est décrite ci dessous :

Pour i := 1 à faire

 1- Tirer au hasard deux individus X et Y de la population sélectionnée à la reproduction ;

 2- Tirer aléatoirement et uniformément "r" un réel entre 0 et 1 ;

 3- **Si** ($r < P_c$) **Alors**

 -Générer aléatoirement une position "P" ;

 -Croiser les individus X et Y à la position "P" pour obtenir

X' et Y'

Sinon

 Garder les deux vecteurs X et Y ;

Fin Si

Fin Pour

Algorithme 5.2. Procédure Croisement

3-L'opérateur de mutation :

L'opérateur de mutation est un opérateur unaire, générant un seul individu à partir d'un autre et qui consiste à tirer aléatoirement un gène dans le chromosome et à le remplacer par une valeur aléatoire. Pour ce qui est du taux de mutation nous avons choisi, $P_m = 0; 02$.

Exemple 5.3 :

Soit un individu X tiré aléatoirement de la population courante.

Dans cet exemple, on mute X à la 5ieme position.

$X = \{2; 2; 1; 2; 3 ; 5; 1; 1; 3; 4; 5; 4; 0; 0; 0; 0; 0; 0\}$

Après mutation on obtient :

$X' = \{2; 2; 1; 2; 5; 5; 1; 1; 3; 4; 5; 4; ; 0; 0; 0; 0; 0; 0\}$

La procédure de mutation est décrite ci dessous :

Pour i := 1 à faire

 1- Tirer un individu de la population sélectionnée à la reproduction

 2-Tirer un réel "a" uniformément dans [0,1] ;

 3- **Si** ($a < P_m$) **Alors**

 3.1- Muter l'individu :

 3.2-Tirer au hasard une position i entre 1 et nbNavire

 4- **Si** (Le nombre de rotations affectées au navire est supérieur à 1) **Alors**

 4.1-Tirer au hasard un jour j de l'année

 4.2- **Si** (Le navire i effectue une rotation le jour j) **Alors**

 Enlever la rotation r au navire i

Sinon

 Affecter au navire i une nouvelle rotation, à partir du jour j, si la
 contrainte Rotation est toujours vérifiée

Fin Si

Sinon

 Affecter au navire i une nouvelle rotation

Fin Si

Fin Si

Fin Pour

Algorithme 5.3 : Procédure Mutation

4-L'opérateur de sélection pour le remplacement

L'opérateur sélection pour le remplacement sélectionne tout simplement les $= 50$ plus mauvais individus pour les remplacer avec les $= 50$ meilleurs individus parmi les $+ n$ individus constitués par génération courante plus les enfants reproduits par les opérateurs croisement et mutation. Donc on utilise la stratégie (+), cette stratégie a été détaillé dans le chapitre précédent.

5-Critère d'arrêt

Le critère d'arrêt pour lequel on a opté, est celui de fixer le nombre maximum d'itération

5.5. Adaptation de la méthode de la recherche Tabou

5.5.1. Le codage de la solution

La solution qui sera créer et en suite manipuler par l'algorithme, sera représentée par le type Sol déclaré précédemment.

5.5.2. Génération d'une solution réalisable

Pour générer une solution réalisable on utilise la même Procédure utilisée pour générer des solutions réalisables pour la méthode génétique, avec une petite modification, au niveau du tri aléatoire des demandes mensuelles, qui sera remplacé par le tri décroissant des demandes mensuelles. On peut se permettre de faire ce tri, car avec la méthode Tabou, on n'a pas besoin de générer plusieurs solutions différentes. Donc l'organigramme de la procédure qui génère des solutions réalisable pour la méthode de la recherche tabou, est représenté dans la figure 4.6 :

5.5.3. Génération des solutions voisines

La génération de solutions voisines est une étape très importante dans l'application de l'algorithme de la recherche tabou. L'algorithme doit avoir accès à toutes les solutions réalisables.

Pour cette raison qu'on définit l'algorithme suivant :

Début

1- Tirer aléatoirement deux navires N1 et N2 de La liste des navires ; **aller en 2**

2-Tirer aléatoirement un mois parmi les mois de la période d'étude ; **aller en 3**

3- **Si** (N1 et N2 effectuent des rotations le mois m) **Alors**

 3.1-Choisir aléatoirement de chaque navire une Rotation, et les substituer

 4- **Si** (la solution est toujours réalisable) **Alors**

 aller en 8

Sinon

 Aller en 1

Fin Si

Fin Si

5- **Si** (L'un des navires N1, N2 effectue des rotations le mois m, et l'autre non) **Alors**

 5.1- Choisir aléatoirement une rotation du premier navire et l'affecter au deuxième

 6- **Si** (la solution est toujours réalisable) **Alors**

 aller en 8

Sinon

 Aller en 1

Fin Si

Sinon

 7- **Si** (les deux navires N1, N2 n'effectuent aucune rotation le mois m) **Alors**

 Aller en 1

Sinon

Fin Si

Fin Si

8-Une solution voisine est générée,

Fin

Algorithme 5.4 : Procédure Génération de solutions voisines

5.5.4. Listes Tabous adaptées à notre problème

Pour éviter le cyclage de l'algorithme, on définit deux listes Tabous, **ListeTran** et **ListeSub**.

1.ListeSub :

Mémorise un nombre fixe de types substitutions de rotations. les cellules de la liste sont représentées par des Quadruplés de la forme suivante :

(Rotation1, Rotation2, Navire1, Navire2). Et on lit :(La rotation1 et la rotation2 effectuées respectivement par le navire1 et le navire2 sont substituées), donc on interdira toute candidature d'une solution obtenue par la modification inverse à celles contenue dans la liste ListeSub.

2. ListeTran :

Mémorise un nombre fixé de types transferts de rotations. les cellules de la liste sont représentées par des triplés de la forme suivante :

(**Rotation Δ , Navire1, Navire2**), et on lit :(La rotation Δ sera transférée du navire1 vers le navire2), donc on interdira toute candidature d'une solution obtenue par la modification inverse à celles contenue dans la liste ListeTran.

Tailles des Listes :

En effectuant des tests, on a fixé les tailles des deux listes à 7. Cette taille est très envisageable dans l'adaptation de la méthode recherche tabou.

5.5.5. Critère d'aspiration et Critère d'arrêt :

Une solution tabou doit dans certains cas perdre son statut tabou pour devenir candidate à la meilleure solution voisine, et d'améliorer la meilleure solution rencontrée, pour cela on définit la fonction d'aspiration suivante

$$A(F(s)) = F(s^{\circ}) .$$

Le critère d'arrêt pour lequel on a opté, est celui de fixer le nombre maximum d'itération.

5.5.6. Algorithme Recherche Tabou Adapté à notre problème :

Début

Var

niter : entier positif ;

{Compte le nombre d'itération effectuées dans l'exploration de toutes les solution passées}

bestiter : entier positif ; {Mémorise l'itération de la meilleure solution retrouvée}

maxiter : entier positif ; {Nombre maximum d'itération}

1- Trouver une solution initiale réalisable par la procédure **SolRéalTabou** ;

2- Initialiser les listes **ListeSub** et **ListeTran** ;

3- **niter** := 0 ;

4- **bestiter** := 0 ;

5- **Tant que** (**niter** \leq **maxiter**) faire

 5.1- **niter** := **niter** + 1 ;

 5.2- **Tant que** (il existe toujours des solutions voisines non explorées et
 n \leq **nbV**) faire

 5.2.1- Générer **s'** une solution voisine avec **GénéVoisin**

 5.2.3- **Si** (**f(s')** < **A(f(s))**) ou (le mouvement vers **s'** n'est pas dans
 listeSub ou **listeTran**) **Alors**

 Ajouter **s'** à l'ensemble des solutions voisines ;

Fin Si

Fait

 5.3- Retenir la meilleure solution dans l'ensemble des solutions voisines ;

 5.4- **s** := **s'** ;

 5.5- Mettre à jour les listes **ListeSub** et **listeTrans** ;

 5.6- Mettre à jour la fonction d'aspiration **A**;

 5.7- **Si** (**f(s)** < **f(s°)**) **Alors**

s° := **s** ; **bestiter** := **niter** ;

Fin Si

Fait

Fin

Algorithme 5.5: Algorithme de la recherche Tabou adapté à notre problème

Chapitre 5. Résolution de problème 1 et Réalisation d'un logiciel

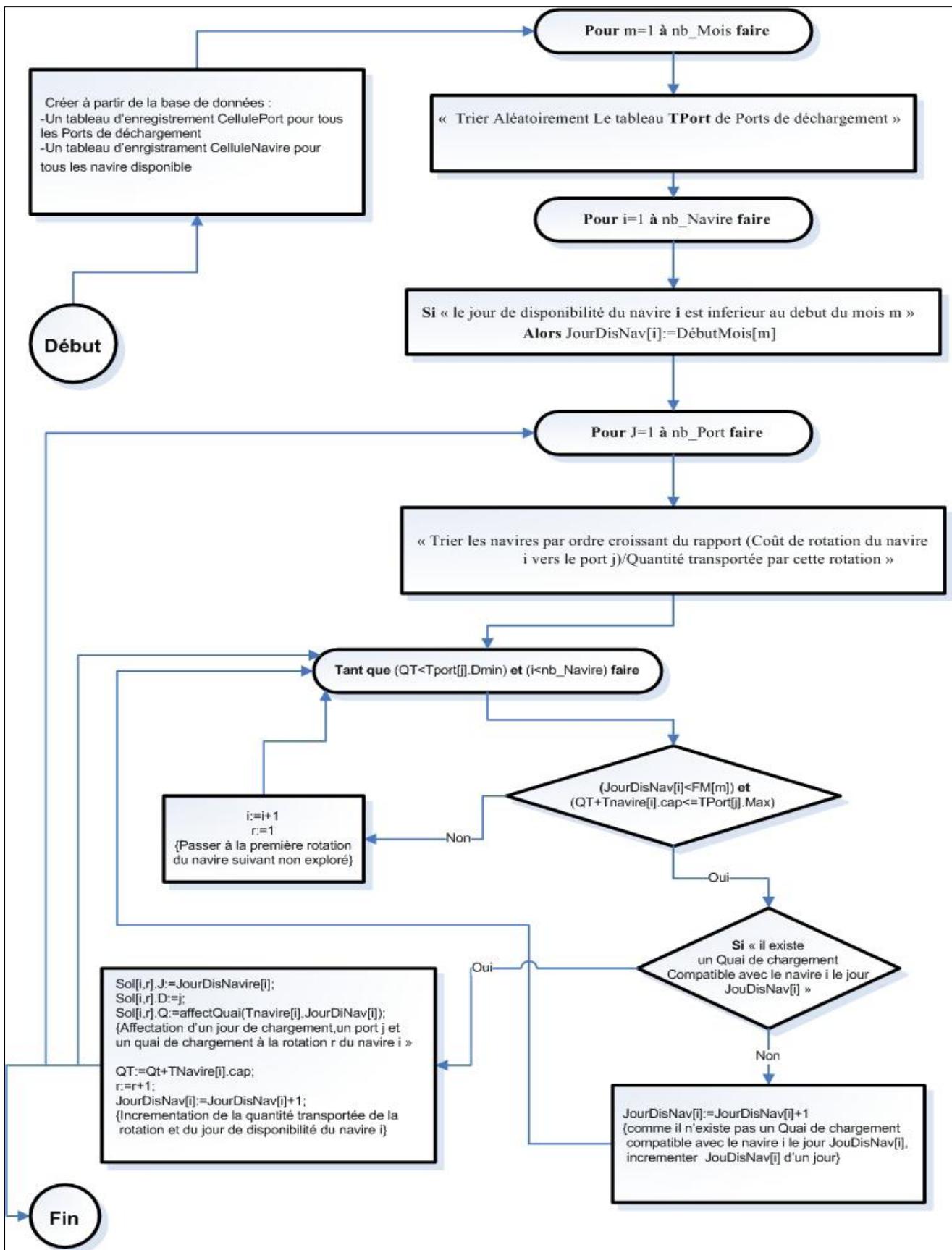


Fig 5.1. Organigramme de la Procédure population initiale

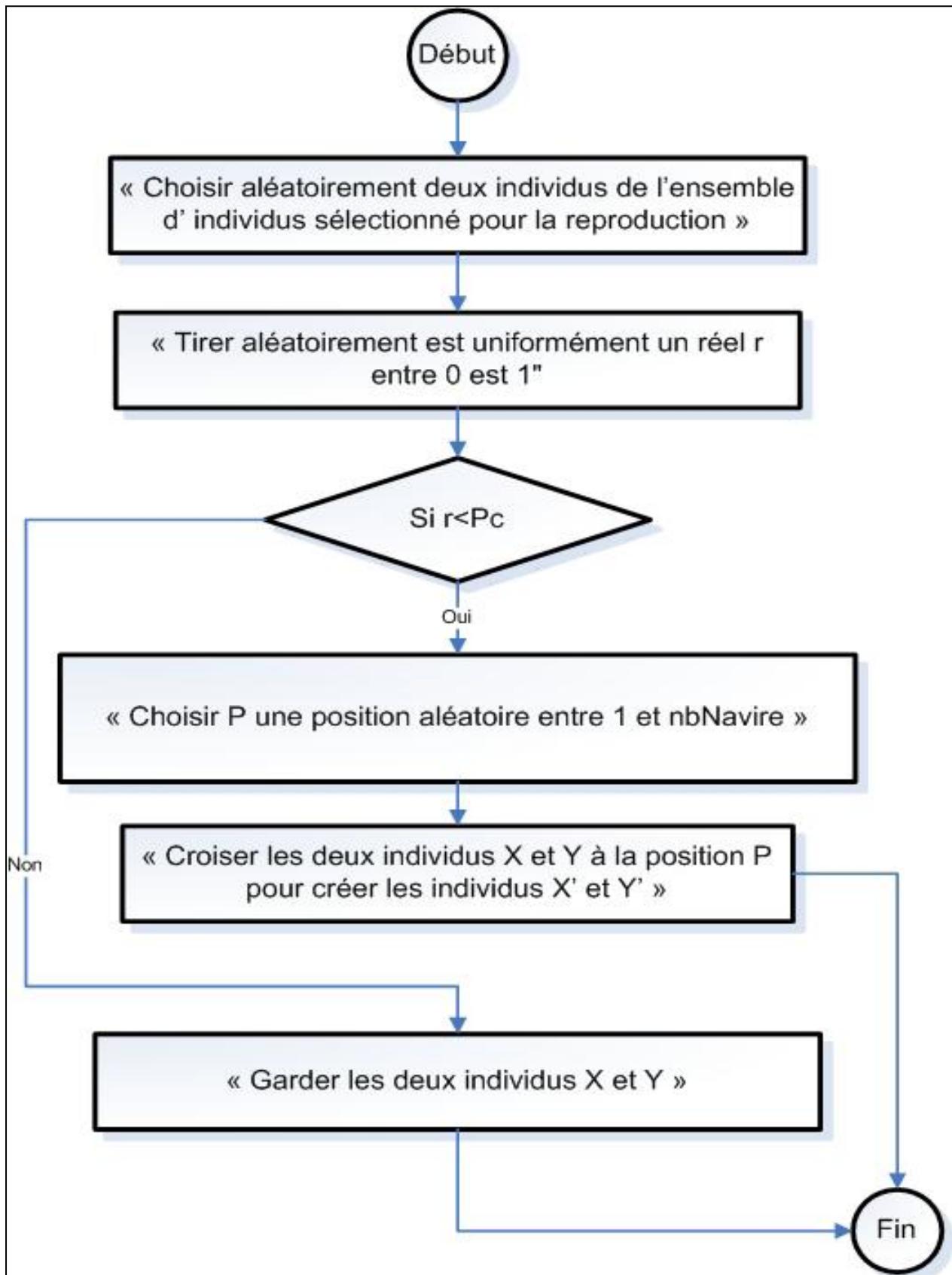


Fig. 5.2. Organigramme de la procédure croisement

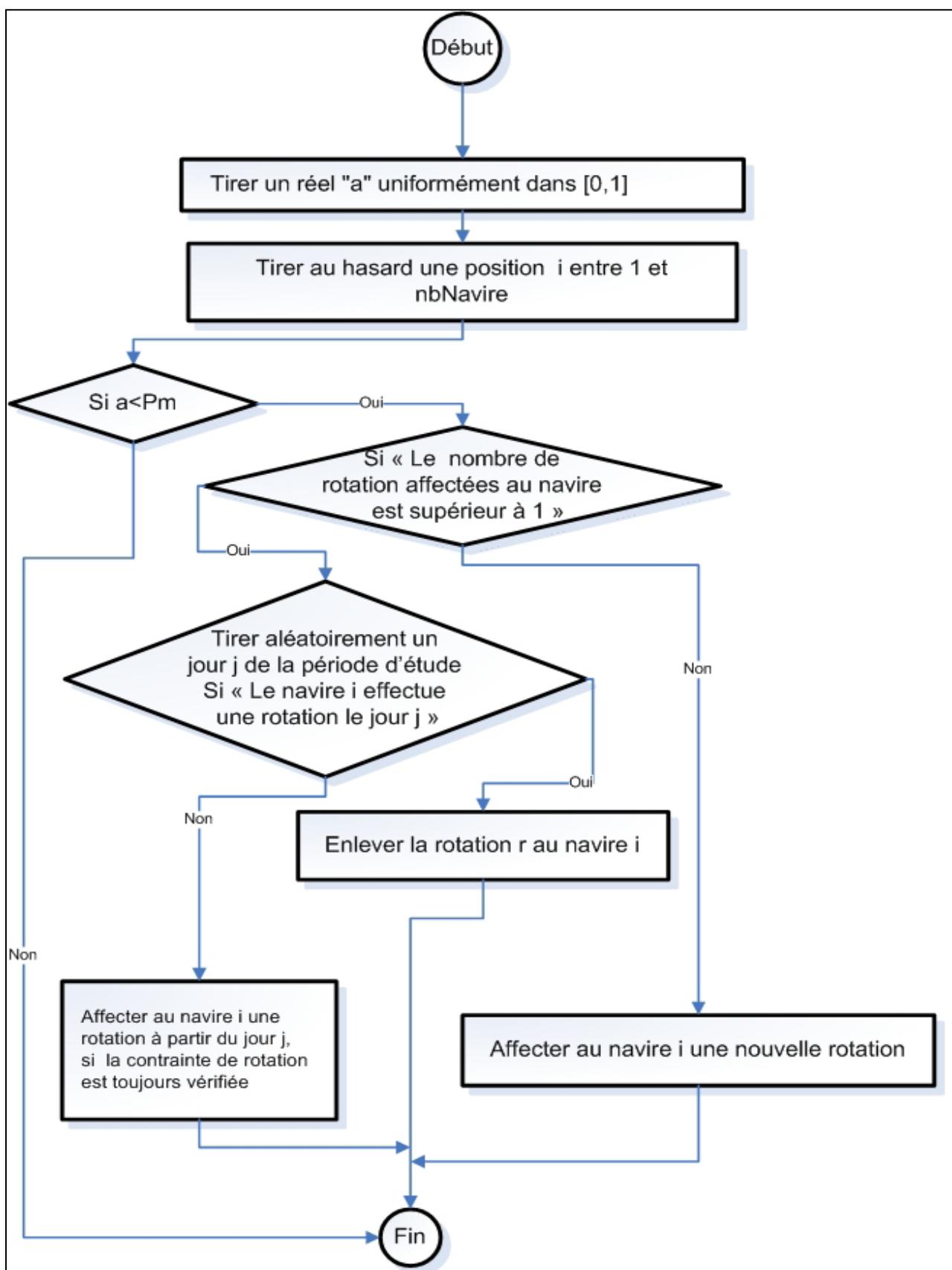


Fig. 5.3. Organigramme de la procédure mutation

Chapitre 5. Résolution de problème 1 et Réalisation d'un logiciel

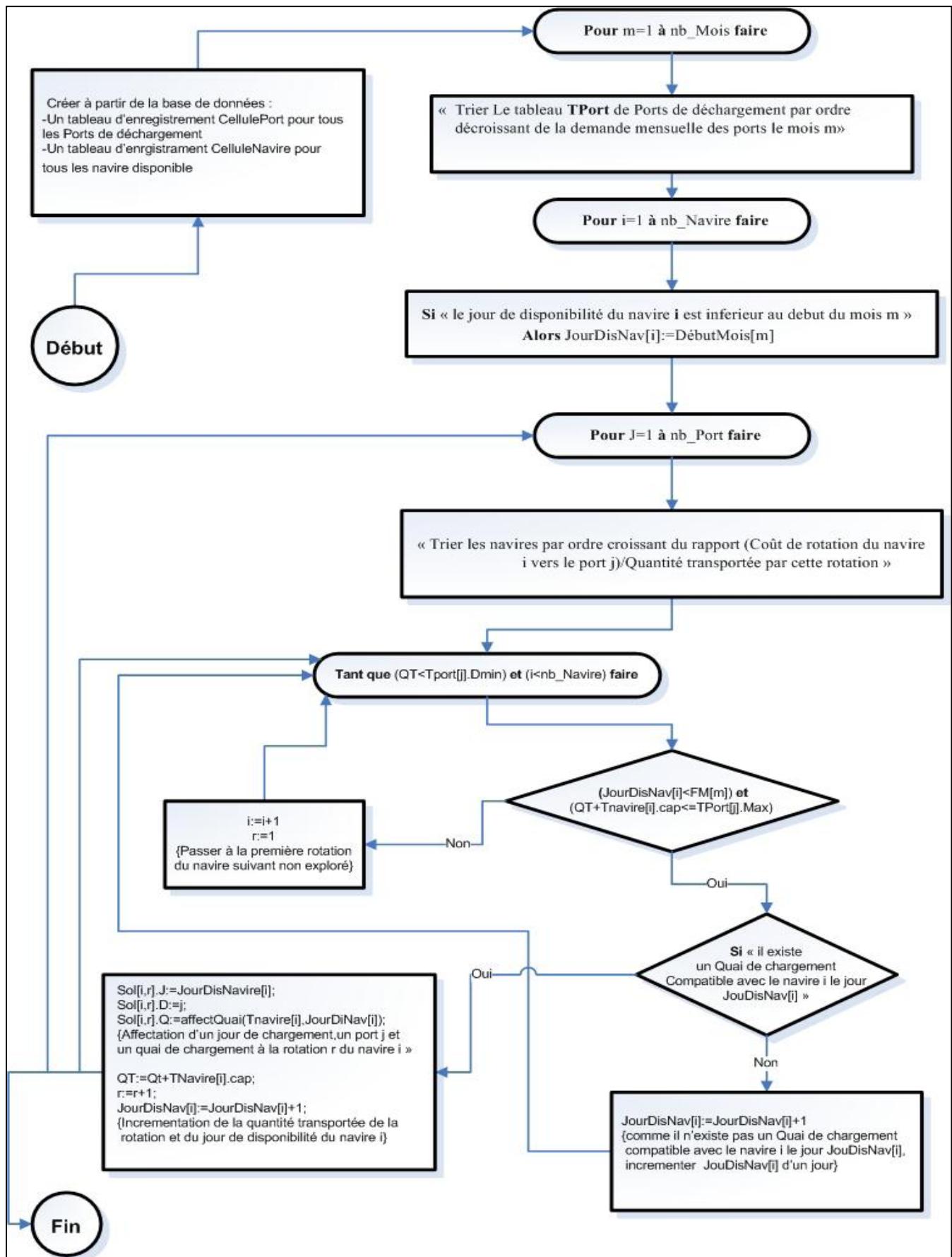


Fig. 5.4. Organigramme de la Procédure SolRéalTabou

5.6. Réalisation d'un logiciel d'application pour notre problème 1 :

Lors de la réalisation de cette étude, nous avons été amenés à concevoir un logiciel dans le but d'appliquer les méthodes de résolution au problème posé.

Notre logiciel est une application MDI qui a été conçue avec Borland Delphi 7 Entreprise.

5.6.1. Qu'est-ce que Delphi ?

Delphi est un environnement de programmation visuel orienté objet pour le développement rapide d'application (RAD) [52] . En utilisant Delphi, vous pouvez créer de puissantes applications pour Microsoft Windows XP, Microsoft Windows 2000 et Microsoft Windows 98, avec un minimum de programmation. Delphi fournit tous les outils nécessaires pour développer, tester et déployer des applications, notamment une importante bibliothèque de composants réutilisables, une suite d'outils de conception, des modèles d'applications et de fiches et des experts de programmation [52],[53],[54].

5.6.2. Qu'est ce qu'une application MDI?

Dans une application MDI (Interface de Document Multiple), plusieurs documents ou fenêtres enfants peuvent être ouverts dans une seule fenêtre parent. Cela est courant dans les applications telles que les tableurs ou les traitements de texte [53].

5.6.3. Présentation du notre logiciel « Flotte GPL 2014 »

Le logiciel Flotte GPL 2014 est destiné à la prise en charge de la sélection et de la planification de la flotte de GPL, susceptibles de permettre à Sonatrach de satisfaire tous ses clients en minimisant le coût de transport. Donc Flotte GPL 2014 permet de :

- 1- Sélectionner la flotte optimale parmi une liste de navires ;
- 2- Etablir un planning sur toute l'année ou sur une saison, en précisant ;
 - ✓ La date de chargement pour chaque rotation ;
 - ✓ La date du retour du navire de cette rotation ;
 - ✓ Le quai de chargement du navire ;

3. La gestion de toutes les données en relation avec le problème, cette gestion est possible grâce à une base de données locales implémentée sur La SGBD Paradox, et une panoplie de masques de saisies qui facilitent et organisent la manipulation de ces données.

Présentation de la "fiche principale" :

Cette fiche est la fenêtre principale du logiciel, elle permet d'accéder à toutes les autres fiches, soit par l'utilisation du menu défilant, soit en utilisant les boutons de raccourci.

Chaque fois que l'utilisateur pointe le curseur sur un bouton raccourci, la barre d'état affiche la commande correspondante.



Fig.5.5. La fiche principale du notre logiciel

Présentation de la fiche "nouveau port" :

Cette Fiche permet l'enregistrement des informations d'un nouveau port de déchargement, en toute sécurité. Cette fiche renvoie des messages d'avertissements, à chaque fois que l'utilisateur essaye d'introduire des données incohérentes.

Ces données pourront être modifiées ou supprimées avec la même facilité, en utilisant d'autres fiches similaires disponibles sur ce logiciel.

Chapitre 5. Résolution de problème 1 et Réalisation d'un logiciel

The screenshot shows a Windows application window titled "Nouveau Port de déchargement". It has two main sections: "Données Port Déchargement" (left) and "Demande du Port" (right).
The "Données Port Déchargement" section contains fields for:

- PORT (text input)
- Distance (Miles) (text input)
- Capacité d'accueil minimale (TM) (text input)
- Capacité d'accueil maximale (TM) (text input)
- Frais portuaires (\$/TM) (text input)
- Débit de déchargement (TM/h) (text input)

A checkbox labeled "Valide" is present.
The "Demande du Port" section contains a table titled "Quantités mensuelle transportées (TM)" with 8 rows (Mois 1 to Mois 8) and columns for "Quantités Min" and "Quantités Max". All values in the table are 0.
At the bottom, there are two radio button options: "Demande min/max" (selected) and "Demande ±10%".
Buttons at the bottom right include "Enregistrer" (Save), "Annuler" (Cancel), and "Aide" (Help).

Fig.5.6. Fiche « nouveau port »

Présentation de la fiche "nouveau navire" :

Cette Fiche permet l'enregistrement des informations d'un nouveau navire, en toute sécurité. Cette fiche renvoie des messages d'avertissements, à chaque fois que l'utilisateur essaye d'introduire des données incohérentes. Ces données pourront être modifiées ou supprimées avec la même facilité, en utilisant d'autres fiches similaires disponibles sur ce logiciel.

The screenshot shows a Windows application window titled "Nouveau Navire". It has two main sections: "Données Navires" (left) and "Caractéristiques Techniques" (right).
The "Données Navires" section contains fields for:

- NAVIRE (text input)
- Cout d'affrètement (1000\$) (text input)
- Propriété du Groupe Sonatrach
- Valide

The "Caractéristiques Techniques" section contains fields for:

- Capacité (TM)
- Date construction
- Longueur (m)
- Largeur (m)
- Tirant d'eau (m)
- Nombres de cuves
- Vitesse (Noeud)
- Consommation FO en mer (TM/J)
- Consommation DO en mer (TM/J)
- Consommation DO en port (TM/J)
- Consommation FO en port (TM/J)
- Débit de déchargement

Buttons at the bottom right include "Enregistrer" (Save), "Annuler" (Cancel), and "Aide" (Help).

Fig.5.7. Fiche « nouveau navire »

Présentation de la fiche "nouveau poste de chargement" :

Cette Fiche permet l'enregistrement des informations d'un nouveau poste de chargement, en toute sécurité. Cette fiche renvoie des messages d'avertissemens, à chaque fois que l'utilisateur essaye d'introduire des données incohérentes. Ces données pourront être modifiée ou supprimer avec la même facilité, en utilisant d'autres fiches similaires disponibles sur ce logiciel.



Fig.5.8. Fiche « Nouveau poste de chargement »

Présentation de la fiche "listes" :

Cette Fiche permet d'afficher la liste des navires, ports ou postes de chargement. Il est recommandé de ne pas modifier les données en utilisant cette fiche, car l'utilisateur risque de saisir des données incohérentes.

The screenshot shows a window titled "Listes" with tabs for "Liste Navires", "Liste Ports de Déchargement", and "Liste Postes de Chargement". The "Liste Navires" tab is selected, displaying a grid of ship data. The columns are: NAVIRES, Année cons, Capacité (Tm), Propriété de SH, Coût Affrètement (\$), Longueur (m), and Largeur (m). The data includes various ship names like Brege eagle, Candidat1, Clipper star, etc., with their respective details. At the bottom are navigation buttons and a "Retour" (Return) button.

NAVIRES	Année cons	Capacité (Tm)	Propriété de SH	Coût Affrètement (\$)	Longueur (m)	Largeur (m)
Brege eagle	1978	44000	Faux	0	228,99	32,1
Candidat1	2000	12000	Faux	0	200	5
Candidat2	2000	4000	Faux	0	200	5
Candidat3	2000	8000	Faux	0	200	5
Candidat4	2000	48000	Faux	0	200	5
Candidat5	2000	4000	Faux	0	200	5
Candidat6	2000	4000	Faux	0	200	5
Candidat7	2000	8000	Faux	0	200	5
Candidat8	2000	1600	Faux	0	200	5
Clipper star	2003	32000	Faux	0	205	32,
Djanet	2000	32000	Vrai	0	230	3
Drothea schul	1980	3000	Faux	0	110,9	15,
Gaschem jumme	1993	2700	Faux	0	113,22	15,
hans mearsk	1993	11700	Faux	0	159,98	25,
Hassi mess 2	2005	32000	Vrai	0	204,92	32,
Henriette mear	1994	11700	Faux	0	159,98	25,
hermann schul	1980	3000	Faux	0	110,9	15,
Maharshi dattat	1983	13000	Faux	0	159,3	24,
Navigator venus	2000	13132	Faux	0	169,97	24,
Niels mearsk	1991	8600	Faux	0	158	21,
Reggane	1999	48000	Vrai	0	231,03	3
Rhoud enouss	2004	32000	Vrai	0	204,92	32,

Fig.5.9. Fiche « listes »

Présentation de la fiche "durées des rotations" :

Chapitre 5. Résolution de problème 1 et Réalisation d'un logiciel

Cette Fiche permet simplement d'afficher les durées de rotation de chaque navire, vers chaque port compatible.

Voir Durées de Rotation des Navires							
	Agadir	Alexandrie	Djorf	Dortyol	Houston	Livourne	Mohame
Adrar				19 J	29 J		22 J
Berge hugin				16 J			16 J
Berge munin		18 J					14 J
Berkine	8 J		7 J				7 J
Brege eagele				22 J	32 J		29 J
Candidat1		14 J				8 J	11 J
Candidat2	8 J	11 J	7 J			6 J	7 J
Candidat3		13 J				7 J	9 J
Candidat4				23 J	32 J		31 J
Candidat5	8 J	12 J	7 J			7 J	7 J
Candidat6	8 J	12 J	7 J			7 J	7 J
Candidat7		13 J				7 J	9 J
Candidat8	7 J		5 J				
Clipper star				19 J	31 J		22 J
Dianet				19 J	31 J		22 J
Drothea schul	8 J		7 J				7 J
Gaschem jumme	8 J		6 J				6 J

Fig.5.10. Fiche « durées des rotations »

Présentation de la fiche "comparaison des méthodes" :

Cette Fiche permet de lancer l'exécution des deux algorithmes de résolutions, ainsi que la comparaison de ces méthodes, et le choix du mode d'affichage graphique ou numérique de la solution.

Comparaison des méthodes

Méthode Génétique	Méthode de la Recherche Tabou
Nombres d'itération <input type="text" value="20"/>	Nombres d'itération <input type="text" value="500"/>
<input type="button" value="Résoudre"/>	<input type="button" value="Résoudre"/>
Coût Total du Transport <input type="text" value="20329799 \$"/>	Coût Total du Transport <input type="text" value="20492686 \$"/>
<input type="button" value="Affichage Alphanumérique"/>	<input type="button" value="Affichage Alphanumérique"/>
<input type="button" value="Affichage Graphique"/>	<input type="button" value="Affichage Graphique"/>
Progression de la résolution <div style="width: 100%; height: 10px; background-color: #ccc; border: 1px solid #ccc; margin-bottom: 5px;"></div> <div style="width: 100%; height: 10px; background-color: #00ff00; border: 1px solid #ccc;"></div>	Progression de la résolution <div style="width: 100%; height: 10px; background-color: #ccc; border: 1px solid #ccc; margin-bottom: 5px;"></div> <div style="width: 100%; height: 10px; background-color: #00ff00; border: 1px solid #ccc;"></div>
<input type="button" value="Retour"/> <input type="button" value="Aide"/>	

Fig.5.11. Fiche « Comparaison des méthodes »

Présentation de la fiche "résolution" :

Cette Fiche permet de lancer l'exécution de l'algorithme de la recherche Tabou seul car ce dernier est généralement plus performant que l'algorithme génétique, ainsi que le choix du mode d'affichage graphique ou numérique de la solution.

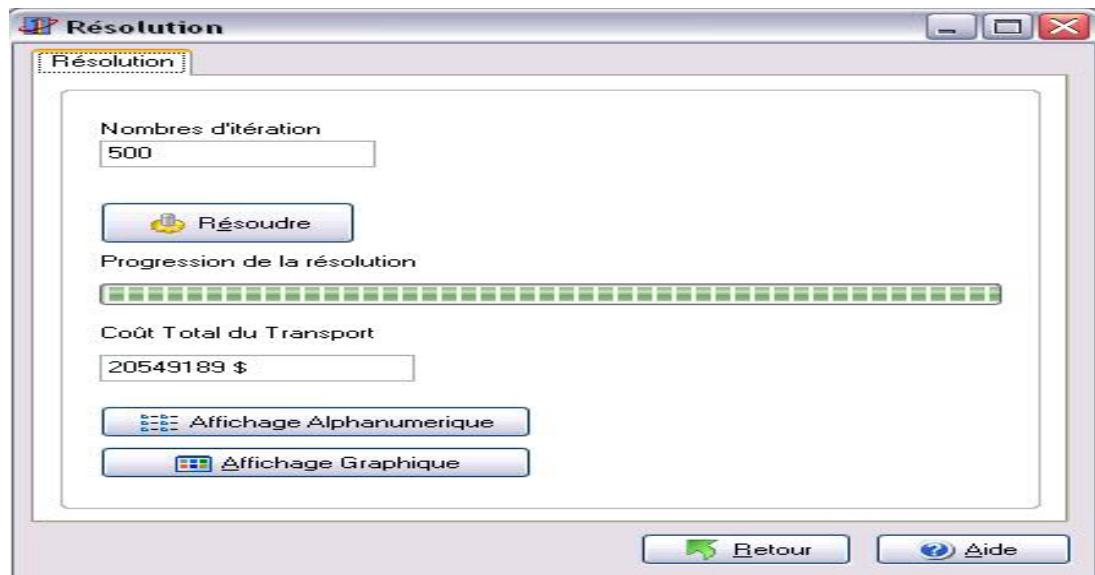


Fig.5.12.Fiche « Résolution »

Présentation de la fiche "affichage alphanumérique de la solution" :

Cette Fiche permet l'affichage de la solution. Elle permet aussi l'impression de la solution.

Affichage Alphanumérique de la solution de la méthode De la Recherche Tabou				
Jour / Détination / Quai	Rotation 1	Rotation 2	Rotation 3	Rotation 4
Adrar	---	---	---	---
Berge hugin	32 / Yarimca / D1	47 / Doryol / D1	276 / Yarimca / D1	---
Berge munin	1 / Alexanderie / S1	19 / Alexanderie / S3	37 / Alexanderie / S3	55 / Alexanderie / S3
Berkine	2 / Mohamadia / S3	9 / Mohamadia / S3	16 / Agadir / S3	24 / Agadir / S3
Brege eagele	---	---	---	---
Candidat1	1 / Alexanderie / S3	15 / Alexanderie / S3	29 / Alexanderie / S3	43 / Alexanderie / S3
Candidat2	---	---	---	---
Candidat3	---	---	---	---
Candidat4	1 / Sues / M6	15 / Sues / M6	32 / Sues / M6	46 / Sues / M6
Candidat5	---	---	---	---
Candidat6	2 / Agadir / S2	10 / Djorf / S3	17 / Djorf / S3	24 / Djorf / S1
Candidat7	1 / Alexanderie / S2	14 / Livourne / S3	21 / Livourne / S3	28 / Livourne / S3
Candidat8	32 / Agadir / S2	61 / Agadir / S1	92 / Agadir / S1	122 / Agadir / S1
Clipper star	3 / Yarimca / M6	336 / Sues / M6	349 / Sues / M6	---
Djanet	2 / Sues / M6	16 / Doryol / M6	35 / Sues / M6	62 / Doryol / M6
Drothea schul	---	---	---	---
Gaschem jumme	3 / Nador / S3	7 / Nador / S3	11 / Nador / S3	15 / Nador / S1

Fig.5.13. Fiche « affichage alphanumérique de la solution »

Présentation de la fiche "affichage graphique de la solution" :

Cette Fiche permet l'affichage intuitif de la solution, car la solution numérique est difficile à comprendre.



Fig.5.14. Fiche « affichage graphique de la solution »

5.7. Conclusion :

Quant à l'objectif, il exprime le vœu d'optimiser le coût total de l'opération de transport. Pour sa résolution, l'utilisation des méthodes exactes en un temps raisonnable s'avère prohibitive, ce qui nous a amené à adopter des méthodes approchées, d'autant plus que certaines d'entre elles ont prouvées leurs grande efficacité pour des problèmes de nature similaire. Dans notre cas (La Recherche Tabou) et (La Méthode Génétique) ont été les heuristiques utilisées à titre comparatif .

C'est dans cette optique que nous avons conçu le logiciel (Flotte GPL 2014). Du point de vu des décideurs, le résultat obtenu dans l'optimisation des coûts de l'investissement sur l'ensemble de la flotte requise était satisfaisant, du moment que les navires propres à la Sonatrach seront utilisés sur toute la période d'étude.

Chapitre 6

*Approche de résolution du problème 2
et réalisation d'un logiciel d'application*

6.1. Introduction

Dans ce présent chapitre, nous allons résoudre le problème 2. Nous présentons dans ce qui suit notre logiciel « Explor 1.0 » et son utilisation.

6.2. Résolution du problème 2.1

Après modélisation, en tenant compte des hypothèses, données, paramètres et contraintes d'un problème, nous avons abouti à un programme non linéaire, mono objectif à variables mixtes positives ou nulles, entières et réelles.

Compte tenue de la complexité de notre programme mathématique, la résolution de notre programme s'est faite par deux méthodes :

- En un premier lieu nous avons utilisé le logiciel Lingo14.0 qui utilise les algorithmes de programmation linéaire successive et les algorithmes de programmations non linéaires.
- Ensuite, nous avons décrit une heuristique qui nous permet de maximiser la distance entre deux lignes de tirs afin de minimiser le déplacement des camions vibrateurs, conduisant ainsi à la minimisation des coûts.

6.2.1. Résolution par des méthodes exactes :

Les problèmes d'optimisation peuvent être classés en linéaire ou non linéaire selon la relation entre les variables. Lingo est un outil qui permet de résoudre un problème d'optimisation qu'il soit linéaire ou non linéaire et de l'analyser. En entrée, ce solveur Lingo doit recevoir un modèle mathématique et il possède quatre solveurs qu'il utilise afin de résoudre les différents types de modèles :

- Solveur direct
- Solveur linéaire
- Solveur non linéaire
- Méthode de type séparation et évaluation

6.2.2. Résolution par des heuristiques

Les heuristiques, à la différence des méthodes exactes, sont souvent construites à partir de l'expérience ou d'analogie, plutôt que d'une analyse scientifique qui prend en compte un maximum d'éléments qui seraient difficilement exploitables.

Le principe de résolution :

Rappelons qu'au jour d'aujourd'hui, c'est le client qui fait le pré planning de l'étude, tous les résultats de l'étude en question figurent sur le cahier des charges qu'il présente à l'entreprise Sonatrach. On y trouve toutes les valeurs des variables de notre modèle mathématique décrit dans le chapitre précédent. En l'absence d'outils d'optimisation, cette dernière applique fidèlement toutes les valeurs des paramètres sismiques exigés par le client.

Dans le but de mettre à la disposition de l'entreprise sonatrach un outil d'aide à la décision, nous avons pensé qu'il est utile de mettre en œuvre un logiciel interactif permettant de faire des simulations pour finir par atteindre de bonnes solutions pour l'entreprise sans altérer l'objectif fixé par le client .

Le principe de l'heuristique tient compte de l'existence de variables entières dans notre modèle mathématique, sachant que c'est plutôt les variables continues qui influent le plus sur la valeur de la fonction objectif, vue que l'optimisation des déplacements des camions vibrateurs se base sur l'optimisation de la distance entre deux lignes de tirs consécutives (X_1) et de la distance entre deux lignes de réception consécutives (X_2). A cet effet, nous avons pensé à fixer les variables entières au niveau de notre modèle mathématique afin de le simplifier. Ces variables qui deviennent donc des constantes, peuvent être aussi bien fixées par le client que par le personnel qualifié de l'entreprise Soantach, à savoir ; ses ingénieurs et techniciens géophysiciens.

L'idée de notre heuristique consiste donc, à faire augmenter la valeur de X_1 tout en faisant diminuer le X_2 ainsi la distance totale parcourue par les camions vibrateurs sera diminuée.

Le modèle devient :

Chapitre 6. Approche de résolution du problème 2

$$\left\{
 \begin{aligned}
 & \text{Min}(z) = C_{13} + C_{14} \times \left(\frac{1}{X_1} \right), \dots \text{avec} \\
 & C_{13} = C_{SSV} \times X_7 + C_{RSV} \left(\frac{X_7 \times X_{11}}{X_{10}} \right) + \left(C_{CR} + C_{EQ} \times (X_5 \times (1+I)) \right) \times \left(\frac{X_7}{X_8} \right) + C_{PCT} \times X_5 \times X_7 \\
 & C_{14} = C_{SCC} \\
 & \frac{X_3}{2} \times \frac{X_4}{2} \geq \frac{F}{C_1} \dots (1) \text{ avec } C_1 = X_5 \times X_7 \\
 & C_2 \times X_4 - \frac{X_4}{2} \geq O_{IN} \dots (2) \Rightarrow X_4 \geq \frac{O_{IN}}{C_2 - \frac{1}{2}}, \text{ avec } C_2 = \frac{X_5}{2X_6} \\
 & C_3 \times X_2 - \frac{X_3}{2} \geq O_{CR} \dots (3), \text{ avec } C_3 = \frac{X_6}{2} \\
 & \left(C_2 \times X_4 - \frac{X_4}{2} \right)^2 + \left(C_3 \times X_2 - \frac{X_3}{2} \right)^2 \geq O_{MAX}^2 \dots (4) \\
 & X_1^2 + X_2^2 \leq O_{MIN}^2 \dots (5) \\
 & C_4 \times X_1 \times X_3 = X_2 \times X_4 \dots (6), \text{ avec } C_4 = \frac{X_8}{X_9} \\
 & C_5 \times X_1 \times X_3 = 1 \Rightarrow X_3 = \frac{1}{C_5 X_1}, \text{ avec } C_5 = X_7 \\
 & X_2 - C_6 X_3 = 0 \Rightarrow X_3 = \frac{X_2}{C_6} \Rightarrow X_2 = \frac{C_6}{C_5 X_1}, \text{ avec } C_6 = X_{10} \\
 & X_1 - (C_7 X_4) = 0 \Rightarrow X_4 = \frac{X_1}{C_7}, \text{ avec } C_7 = X_{11} \\
 & X_1, X_2, X_3, X_4 \geq 0, \text{ et } \in \mathbb{N}
 \end{aligned}
 \right.$$

(1) devient : $\frac{1}{2 \times C_5 \times X_1} \times \frac{X_1}{2 \times C_7} \geq \frac{F}{X_5 \times X_7} \Rightarrow \frac{1}{2} \times \frac{1}{2} \geq F \times \frac{X_{11}}{X_5}$

(2) devient : $X_4 \geq \frac{O_{IN}}{\left(C_2 - \frac{1}{2} \right)} \Rightarrow \frac{X_1}{C_7} \geq \frac{O_{IN}}{\left(C_2 - \frac{1}{2} \right)} \Rightarrow X_1 \geq \frac{C_7 \times O_{IN}}{\left(C_2 - \frac{1}{2} \right)}$

$$(3) \text{ devient : } \underbrace{\frac{C_{13} \times C_6}{C_5}}_{C_8} \geq \frac{1}{X_1} - \frac{1}{2 \times C_5} \times \frac{1}{X_1} \geq O_{CR} \Rightarrow X_1 \leq \frac{C_8 - C_9}{O_{CR}}$$

Sachant que $X_2 = \frac{C_6}{C_5 \times X_1}$ alors,

$$(4) \text{ devient : } X_1^2 \times \underbrace{\frac{1}{C_7^2} \times \left(C_2 - \frac{1}{2} \right)}_{C_{10}} + \frac{1}{X_1^2} \times \underbrace{\left(\frac{C_6}{C_5} \right)^2 \times \left(C_3 - \frac{1}{2 \times C_6} \right)^2}_{C_{11}} \geq O_{MAX}^2$$

$$\Rightarrow C_{10} \times X_1^2 + \frac{C_{11}}{X_1^2} \geq O_{MAX}^2 \Rightarrow C_{10} \times X_1^4 - O_{MAX}^2 \times X_1^2 + C_{11} \geq 0$$

$$(5) \text{ devient : } X_1^2 + \underbrace{\left(\frac{C_6}{C_5} \right)^2 \times \frac{1}{X_1^2}}_{C_{12}} \leq O_{MIN}^2 \Rightarrow X_1^4 - X_1^2 \times O_{MIN}^2 + C_{12} \leq 0$$

D'autre part, de (2) et (3) on a :

$$\frac{C_7 \times O_{IN}}{\left(C_2 - \frac{1}{2} \right)} \leq X_1 \leq \frac{C_8 - C_9}{O_{CR}}$$

En résumé, on a :

$$\begin{cases} C_{10} \times Y_1^2 - O_{MAX}^2 \times Y_1 + C_{11} \geq 0 \\ Y_1^2 - O_{MIN}^2 \times Y_1 + C_{11} \leq 0 \\ \frac{C_7 \times O_{IN}}{\left(C_2 - \frac{1}{2} \right)} \leq X_1 \leq \frac{C_8 - C_9}{O_{CR}} \\ X_1^2 = Y_1 \end{cases}$$

Notons D l'ensemble des valeurs possibles pour X_1

Le problème revient donc, à l'étude de la fonction à minimiser

$$Min(z) = C_{13} + C_{14} \times \left(\frac{1}{X_1} \right) \quad \text{Sachant que } X_1 \in D \subset \mathbb{R}^+$$

6.2.3. Implémentation et résultats

Nous aboutissons maintenant à l'étape finale à savoir l'élaboration d'un logiciel aussi convivial que possible, sans perdre de vue son aspect pratique et encore moins sa performance, muni d'une interface claire et accessible, facilitant ainsi son utilisation.

Avant de procéder à la présentation du logiciel, une description de l'environnement de programmation utilisé (**C++Builder**) s'avère nécessaire.

6.2.3.1. Langage de programmation C++Builder

C++Builder est un logiciel de développement rapide d'applications (*RAD*) conçu par Borland qui reprend les mêmes concepts, la même interface et la même bibliothèque que Delphi en utilisant le langage C++. Il permet de créer rapidement des applications Win32 ainsi qu'une interface graphique avec son éditeur de ressources. Il est compatible avec la norme ISO C++.

6.2.3.2. Présentation du notre logiciel et résultats

L'aboutissement de l'étude théorique s'est soldé par la création du logiciel **Explor1.0** qui implémente les différents algorithmes (Les Algorithmes de programmation non linéaire avec le logiciel Lingo 14.0, les heuristiques) ; en utilisant C++Builder.

Nous présenterons aussi les résultats obtenus par son exécution.

Descriptif du logiciel

Nom du logiciel : Explor1.0

Outil de développement : C++Builder .

Version du logiciel : V1.0

Réalisé par : Karim Benalia

Mode d'utilisation du logiciel

Lors du lancement de l'application, la fenêtre suivante s'affiche :

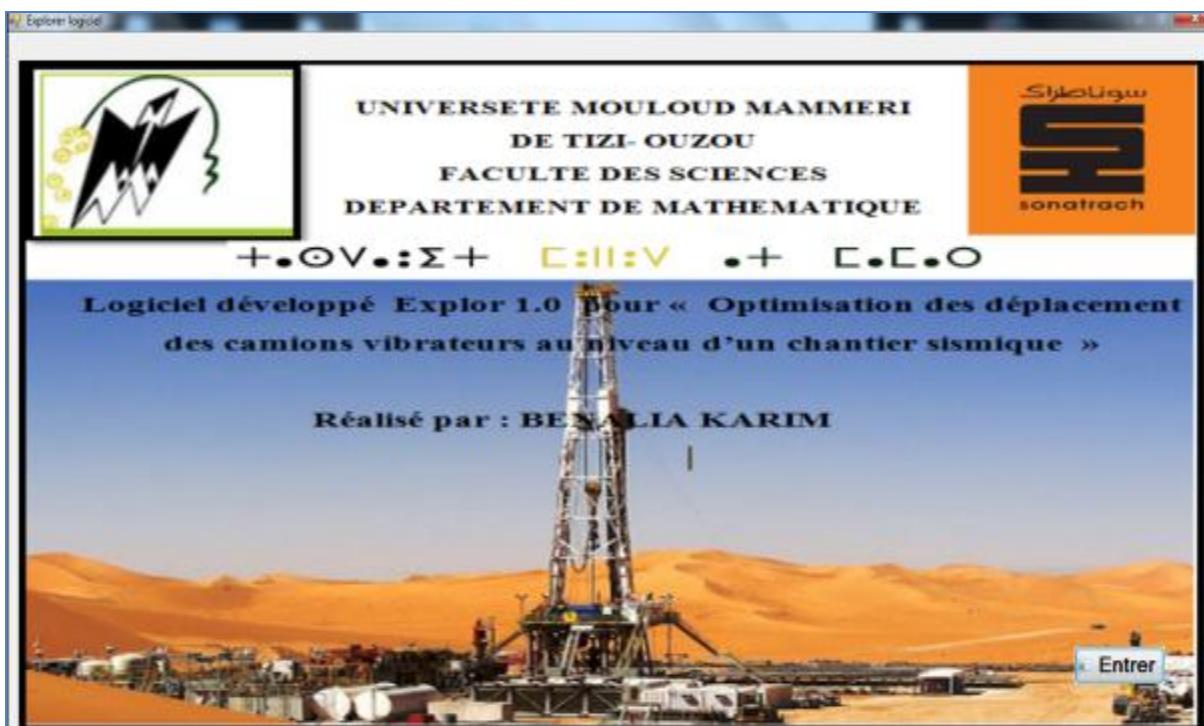


Fig.6.1. Fiche principale du notre logiciel

Par la méthode exacte :

Pour résoudre le modèle mathématique d'une manière exacte, cliquer sur **Méthode exacte** et la fenêtre suivante apparaît :

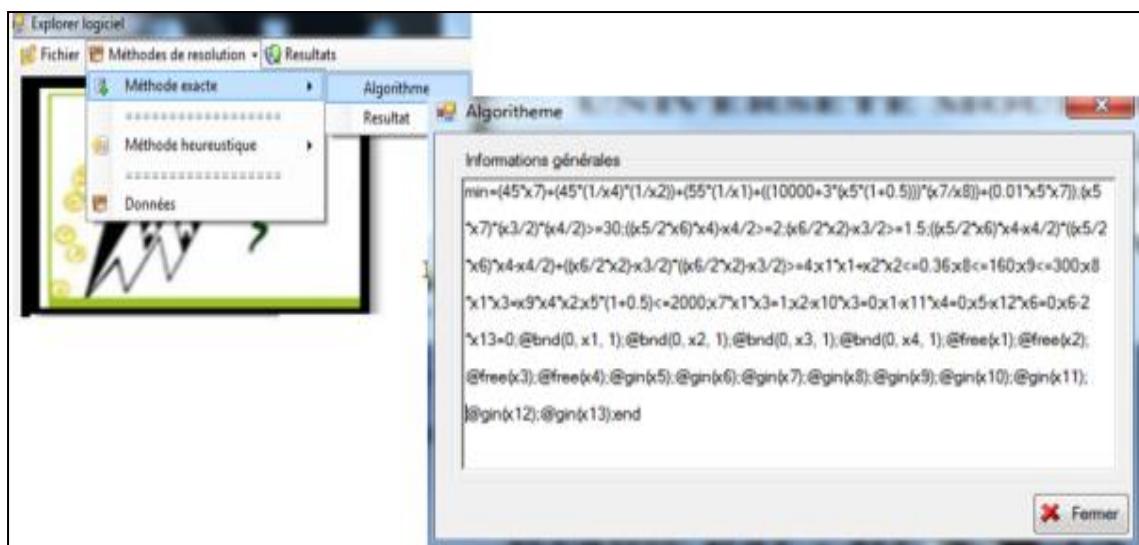


Fig 6.2. Résolution par la méthode exacte

Le résultat d'exécution s'afficher comme suit :

Chapitre 6. Approche de résolution du problème 2

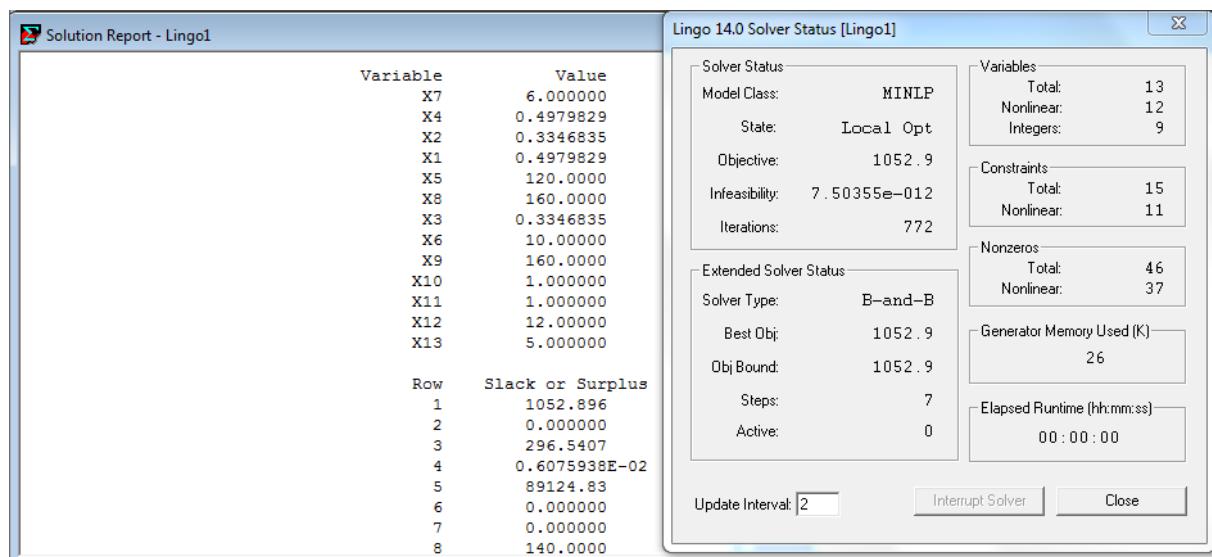


Fig 6.3. Résultat d'exécution de la méthode exacte

La solution est la suivante:

$$X = (0.497, 0.334, 0.334, 0.497, 120, 10, 6, 160, 160, 1, 1, 12, 5), Z^* = 1052.9$$

Par la méthode heuristique :

Pour résoudre le modèle en utilisant l'heuristique, il suffit de cliquer sur le bouton

Méthode Heuristique ainsi s'ouvre une nouvelle fiche pour saisir les valeurs des variables à fixer comme suit :

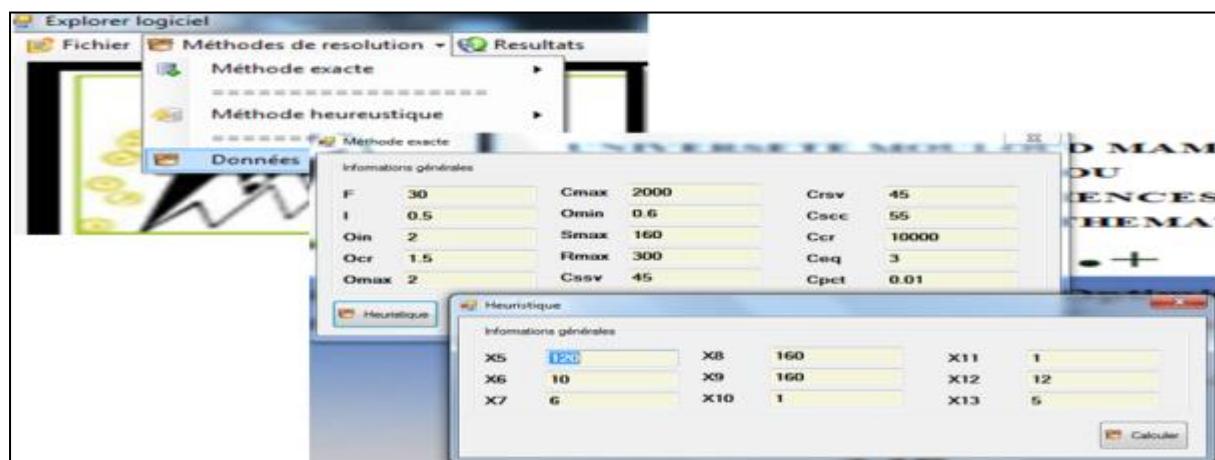


Fig. 6.4. Résolution par la méthode heuristique

Dans ce cas, nous avons fixés les valeurs entières par celles fournies par la méthode exacte et nous avons obtenu les mêmes valeurs des autres variables continues que celles données par la méthode exacte.

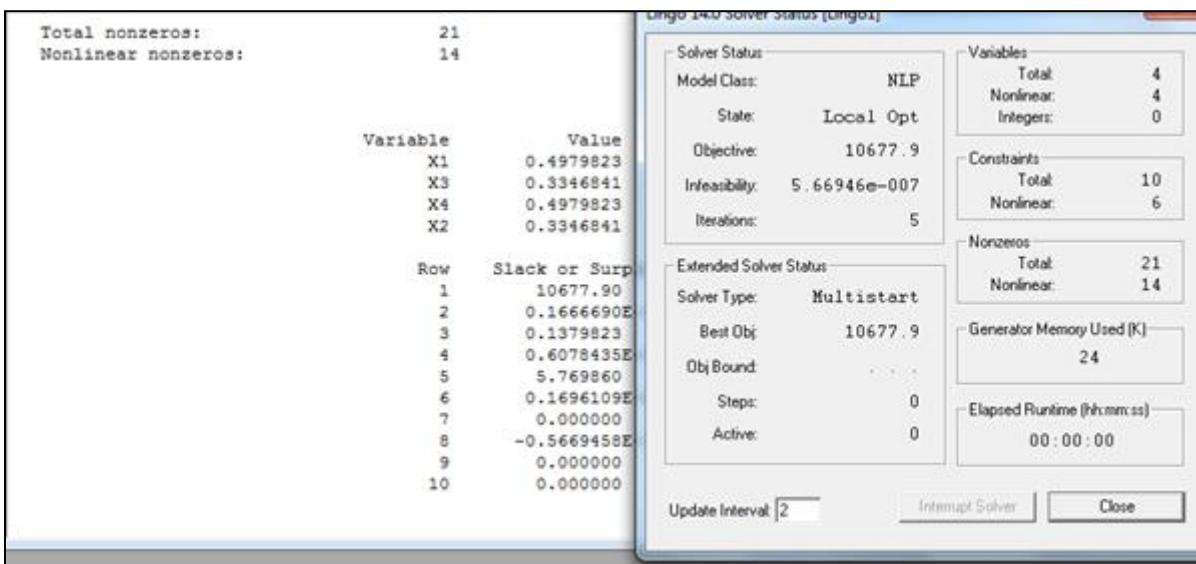


Fig. 6.5. Résultat d'exécution de la méthode heuristique

Résultats et interprétation

1-L'exécution du logiciel par la méthode exacte, donne la solution de la figure 6.3 (Résultat d'exécution de la méthode exacte) qui se présentent comme suit :

- La distance entre deux lignes de tirs consécutifs représentés par X_1 est égale à 0.497 Km
- La distance entre deux lignes de réceptions consécutives représentées par X_2 est égale à 0.334 Km
- La distance entre deux points de tirs consécutifs représentés par X_3 est égale à 0.334 Km
- La distance entre deux points de réception consécutives représentées par X_4 est égale à 0.497 Km
- Ce qui implique que le nombre de point vibré (VP) par Km^2 est égale à 6 représenté par X_7 , l'équipe sismique est limité par 160 points vibrés par jour représenté par X_8 , elle doit balayer 26.6 Km^2 de surface d'étude par jour afin de satisfaire la demande du client, pour que cela soit possible le nombre de canaux de réception déplacés par jour pour être posé à l'avant du profil doit être égal à 160, représenté par X_9 .
- Le coût de la prospection sismique est de 1052.9 Da par Km^2 .

2-L'exécution du logiciel par la méthode heuristique avec les données de la figure 6.4 (Résolution par la méthode heuristique), donne la solution de la figure 6.5 (Résultat

d'exécution de la méthode heuristique). Dans ce cas, nous avons fixés les valeurs entières par celles fournies par la méthode exacte et nous avons obtenu les mêmes valeurs des autres variables continues que celles données par la méthode exacte. Ce résultat ne fait que conforter notre démarche de conception de l'heuristique.

6.3. Résolution du problème 2.2

Chois de la méthode de résolution

L'analyse du problème posé, montre qu'on est en présence d'un problème linéaire qui est en fonction de variables bivalentes, de plus le problème posé n'étant pas de taille importante nous envisageons d'utiliser des méthodes exactes pour sa résolution (méthode de branch and bound).

Utilisation de logiciel Visual Xpress

Visual express est un outil qui permet de résoudre un problème d'optimisation linéaire. En entrée, ce solveur Visual Xpress doit recevoir un modèle mathématique et il possède trois solveurs qu'il utilise afin de résoudre les différents types de modèles :

- Solveur direct
- Solveur linéaire
- Méthode de type séparation et évaluation

Modèle Xpress et Résultats :

Le modèle Xpress sans la contrainte 4 (modèle du problème 2.2 voir chapitre 2) s'exprime simplement. Pour éviter d'avoir à saisir le distancier dans son ensemble, et comme il est symétrique (la matrice des distance est symétrique), la moitié du distancier est lue dans un fichier (distance de i à j pour laquelle $i < j$), l'autre moitié est remplie par une section *Assign*.

Chapitre 6. Approche de résolution du problème 2

The screenshot shows a software interface for solving optimization problems. On the left, the model file is displayed in a text editor:

```

MODEL Forage pétrole et gaz
LET
n=7 ! Nombre de puits
Q= 30 ! durée maximale du travail d'un appareil
Tables
D(n,n)
q(n)
DATA
D(1,1)=0, 886, 649, 757, 431, 659, 350
D(2,1)=886, 0, 768, 1079, 693, 324, 1005
D(3,1)=649, 768, 0, 416, 707, 572, 567
D(4,1)=757, 1079, 416, 0, 990, 869, 500
D(5,1)=431, 693, 707, 990, 0, 486, 659
D(6,1)=659, 324, 572, 869, 486, 0, 781
D(7,1)=350, 1005, 567, 500, 659, 781, 0
q=3, 2.5, 3, 1.5, 2, 4, 2
ASSIGN
D(i=1:n,j=1:n|D(i,j)<>0)=D(j,i)
Variables
x(n,n)
CONSTRAINTS ! Minimiser le trajet total a
Trajet : sum(i=1:n,j=1:n)D(i,j)*x(i,j)$
entree(j=1:n):sum(i=1:n|i<>j)x(i,j)=1
sortie(i=1:n):sum(j=1:n|i<>j)x(i,j)=1
BOUNDS
x(i=1:n,j=1:n).bv.
end

```

A modal window titled "View Results" is open, displaying the following information:

- Problem Statistics:** Rows: 14, Columns: 161, Non-Zeroes: 84
- Global Statistics:** Entities: 42, Sets: 0, Set Members: 0
- Solution Summary:** Iterations: N/A, Objective Value: 2920
- Status:** Global search complete - an integer solution has been found
- Tables:** D, n
- Variables:** x
- Constraints:** entree, sortie

Below the "View Results" window is another window titled "Variable" containing a matrix:

x	1	2	3	4	5	6	7
1	0	0	0	0	1	0	0
2	0	0	0	0	0	1	0
3	0	0	0	1	0	0	0
4	0	0	1	0	0	0	0
5	0	0	0	0	0	0	1
6	0	1	0	0	0	0	0
7	1	0	0	0	0	0	0

At the bottom of the "Variable" window, there is a "Display" dropdown with two options: "Primal" (selected) and "Reduced Costs".

Fig.6.6. Résultat d'exécution du programme

La résolution de ce modèle donne la solution de la figure 6.7. Cette solution contient trois sous-cycles qu'ils faut empêcher de se reformer. La stratégie à adapter consiste à supprimer un sous-cycle à la fois, en s'attaquant au plus petit d'abord. On choisir par exemple de briser le sous-tour $(2,6,2)$, en ajoutant la contrainte S1 au modèle. Cette contrainte est une traduction de la contrainte (4) pour le sous-ensemble $s = \{2, 6\}$.

$$S1: X(2,6) + X(6,2) < 1$$

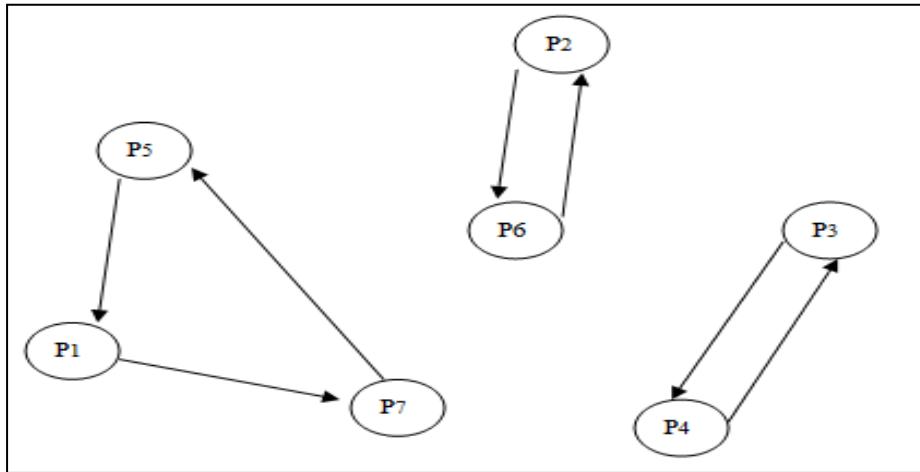


Fig.6.7. première solution avec trois sous-cycles

On résolvant le modèle Xpress modifié, on obtient la solution de la figure 6.8. Là encore trois sous -cycles sont présents. On s'attaque maintenant au sous-cycle (1,7,1) en ajoutant la contrainte S2 au modèle précédent (on pourrait aussi introduire en une seule fois toutes les contraintes interdisant la réapparition des sous-cycles formés).

$$S2 : X(1,7) + X(7,1) < 1$$

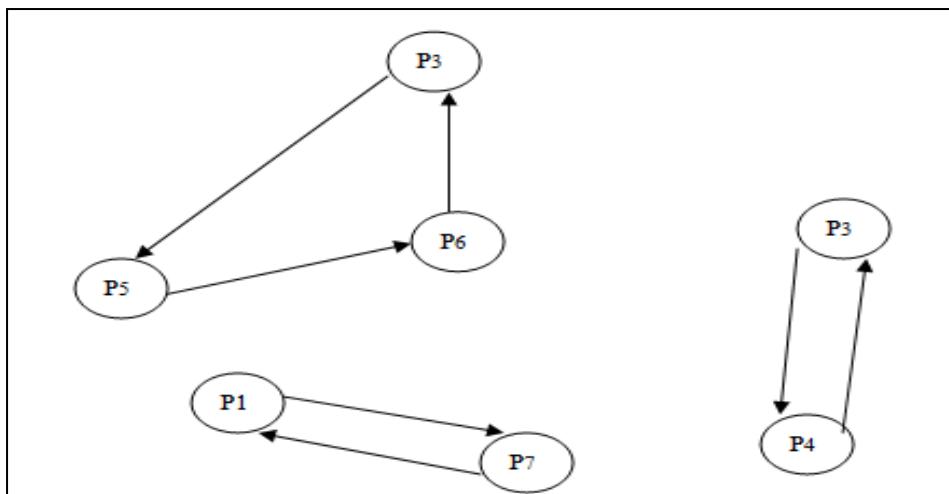


Fig.6.8. Deuxième solution avec trois sous-cycles

Si on résout de nouveau, on trouve le résultat de la figure 6.9. il n'y a plus de sous-cycles, c'est donc la solution optimale. La distance totale est de 3222 km.

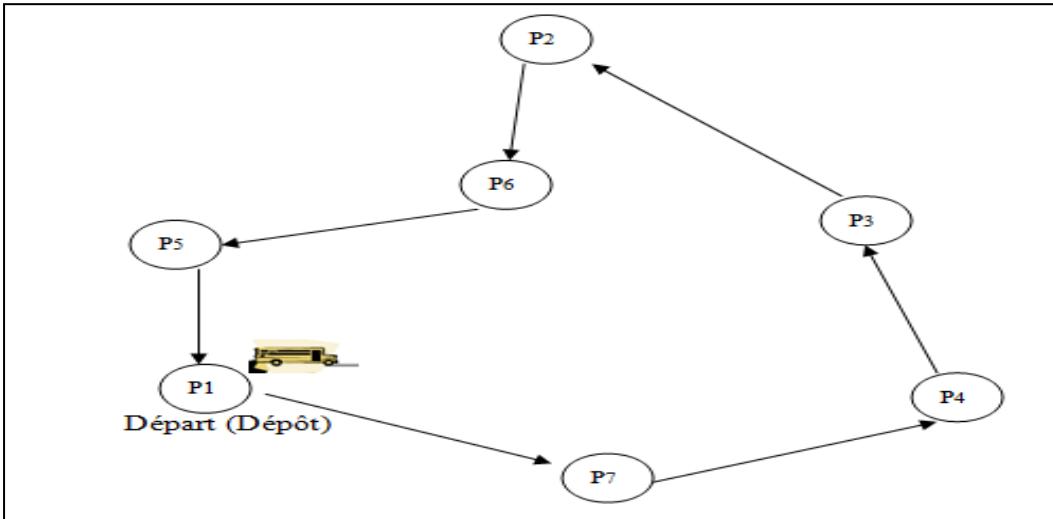


Fig.6.9. Solution optimale

6.4. Conclusion :

Ce chapitre a été consacré à la résolution du problème 2. La résolution de première sous problème 2.1 s'est faite par deux méthodes :

- La première méthode (méthode exacte) consiste en l'utilisation du logiciel LINGO14.0, qui est doté d'un solveur non linéaire pour la résolution des programmes non linéaires ; ce dernier a utilisé les algorithmes de programmation non linéaire avec contraintes pour la résolution.
- La seconde consiste à appliquer une méthode heuristique efficace.

Pour faire des méthodes que nous proposons des outils d'aide à la décision au niveau de l'entreprise, nous avons développé un logiciel nommé **Explor.1** en utilisant le langage C++. La méthode interactive que nous avons développée permet à l'utilisateur de faire des simulations des paramètres sismiques pour obtenir les résultats escomptés.

Dans le second sous problème on a opté pour une méthode de séparation et évaluation "Branch and Bound" pour sa résolution.

Conclusion

Dans ce travail nous avons essayé d'utiliser plusieurs techniques acquises durant notre formation master en recherche opérationnelle, pour cela nous avons traité trois problèmes différents.

Problème 1 « Optimisation de la taille et de la structure d'une flotte maritime pour l'exportation des produits GPL » : Se trouvant dans un environnement concurrentiel, la Société Pétrolière Sonatrach envisage l'adaptation d'une stratégie de transport maritime des GPL à la mesure de ses ambitions, à savoir : Assurer le transport de la totalité des volumes des GPL disponibles à l'exportation par ses propres moyens (CIF / CFR). Cette stratégie implique entre autre l'élaboration d'un planning d'affectation et la détermination de la taille et de la structure d'une flotte optimale, une compagnie pétrolière de grande envergure telle que Sonatrach, doit s'orienter vers l'utilisation des méthodes scientifiques basées sur des techniques de la recherche opérationnelle.

Dans ce contexte, le département des ventes GPL, branche commerciale, nous a chargé de réaliser un modèle qui minimise le coût total du transport entre autre le coût d'affrètement qui constitue 70 % de ce coût total, et ainsi réduit au mieux les besoins en navires à affréter.

Une formulation mathématique en termes de programme linéaire à variables bivalentes est faite. Cette formulation tient compte de toutes les contraintes imposées.

Quant à l'objectif, il exprime le vœu d'optimiser le coût total de l'opération de transport. Pour sa résolution, l'utilisation des méthodes exactes en un temps raisonnable s'avère prohibitive, ce qui nous a amené à adopter des méthodes approchées, d'autant plus que certaines d'entres elles ont prouvées leurs grande efficacité pour des problèmes de nature similaire. Dans notre cas (La Recherche Tabou) et (La Méthode Génétique)ont été les heuristiques utilisées à titre comparatif.

C'est dans cette optique que nous avons conçu le logiciel (Flotte GPL). Du point de vu des décideurs, le résultat obtenu dans l'optimisation des coûts de l'investissement sur l'ensemble de la flotte requise était satisfaisant, du moment que les navires propres à la Sonatrach seront utilisés sur toute la période d'étude.

Conclusion générale

L'étude qui vient ainsi de s'achever nous a permis d'atteindre les objectifs qui nous ont été assignés au début de notre travail.

Au cours de cette étude notre savoir a été enrichi en découvrant le domaine du transport maritime et celui de la commercialisation des hydrocarbures. Mais, il nous reste tant à découvrir car la recherche opérationnelle est un champ bien vaste et fascinant. Passer de la théorie à l'application est certes difficile mais passionnant.

Problème 2 « Optimisation sur les opérations d'exploration et d'exploitation du pétrole et du gaz de sonagraphe » : Pour réaliser ce projet, nous avons été amenées à traiter deux sous problèmes à savoir:

- Optimisation des déplacements des camions vibrateurs au niveau d'un chantier sismique pour l'exploration du pétrole et du gaz.
- Elaboration d'un schéma optimal de la fermeture des puits de pétrole et de gaz de Sonatrach

Le premier sous problème il s'agit de rechercher la meilleure combinaison possible des paramètres sismiques afin de satisfaire les clients de la Sonatrach à moindre coût. Pour ce faire, un déplacement sur site (Hassi Messaoud) a été nécessaire pour acquérir les différentes notions spécifiques au domaine de la géophysique afférentes à notre problème, pour se familiariser avec la terminologie d'abord qui est propre langage des géophysiciens pour bien assimiler ensuite, la problématique posée. Cette phase nous a permis de pouvoir aborder l'étape de formulation mathématique.

Le modèle mathématique obtenue est donc, un programme non linéaire à variables mixtes, positives ou nulles. Deux approches de résolution du programme sont retenues :

La première approche consiste en l'utilisation du logiciel LINGO14.0, qui est doté d'un solveur non linéaire pour la résolution des programmes non linéaires. La seconde consiste à appliquer une méthode heuristique efficace.

Pour faire des méthodes que nous proposons des outils d'aide à la décision au niveau de l'entreprise, nous avons développé un logiciel nommé **Explor 1.0** en utilisant le langage C++ La méthode interactive que nous avons développée permet à l'utilisateur de faire des simulations des paramètres sismiques pour obtenir les résultats escomptés.

Conclusion générale

Le second sous problème a pour objectif de trouver un chemin optimal de la fermeture des puits de pétrole et de gaz de sonatrach

Dans le présent travail nous avons proposé un modèle mathématique linéaire du problème de fermeture des puits de pétrole et de gaz de Sonatrach. Notre modèle est une approche qui nous mène au problème de tournées de véhicules connu sous le nom VRP.

Pour la résolution de notre modèle nous avons adopté une méthode exacte à savoir la méthode de séparation et évaluation.

Notre projet nous a permis de :

- ✓ Découvrir le domaine de la recherche des hydrocarbures en particulier celui de la sismique réflexion ainsi que le forage .
- ✓ Approfondir nos connaissances en programmation linéaire et non linéaire.
- ✓ Approfondir nos connaissances en optimisation méta heuristique et les heuristiques.
- ✓ Maîtriser les langages de programmation « Delphi » et « C++Builder » pour développer une application.

Enfin, nous espérons que l'étude effectuée présente un intérêt pour l'entreprise, et apporte un éclairage aux étudiants qui auront à préparer leur projet de fin d'études ultérieurement.

Références bibliographiques

- [01] Le revue (Documents Sonatrach) : N 1076 Mars 2013,
- [02] www.sonatrach.dz
- [03] Groupe d'expert sur les échanges et les interconnexions de gaz naturel (Janvier 2005): «Vison du marché Nord-américain du gaz naturel »
- [04] Internet. Le site <www.google.com / Panorama pétroliers>.
- [05] Slimana S. Kerbou B. Strategie du Transport Maritime des GPL, PFE USTHB, 2001. Mémoire d'ingénieur en recherche opérationnelle.
- [06] Guerrache H. Nebli F. Optimisation des expeditions du GPL par navires à partir des ports d'Arzew, PFE USTHB, 1996. Mémoire d'ingénieur en recherche opérationnelle.
- [07] L'association Canadienne Du Gaz (2003):«Comprendre le marché Nord-américain du Gaz Naturel»
- [08] <http://www.connaissancedesenergies.org/>
- [09] Jean Paul NGUYEN. Techniques d'exploitation pétrolière : LE FORAGE, Edition TECHNIP, 1993.
- [10] Document de Sonatrach, Portail Algérien des énergies renouvelables juin 2013
- [11] Jin-Kao Hao, Philippe Galinier, Michel Habib « Métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes- *LGI2P, Ecole des Mines d'Alès, Parc Scientifique Georges Besse, 30000 Nîmes – novembre 1999*
- [12] B. SADI, « Théorie des graphes – complexité algorithmique – », Office des publications universitaires.
- [13] J-F.SCHID, Introduction aux méthodes heuristiques -Graphes et RO TELECOM Nancy 2A
- [14] M. Troyon, « Quelques heuristiques et résultats asymptotiques pour trois problèmes d'optimisation combinatoire », Thèse n° 754, Lausanne, (1988).

Références bibliographiques

- [15] Pierre Fou « Méthodes heuristiques en Optimisation Combinatoire » -
Université Paris 2011-2012
- [16] Pereira Francisco B.,Tavares Jorge, Machado Penousal, costa Ernesto, GVR: A
New Genetic Representation for the Vehicule Routing Problem, url:
http://Osiris.tuwien.ac.at/~wgarn/vehiculeRouting/GECC02_VRPCoEvo.pdf
- [17] Goldberg D. Genetic Algorithm in Search, Optimisation and machine learning,
addison-Wesley,Reading Mass, 1989.
- [18] Ralph Chill- « calcul différentiel », Laboratoire de Mathématiques et
Applications de Metz Année 2010/2011.
- [19] El Amir DJEFFAL, « ETUDE DE QUELQUES ALGORITHMES DE POINTS
INTERIEURS POUR LA PROGRAMMATION CONVEXE » thèse doctorat en
mathématique appliquées – université de Batna –septembre 2013
- [20] KEBBICHE Zakia, «Etude et extensions d'algorithmes de points intérieurs pour
la programmation non linéaire », thèse Doctorat de l'université du Havre
(spécialité optimisation), 2007.
- [21] S. Bazarra, H.D. Sherali and C.M. Shetty, " Nonlinear programming, theory and
algorithms ", Second edition, 1993.
- [22] Thierno Diallo « Etude et illustration de méthodes itératives d'optimisation non
linéaire », Ecole Polytechnique Fédérale de Lausanne avril 2006.
- [23] David RIVREAU « Programmation mathématique » Institut de Mathématiques
Appliquées /Mél : david.rivreau@ima.uco.fr
- [24] Mohad OUANES – Maitre de conférence-UMMTO de Tizi ouzou « Cour de
Optimisation non linéaire » Master 2, année 2010-2011.
- [25] Christelle Guéret ,Christian Prins, Marc Sevaux. « .Programmation linéaire. 65
Problèmes d'optimisation modélisés » Editions Eyrolles 2003.
- [26] R. Ouafi « Approche algorithmique pour une classe de problème de découpe
»Thèse de doctorat. Recherche Opérationnelle. U.S.T.H.B. 2004.
- [27] Paul Feautrier- « Recherche opérationnelle » - 20 mai 2004
- [28] Pierre Lopez « cour de Graphe », LAAS-CNRS, 30 novembre 2005
- [29] A. Abbas & F. Laib, « Optimisation du coût de transport des déchets ménager »,

Références bibliographiques

- Mémoire de fin d'étude, Université de Béjaïa, Département de R.O, (1998).
- [30] Didier Maquin « Eléments de Théorie des Graphes » INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE, 3 mai 2003.
- [31] M. Minoux & A. Billionnet, « Informatique et recherche opérationnelle », Université Pierre et Marie Curie- Paris 6, (1999/2000).
- [32] Sari M., Investigation des Modèles à coûts unitaires Variables dans le cas d'un Système de Distribution à dépôt unique ou à plusieurs dépôts, Thèse Magister,Département de Génie Industrielle, E.N.P, 1997.
- [33] Philippe Iacomme, Marc sevaux – « algorithmes et graphes » - Groupe Eyrolles, 2003,
- [34] Ochi Luis S., Vianna Dalessandro S., Filho André O. and Drummond Lucia M.A., A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet. Lecture Notes in Computer Science 1391(1993), 216-224.
- [35] Manchado, P., Tavares,J-, Pereira, F. B. and Costa,E-, Vehicle Routing Problem Doing it the Evolutionary Way, GECCO-2002 Proceedings.
- [36] Fisher M.L., Jaikumer R., A generalized assignment heuristic for the vehicle routing, Network 13,109-124, 1981.
- [37] Benantar A.A, Boussaa S, Optimisation du Schéma de Distribution des Carburants, Cas :Région Est, Projet de Fin d'Etudes, Département de Recherche Opérationnelle,U.S.T.H-B, 2004.
- [38] Baker E.K & ScharfferJ.R.; 1988, Solution Improvement Heuristics for the Vehicule Routing problem and scheduling Problem with time Window constraints.
- [39] Cordeau J.F,Gendreau M., Laporte G., Potvin J-Y. & Semet F-, A Guide to Véhicule Routing problem Heuristics. *Journal of the Operational Research Society* (2002).
- [40] Gambardella Luca Maria, Metaheuristics Network, vehicle routing problems (VRP) hnp://www.idisia.ch/luca.
- [41] K. TARI, notes de cours de 3ème année R.O, « Théorie des graphes », Université de Béjaïa, 1998.
- [42] C. Berge, « Graphes et hypergraphes »Dunod Editions, Paris, Bruxelles, Montréal, (1973).

Références bibliographiques

- [43] Beghouini Mohamed, Bouderbala Abdelkader., Problème de distribution du gaz butane au niveau d'Alger, *Projet de fin d'études*, Département de Recherche Opérationnelle, U.S.T.H.B, 1996.
- [44] Bendaas A, Baghouti L, Optimisation du Schéma de Distribution des Carburants, Projet de Fin d'Etudes Département de Recherche Opérationnelle, U.S.T.H.B, 2003.
- [45] Gendreau M-, Laporte G., and Potvin J.Y., Metaheuristics for the VRP. In the vehicle routing problem, P. Toth and D- Vigo (eds), SIAM Monographs on Discrete Mathematics and Application, Philadelphia, (2002) 129-154.
- [46] A.Chaouch, J.L. Mari, 3D Land Seismic Surveys : Definition of Geophysical Parameters, dans *Oil & Gas Science and Technology*,
- [47] Institut Français du Pétrole, 2006, pp.611-630.
- [48] G. Henry, La sismique réflexion. Principes et développements, Edition Technip, mai 2000.
- [49] G. Henry, Géophysique des bassins sédimentaires, Edition Technip.
- [50] G. DEUTH & G. RISHER, « Le gros livre Borland Delphi », (1988).
- [51] C. Prins, « Algorithmes de graphes avec programmes en Pascal », Paris : Eyrolles, (1997).
- [52] F.Beaulieu, La programmation orienté objet, Simon & Schuster, 2004.

A.1. langages de programmation

A.1.2. Le langage de programmation Delphi :

Delphi est un langage de programmation orienté objet (il fait appel à une conception visuelle des applications) permettant de développer des applications pour Windows 95, 98, 2000 et Windows XP. Il représente la suite logique de la famille Turbo Pascal avec ses nombreuses versions. C'est un outil moderne, il prend en charge le maintien automatique d'une partie du code source et réduit considérablement le codage manuel par rapport au Turbo Pascal.

Delphi comporte une importante bibliothèque de composants et supporte les bibliothèques importées. Il permet aussi de concevoir, développer et manipuler des bases de données comme : Paradox7, dBase.

A.1.2. Le langage de programmation C++/ C++ Builder

Tous d'abord, C++Builder est un IDE (un environnement de développement intégré) qui regroupe tout un ensemble d'outils permettant d'effectuer un maximum de tâches de développement au sein du même environnement de travail.

C++ Builder est de plus un environnement de développement visuel C++ RAD (Rapid Application Development) permettant également le développement rapide d'applications base de données, ainsi que les application serveurs web.

C++ Builder permet le développement multi-plateformes (Windows et Linux) au moyen d'une librairie spéciale, la CLX. Si l'on ne se soucie que de Windows, la VCL est utilisée par défaut. La VCL (Visual Components Library) est une bibliothèque de composants visuels de Borland. C'est une hiérarchie de classes représentant des composants visuels qui permet de concevoir rapidement des applications. Les composants que vous voyez dans la palette de C++Builder appartiennent à la VCL. La VCL a été écrite en Pascal Object.

A.1.3. Le langage de programmation Microsoft Visual Studio

Microsoft Visual Studio est une suite de logiciels de développement pour Windows conçue par Microsoft. La dernière version s'appelle Visual Studio 2013.

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications Web ASP.NET, des Services Web XML, des applications bureautiques et des

Annexe : les langages de programmation utilisés dans ce mémoire

applications mobiles. Visual Basic, Visual C++, Visual C# et Visual J# utilisent tous le même environnement de développement intégré (IDE, Integrated Development Environment), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages.

A.2. Logiciels d'application

A.2.1. Logiciel Lingo :

LINGO est un outil complet conçu pour formuler rapidement, facilement et efficacement les problèmes d'optimisation de modèles linéaires, non linéaires, quadratiques, de cônes du second degré et stochastiques. LINGO met à votre disposition : un langage puissant et un environnement complet pour construire et éditer vos modèles, le tout complété d'un jeu de solveurs ultra-performants

Des solveurs ultra-performants

LINGO est livré avec un jeu de solveurs pour l'optimisation linéaire, non-linéaire (convexe ou non convexe), quadratique, sous contraintes, et en nombre entier.

Documentation et aide exhaustive

LINGO est livré avec une documentation et une aide en ligne exhaustives : description détaillée de toutes les commandes et fonctions, jeu d'exemples réutilisables et modifiables. Les versions "Super" et "Larger" sont livrées avec l'ouvrage de référence "Optimization Modeling with LINGO", détaillant et expliquant les principales classes de problèmes d'optimisation linéaire, non-linéaire et en nombre entiers.

A.2.2. Logiciel Visual xpress

Visual express est un outil qui permet de résoudre un problème d'optimisation linéaire. En entrée, ce solveur Visual Xpress doit recevoir un modèle mathématique et il possède trois solveurs qu'il utilise afin de résoudre les différents types de modèles :

- Solveur direct
- Solveur linéaire
- Méthode de type séparation et évaluation