



République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Hadj-Lakhdhar – Batna
Faculté de technologie
Département de Génie Industriel



Mémoire

Présenté pour obtenir le diplôme de
Magister en Génie Industriel et productique

Par : LAGGOUN ASSIA

THEME

DEVELOPPEMENT D'UNE APPROCHE POUR LA
RESOLUTION D'UN PROBLEME DE LOT SIZING
AVEC TRANSPORT

Encadreur :

Dr Nadia Kinza Mouss

Co-encadreur :

Mr Ahmed Senoussi

Devant le jury :

Pr Mouss Hayet Leila	Professeur	Univ Hadj Lakhdar Batna	Président
Dr Mouss Nadia Kinza	MCA	Univ Hadj Lakhdar Batna	Rapporteur
Pr Kezar Okba	Professeur	Univ Mohamed khaidhar Biskra	Examineur
Dr Abdelhamid Samir	MCA	Univ Hadj Lakhdar Batna	Examineur

2013

Remerciements

Au nom de dieu je commence mes remerciements.

Je tiens avant tout à exprimer ma profonde gratitude et respect à mon encadreur, Dr MOUSS NADIA KINZA pour sa confiance, pour sa patience, ses conseils, ses encouragements, et sa gentillesse. Je remercie également mon co-encadreur monsieur AHMED SENOUSSE pour les orientations et l'aide qu'il m'a porté.

Je suis très reconnaissante envers professeur MOUSS LEILA HAYET directrice du Laboratoire d'Automatique et de Productique et Dr MOUSS MOHAMED DJAMEL chef de département Génie Industrie, dont j'admire le calme, la gentillesse, les compétences, et leur bonne gestion. Je suis fière d'être une étudiante de ce département. Un grand merci aussi à Mr KANIT ABDELGHAFOR, responsable de la post graduation, et Mr MARREF TAREK responsable de la pédagogie.

Sans oublier tous les enseignants spécialement mes enseignants de l'année théorique ; je les remercie pour leurs cours, leurs gentillesse et leurs conseils. Je remercie également tous les étudiants du Département de Génie Industriel.

Enfin merci de tout mon cœur à toute ma famille. Sans oublier Mes amis et mes collègues du centre de la formation professionnelle BATNA 1.

Au nom de dieu je termine mes remerciements.

*LAGGOUN ASSIA
A BATNA JUIN 2013*

DEVELOPPEMENT D'UNE APPROCHE POUR LA RESOLUTION D'UN PROBLEME DE LOT SIZING AVEC TRANSPORT

Résumé :

Le travail présenté dans le cadre de ce mémoire est relatif à un problème de lot sizing. L'objectif consiste à optimiser le coût total de la chaîne, constituée d'un coût de production, de stockage et de transport. Le problème a été d'abord résolu par une méthode exacte ; qui a montrée ses limites si le nombre de client et la période augmentent. Aussi nous avons proposé l'utilisation des algorithmes génétiques comme heuristique à la solution de ce problème. Les résultats obtenus sont satisfaisants.

Mots clefs : Lot sizing, Chaîne logistique, Algorithme génétique.

Development Of An Approach To Solve A Lot Sizing Problem With Transport

Abstract :

The work presented is about a lot sizing problem. The objective is to optimize the total cost of the supply chain, consisting of a production cost, storage cost and transport cost. First the problem was solved by an exact method, which has shown its limits if the number of customers and periods increase. Also we proposed the use of genetic algorithms as a heuristic to solve this problem. The results are satisfactory.

Key words : Lot Sizing, Supply chain, Genetic Algorithm.

تطوير منهجية من اجل حل مسألة تحجيم دفعات الإنتاج مع النقل

ملخص :

العمل المقدم في هذه المذكرة حول مسألة تحجيم دفعات الإنتاج. الهدف هو تحسين التكلفة الإجمالية للسلسلة اللوجيستية ، والتي تتألف من تكلفة الإنتاج والتخزين والنقل. في البداية تم حل المشكلة بواسطة طريقة دقيقة ، التي أظهرت حدودها إذا زاد عدد العملاء و فترات الإنتاج. كما اقترحنا استخدام الخوارزميات الجينية كطريقة مقربة ومهجنة لحل هذه المسألة. وكانت النتائج مرضية.

الكلمات الدلالية: تحجيم دفعات الإنتاج . سلسلة لوجيستية . خوارزمية جينية.

Sommaire

Liste des abréviations	I
Liste des figures	II
Liste des tableaux	III
Introduction générale	IV

Chapitre 1 : Concepts de base du problème de lot Sizing avec transport

Introduction	2
1.1. La chaîne logistique (Supply Chain)	2
1.2. Généralités sur la production et le problème de lot sizing	3
1.2.1. Gestion de la production	3
1.2.2. Niveau de décision	5
1.2.3. Planification de la production	5
1.3. Présentation du problème de lot sizing	6
1.3.1. Classification des critères de lot sizing	7
1.3.2. Classification du problème de lot sizing	8
1.3.3. Méthodes de résolution du problème de lot sizing	11
1.4. Formulation mathématique du problème de lot sizing	14
Conclusion	15

Chapitre 2 : Problème du lot Sizing avec transport

Introduction	17
2.1 Description du problème	17
2.2 Notations	18
2.2.1 Unité de production	18
2.2.2 Dépôt central	19
2.2.3 Transport	19
2.2.4 Stock client	19
2.2.5 Les indices	19
2.2.6 Les ensembles	19
2.2.7 Les variables de décision	19
2.3 Modèle mathématique en PLNE	20
2.3.1 Coût de production	20
2.3.2 Coût de stockage du DC	20
2.3.3 Coût de stockage du client	20
2.3.4 Coût de transport	21
2.4. Résolution du problème par GAMS-CPLEX	22

2.4.1. Description des instances.....	22
2.4.2. Tableau des instances et des résultats.....	23
2.4.3. Interprétation des résultats.....	24
Conclusion.....	26
Chapitre 3 : Résolution du problème par les algorithmes génétiques.	
Introduction	28
3.1. Principe des algorithmes génétiques.....	28
3.1.1. Concepts des algorithmes génétiques.....	28
3.1.2. Opérations des algorithmes génétiques.....	30
3.1.3. Structure générale de l’algorithme génétique	33
3.2. Application des algorithmes génétiques pour la résolution du problème de lot sizing.....	34
3.2.1. Codage du chromosome.....	35
3.2.2. Principe du zéro Switch.....	36
3.2.3. Les algorithmes de réparation.....	38
3.2.4. Algorithme génétique multi chromosome.....	46
3.3. Implémentation de l’algorithme génétique intégral.....	47
3.3.1. Résultats numériques.....	47
3.3.2. Comparaison de l’heuristique avec GAMS.....	49
3.3.3. Interprétation des résultats.....	50
Conclusion.....	51
Conclusion générale.....	53
Références bibliographiques	55

Liste des abréviations

Lot Sizing	Dimensionnement des lots de production
PLNE	Programmation Linéaire en Nombre Entier
GP	Gestion de Production
LSP	Problème de lot sizing
LSP-1P	LSP à un produit
LSP-1PC	LSP à un produit avec contrainte de capacité
LSP-1PS	LSP à un produit sans contrainte capacité
LSP-FT	LSP avec contrainte des livraisons dans une fenêtre de temps.
GA	Algorithme génétique
VRP	Problème de tourné de véhicule
GAMS	General Algebraic Modeling System
TSP	Problème du voyageur de commerce
GUI	Interface graphique

Liste des figures

Figure.1.1 Représentation de la chaine logistique	2
Figure 1.2. Les éléments de la gestion de production	3
Figure 1.3. Niveaux décisionnels dans une chaine logistique	5
Figure.1.4. Classification des LSP basé sur le temps	9
Figure.1.5. Classification des LSP basé sur les niveaux de produit	10
figure.1.6. Classification des méthodes de résolution.	11
Figure 1.7. La structure du modèle de WAGNER et WITHIN	14
Figure.2.1. Schéma mono usine mono dépôt et multi client et stock.	18
Figure .2.2 Graphe du temps de calcul en fonction du nombre de véhicule, de client et de période.	25
Figure 2.3. Graphe du temps de calcul en fonction de la capacité de production pour $(n_v, n_i, n_t) = (10, 20, 10)$.	26
Figure.3.1. Les cinq niveaux d'organisation dans les algorithmes génétiques.	29
Figure.3.2. Codage des variables	29
Figure 3.3. Présentation de la population à une génération i .	30
Figure .3.4. Les opérations des algorithmes génétiques	30
Figure.3.5. Roulette de Wheel.	31
Figure.3.6. Opération de Croisement à un point.	32
Figure.3.7. Opération de Croisement multipoint	32
Figure.3.8. Opération de mutation.	33
Figure .3.9. Décomposition du système étudié en 3 sous structures.	34
Figure 3.10. Organigramme de l'algorithme génétique	35
Figure .3.11. Structure du multi chromosome.	35
Figure.3.12. Principe du zéro Switch.	37
Figure .3.13. Calcul de la quantité à livrer sans réparation.	38
Figure .3.14. Calcul de la quantité à livrer avec réparation.	39
Figure .3.15. Affectation des véhicules aux différents clients	41
Figure.3.16. Affectation des véhicules aux clients à un instant t	41
Figure.3.17. Interface de l'algorithme génétique pour le problème de lot sizing avec transport.	48
Figure.3.18. Courbe du coût en fonction des générations	47
Figure 3.19. Graphe représentant le temps de calcul des deux méthodes en fonction du nombre des clients.	51
Figure 3.20. Graphe représentant le temps de calcul des deux méthodes en fonction du nombre des périodes	51

Liste des tableaux

Tableau 2.1. Configuration des paramètres (V, I, T, P, Pu, Tu, Su).	23
Tableau 2.2. Différentes structures des coûts.	23
Tableau 2.3. Les instances et les résultats du solveur CPLEX.	24
Tableau 2.4. Les instances pour $(nv, ni, nt) = (10, 20, 10)$.	25
Tableau 3.1. Calcul du coût pondéré	42
Tableau.3.2. Tri du CP et du ql	43
Tableau .3.3. Affectation de la plus petite valeur du ql à la plus petite valeur de CP.	43
Tableau .3.4. Affectation de la plus grande valeur du ql à la plus petite valeur de CP.	44
Tableau .3.5. Affectation de la plus petite valeur du ql à la plus grande valeur de CP.	44
Tableau .3.6. Affectation de la plus grande valeur du ql à la plus grande valeur de CP.	45
Tableau.3.7. Quantité transporté.	45
Tableau.3.8. Propriétés de l'algorithme génétique	49
Tableau.3.9. Comparaison entre la méthode heuristique et la méthode exacte.	50

Introduction générale

La forte concurrence sur le marché, exige une focalisation sur la chaîne logistique. Son informatisation pour une optimisation du système s'est développée avec l'avènement des nouvelles technologies. L'optimisation de la chaîne logistique vise donc à réduire au mieux, les délais et les coûts engendrés entre le fournisseur et le client. Une chaîne logistique optimisée permet de réduire le nombre d'opérations, de diminuer les coûts et d'améliorer la productivité tout en assurant une qualité de service optimale pour le client final.

Notre étude se situe dans le domaine de la recherche opérationnelle. Au cours de cette recherche nous allons résoudre le problème de lot sizing avec transport, notre objectif consiste à optimiser le coût total de la chaîne constitué d'un coût de production, coût de stockage et coût de transport.

Ce mémoire est constitué de trois chapitres. Dans le premier chapitre, on présente quelques définitions des concepts de base sur le problème de lot sizing, en commençant par la chaîne logistique et la gestion de la production, passant par les classifications et les méthodes de résolution du problème de lot sizing. On termine avec une formulation mathématique du problème de lot sizing de base.

Le second chapitre présente la modélisation de la structure étudiée par PLNE et la proposition d'une solution à l'aide du solveur CPLEX.

Le dernier chapitre contient la description de la résolution du problème par un algorithme génétique. Aussi on compare les résultats obtenus avec la méthode exacte du chapitre précédent.

On termine par une conclusion générale et des perspectives.

Chapitre 1 :

Concepts de base du problème de lot sizing avec transport.

Dans ce chapitre nous présentons les concepts de base du problème de lot sizing avec transport. Nous allons commencer par des définitions sur la chaîne logistique et la gestion de production, le fonctionnement d'un système de gestion de production en faisant un focus sur le problème de lot sizing. Nous citons des classifications de problème de lot sizing et les méthodes de résolutions. Nous donnons la formulation mathématique du modèle de Wagner et nous terminons par une conclusion.

Introduction

La maîtrise de la chaîne logistique est un des enjeux clés pour les entreprises. En pleine mutation, suite aux développements technologiques, qui ont amélioré la façon de travailler, et à la mondialisation, ce domaine connaît une forte croissance. La logistique consiste à concevoir, piloter et gérer l'acheminement optimal des produits depuis les lieux d'approvisionnement à ceux de leur distribution, en passant par les échanges de production.[1]

Si un client a besoin d'une quantité bien déterminée d'un produit, et dans des dates bien déterminées sur un horizon de temps fini, le producteur doit respecter ces dates, et les conditions imposées par le client. Une synchronisation doit se faire tout au long de la chaîne logistique de la production au transport.

Dans ce chapitre nous allons présenter les concepts de base de la chaîne logistique et plus précisément le lot sizing.

1.1. Chaîne logistique (Supply Chain) :

C'est un réseau d'organisations et de processus d'affaires permettant : l'approvisionnement en matières premières, la fabrication et la distribution des produits aux clients. Les flux physiques, d'information et financier se déplacent sur la chaîne logistique dans les deux sens [2]. La chaîne logistique est constituée de fournisseurs, producteurs et clients avec des routes de transport et des dépôts de stockage, elle est donc formé de : zone de production, zone de commercialisation, route de distribution (figure1.1):

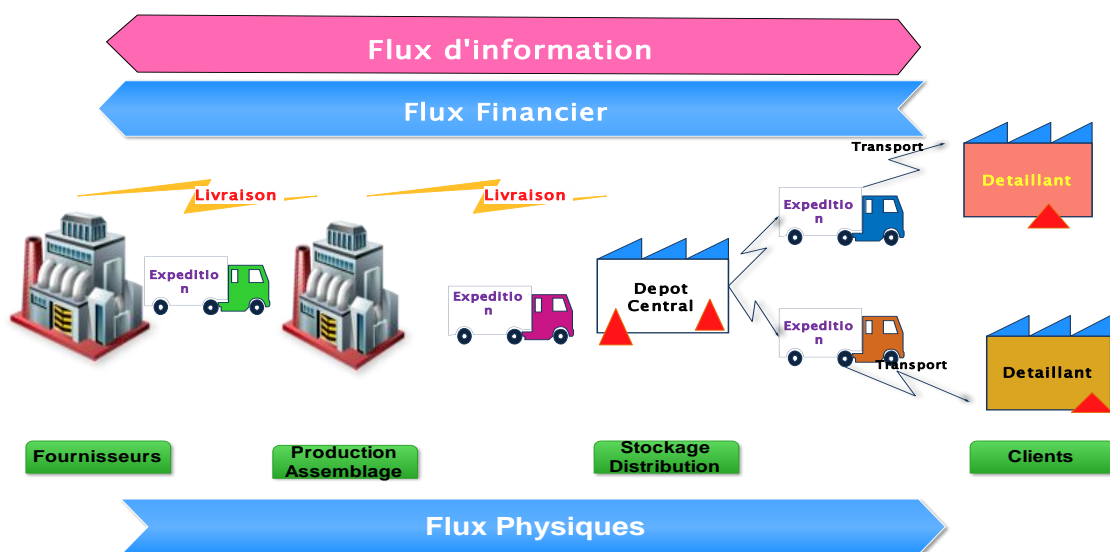


Figure.1.1 Représentation de la chaîne logistique

Pour gérer la chaîne logistique, il faut gérer les flux en optimisant autant que possible les coûts, qui sont principalement [3] : les coûts de conception des produits, les coûts d'approvisionnement, les coûts de production, les coûts d'immobilisation financière et de possession des stocks, les coûts d'acheminement ou de transfert, les coûts de rupture et les coûts d'assurance...etc.

1.2. Généralités sur la production et le problème de lot sizing :

1.2.1. Gestion de la production (GP):

La gestion de la production est la synchronisation de l'ensemble des actions de production agissant sur un flux de matière qui traversent l'entreprise, en tenant compte des contraintes et des critères de performance, à partir de ressources physiques, humaines et financières [4]. La figure 1.2 représente les fonctions de la gestion de la production.

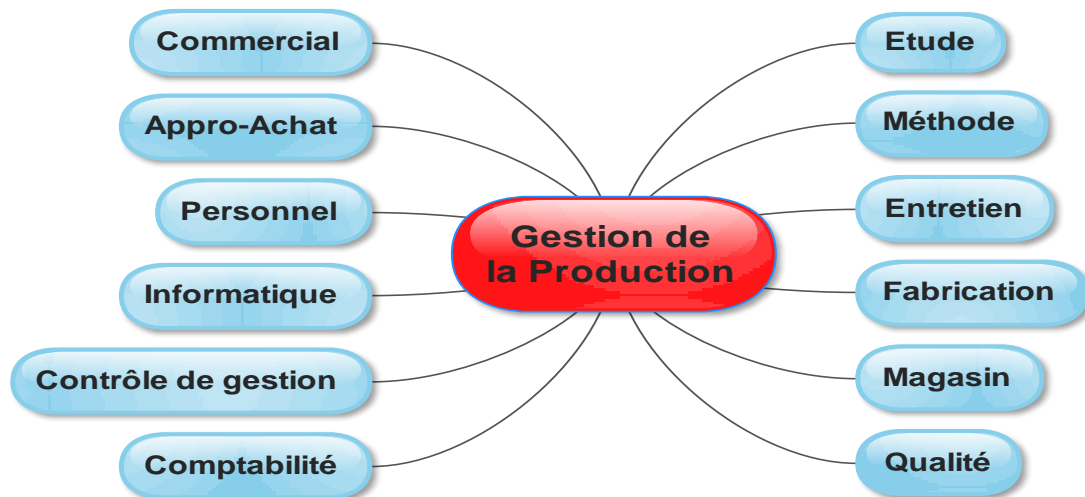


Figure 1.2. Les éléments de la gestion de production [4].

1.2.1.1. La production :

Le service de production assure la transformation respectivement de la matière première ou des produits semi finis en produits finis ou produits semi finis. Trois critères doivent être vérifiés, qualité coût et délais.

Afin d'arriver à réaliser et concrétiser le maximum de condition de production et de satisfaire les commandes clients, et à moindre coût, il faut prévoir une bonne planification de la production.

1.2.1.2. Le transport :

Selon [5], dans une chaîne logistique, le transport en général est celui des marchandises, c'est un domaine très vaste où l'optimisation n'est jamais aisée. Il constitue une phase très importante du processus logistique, il est l'interface entre la phase de production et le marché.

On classe les problèmes de transport en deux types :

- Les problèmes de tourné de véhicule : On peut citer le problème du voyageur de commerce (TSP), et le Vehicle Routing Problem (VRP).
- Les problèmes modélisables par flux : Consiste à modéliser l'approvisionnement en produit depuis un ensemble de fournisseurs vers un ensemble de clients.

1.2.1.3 Le stockage :

Le stock dans un système de production a pour but de compenser les écarts entre la prévision et la réalité, minimiser les délais d'attente chez le client, éviter la hausse des prix de la matière première et éviter l'accident de retard de production. Ainsi quatre types de stocks sont définis [5]:

- Stock de marchandise
- Stock de matière première
- Stock de produit en cours de fabrication
- Stock de produit fini.

La classification ABC est l'une des méthodes les plus répandues pour la gestion des stocks, elle consiste à établir une hiérarchie de produit en trois classes.

- Classe A : produits très chers et rares qui nécessitent un délai de livraison très long.
- Classe B : produits moyennement chers.
- Classe C : produits courants ou peu chers.

L'optimisation des stocks est un problème important dans la chaîne logistique à cause des coûts de stockage (espace + argent + organisation + recherche + charge et décharge + classification et tri ...).

Les fonctions de production, stockage et de transport sont étroitement liées. Une approche d'optimisation intégrée consiste à lier au moins deux fonctions en même temps par exemple :

- Production-transport- stock.

- Production-transport.
- Production-stock

1.2.2 Niveaux de décision

L'optimisation d'un système de production consiste en une bonne prise de décision. Les différentes décisions prises au sein d'un système de production ont fait l'objet de plusieurs essais d'évaluation et de classification, la plus traditionnelle est la classification par niveau, [1] :

On distingue trois niveaux de décisions dans une entreprise (figure 1.3) :

1.2.2.1 Niveau stratégique

Appelé également décisions à long terme (l'horizon porté sur plusieurs années). On peut citer comme exemple l'installation d'une nouvelle usine.

1.2.2.2 Niveau tactique

Les décisions sont à moyen terme (l'horizon de plusieurs mois), on peut citer à titre d'exemple, la planification de la production, c'est-à-dire le calcul du plan de la production pour 6 à 8 mois.

1.2.2.2 Niveau opérationnel

Le terme opérationnel veut dire gérer le fonctionnement quotidien de la production, parmi ces décisions on trouve : la gestion de stock, l'ordonnancement, le plan de transport ... etc

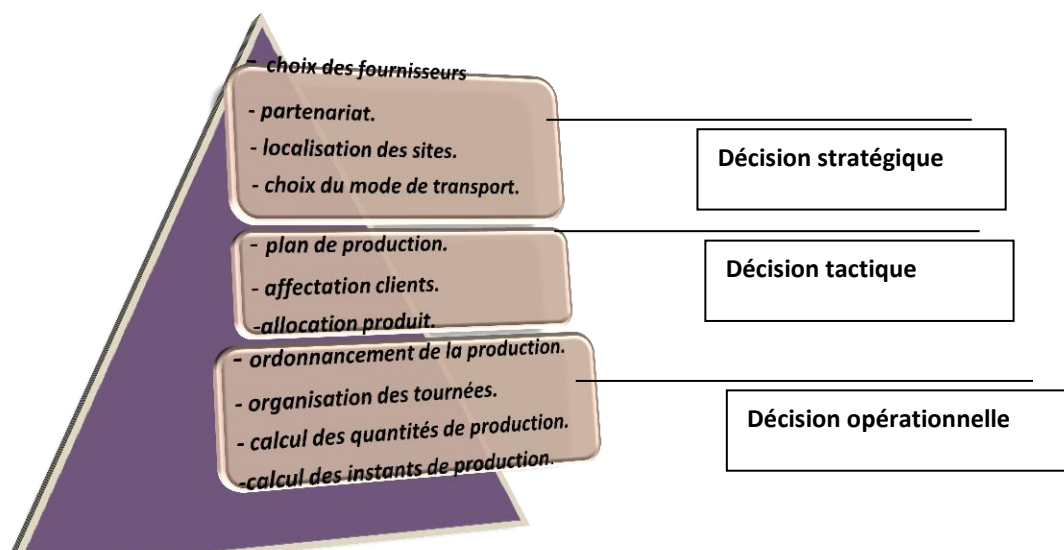


Figure 1.3. Niveaux décisionnels dans une chaîne logistique [12].

L'objectif majeur de cette classification est de simplifier le processus de décision. Et afin d'arriver à cet objectif, il faut s'assurer de la cohérence des décisions avec les contraintes du niveau supérieur.

1.2.3 Planification de la production

La planification se fait au niveau tactique de la gestion de production, elle constitue une passerelle entre les décisions stratégiques et opérationnelles. La règle est : l'utilisation optimale des moyens de production, autrement dit satisfaire les demandes à moindre coût.

Le problème de planification dans une chaîne logistique est la détermination de la circulation de la marchandise depuis le fournisseur en passant par la production et arrivant au client. Le but est d'élaborer un plan de production qui permet d'assurer la coordination des données dans les différentes phases de la chaîne logistique. (Acheter, produire, vendre).

Dans notre cas on s'intéresse au problème de détermination du lot de production (lot sizing) à moyen et court terme en tenant compte du transport du producteur vers les clients.

1.3 Présentation du problème de lot sizing :

Le Lot Sizing Problem (LSP) en terminologie anglaise, est un problème de planification de la production où la demande en produits varie à travers un horizon fini T. L'objectif est de déterminer les périodes de production à travers cet horizon ainsi que les quantités de produits à fabriquer durant chacune des périodes, sachant que chaque série de production entraîne un coût de réglage supplémentaire. La quantité totale de production doit satisfaire la demande du client tout en minimisant le coût global de la chaîne [5].

La vision de la planification de la production d'un point de vue logistique dépend de plusieurs paramètres. Nous présentons dans ce qui suit un certain nombre de questions et en fonction de la réponse donnée le plan de planification est élaboré.

1. Produisons nous plusieurs types de produits ou un seul?
2. L'approvisionnement se fait par un ou plusieurs fournisseurs?
3. Existe-t-il une ou plusieurs zones de productions?
4. Existe-t-il une ou plusieurs zones de stockage?
5. Existe-t-il un ou plusieurs chemins?
6. Existe-t-il un ou plusieurs véhicules?

7. Tous les véhicules ont-ils la même capacité (homogène ou non)?
8. Chargeons nous un type ou plusieurs types de produit dans un véhicule?
9. Le véhicule doit être plein ou semi plein?
10. Un véhicule doit faire une ou plusieurs tournées?
11. La capacité de stockage est limitée ou non?
12. Le coût du stockage est le même pour toutes les quantités du même produit?
13. La commercialisation se fait elle directement après la production sans passer par le stockage?

Les premiers travaux sur le PLS sont ceux de Wagner et Within [6] en 1958, depuis, beaucoup de recherches ont été menées sur les PLS. L'objectif est d'arriver à calculer une taille de production minimale.

1.3.1 Classification des critères du lot sizing :

Cathy Wolosewicz [4] et Nadjib Brahim [7] ont cité dans leurs travaux un certain nombre de critères de lot sizing, nous citons :

1.3.1.1. L'échelle de temps :

La planification se fait sur des périodes discrètes ou sur un horizon continu de temps, on trouve trois modes

- a) Période de petite taille : (Small time buckets) 1 heure, 1 jour.
- b) Période de grande taille : (Big time buckets) 2 jours, 2 semaines.
- c) Période de très grande taille : (very big time buckets) 2 mois, ou 1 trimestre.

1.3.1.2 Le nombre de niveau :

Si les demandes émanent de l'extérieur de l'entreprise on parle d'un seul niveau, par contre si des demandes émanent de l'entreprise elle-même, dans le cas d'un produit composé de plusieurs produits, on parle de plusieurs niveaux [4].

1.3.1.3 Le nombre de produit :

Le nombre de produit dépend du nombre de niveaux. Pour un niveau on a un produit sinon on a plusieurs produits (la nomenclature).

1.3.1.4 La capacité :

Le terme capacité est utilisé pour plusieurs grandeurs comme par exemple : capacité des employés, capacité de stockage, capacité de produire.

1.3.1.5 les demandes :

On trouve 3 groupes de demandes :

Groupe 1 :

1. Demande constante : les demandes ne changent pas sur l'horizon
2. Demande dynamique : les valeurs varient au cours du temps

Groupe 2 :

1. Demande certaines : les valeurs sont connues à l'avance
2. Demande stochastique : basé sur les probabilités

Groupe 3 :

1. Demande indépendante : pas besoin d'autres produits
2. Demande dépendante : un produit dépend d'autres produits.

1.3.1.6 Les coûts de lancement (setup)

C'est le coût de préparation des machines pour un chargement de produit. Ce chargement est dans la plupart des cas coûteux et long.

1.3.2 Classification du problème de lot sizing :

BRAHIMI [7] a introduit un ensemble d'abréviations pour faciliter la lecture des problèmes de lot sizing tel que :

- **LSP** : problème lot sizing.
- **LSP-IP** : LSP à un produit
- **LSP-IPS** : LSP à un produit sans contrainte de capacité
- **LSP-IPC** : LSP à un produit avec contrainte de capacité
- **LSP-FT** : LSP à un produit avec contrainte des livraisons dans une fenêtre de temps.

Les classifications des LSP diffèrent dans la littérature d'un auteur à un autre et d'un champ d'exécution à un autre ; [5] présente deux classifications, la première se base sur le critère de temps (figure1.4), la deuxième sur le niveau du produit (figure1.5).

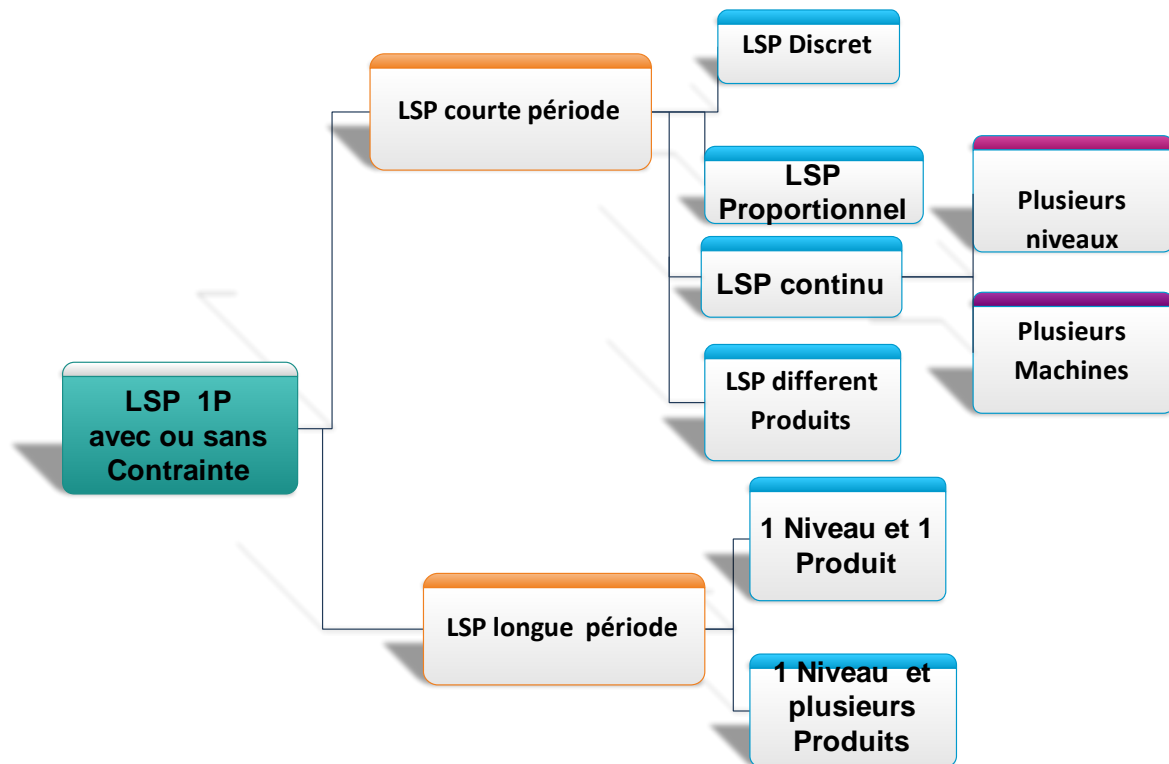


Figure.1.4. Classification des LSP basée sur le temps [5]

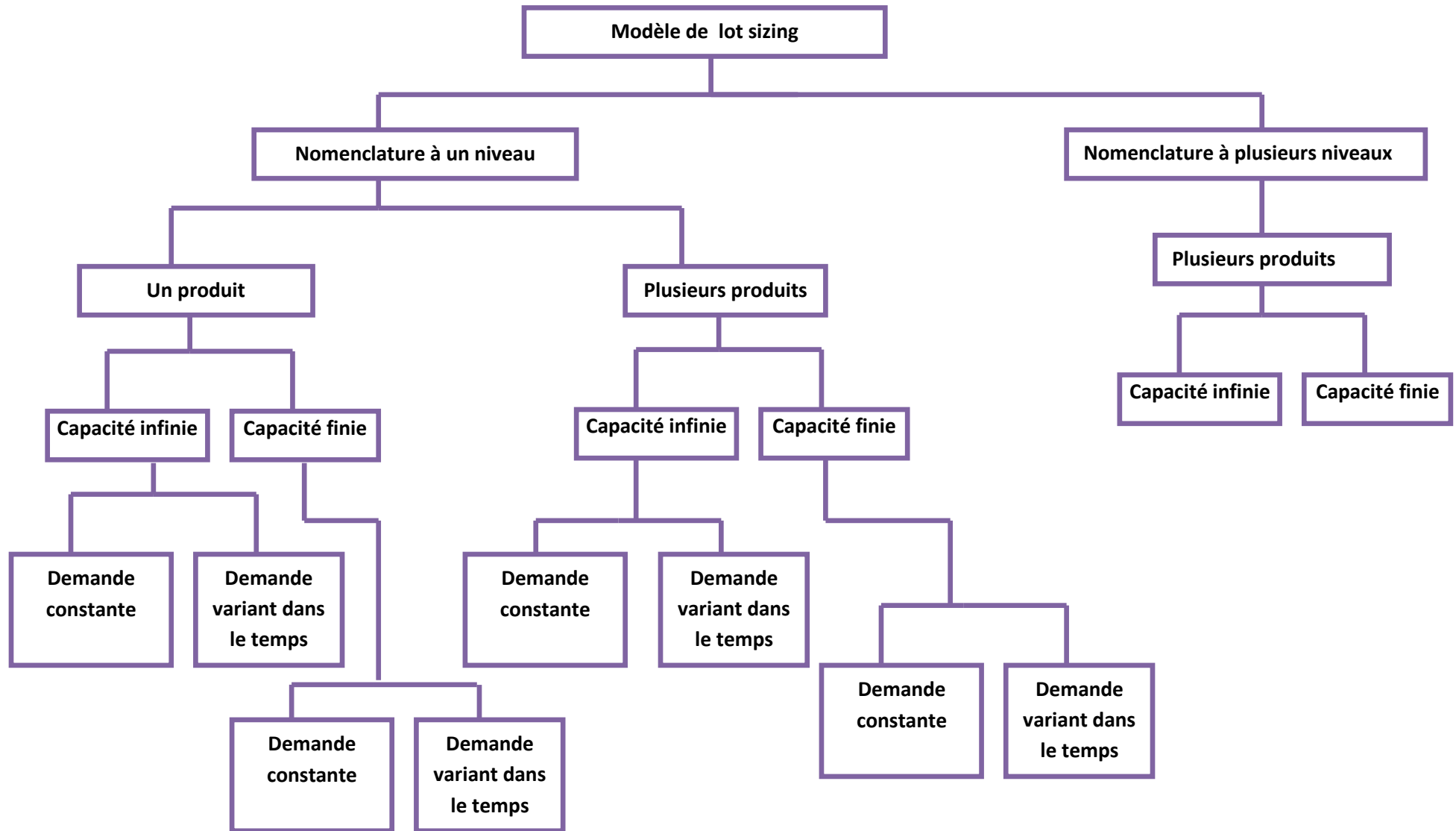


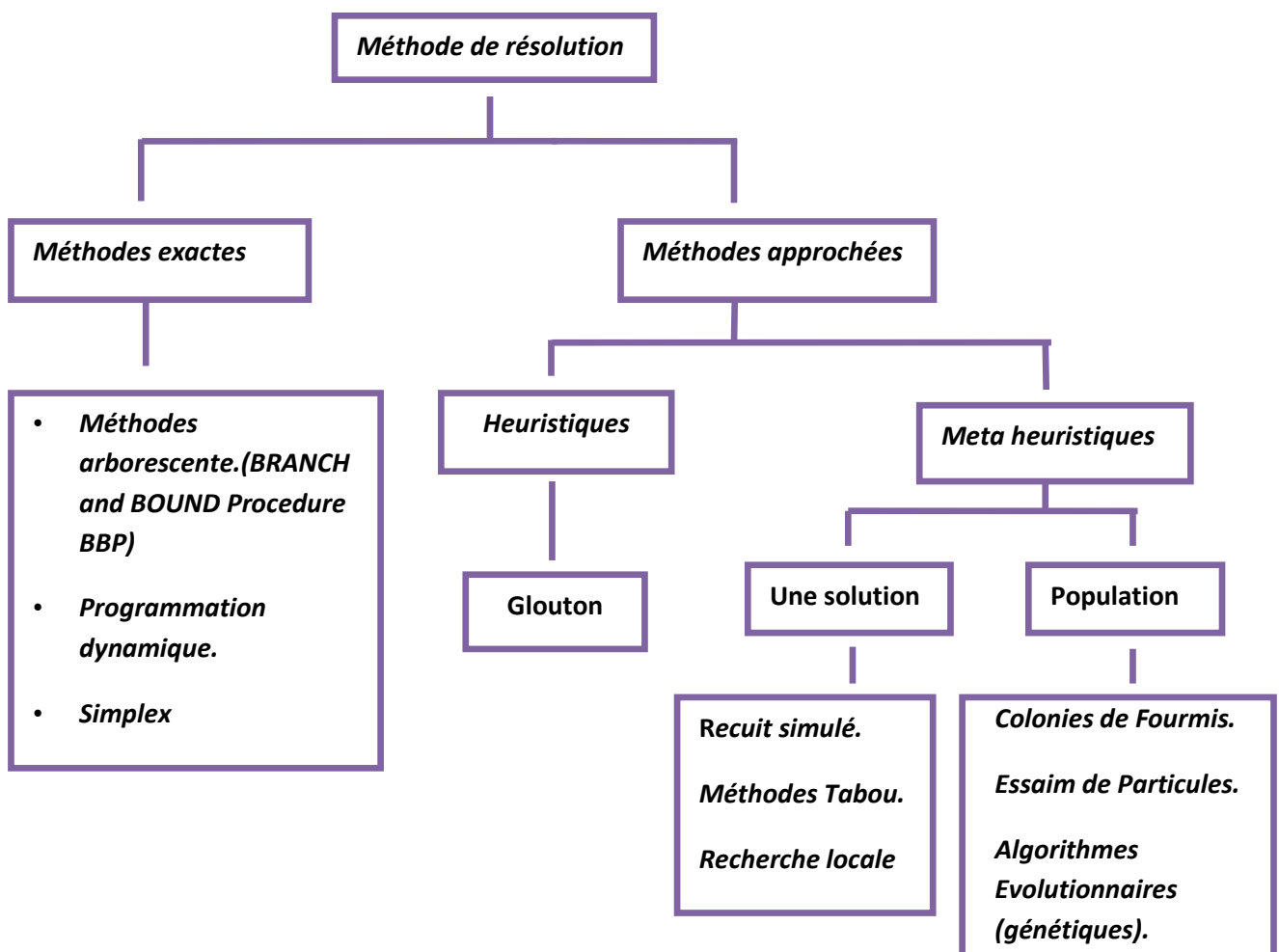
Figure.1.5. Classification des LSP basée sur les niveaux de produit [5]

1.3.3 Méthodes de résolution du problème de lot sizing:

Pour résoudre les problèmes de lot sizing, plusieurs techniques ont été développées

- les méthodes exactes qui recherchent une solution optimale d'une manière certaine.
- les méthodes approchées qui se contentent de trouver une bonne solution au problème. Ces méthodes s'appliquent si les méthodes exactes sont impossibles à utiliser. Elles assurent un compromis entre la qualité de la solution retenue et le temps nécessaire de calcul.

La figure 3.7 résume les différentes méthodes utilisées.



La figure.1.6. Classification des méthodes de résolution du LSP [5].

1.3.3.1 Méthodes exactes :

Les méthodes exactes fournissent systématiquement une solution (optimale) au problème traité si une telle solution existe. Dans le cas contraire, ce type de méthode permet d'affirmer qu'il n'existe pas de solution au problème traité. On trouve (voir figure 1.6)

Dans la pratique, certains problèmes combinatoires sont considérés comme des problèmes difficiles, ou non traitables (intractable problems). En termes de complexité, ils sont généralement des problèmes NP-complets ou NP-durs (NP-hard) [8].

Une alternative qui a fait ses preuves en pratique pour de nombreux problèmes difficiles de grande taille est l'utilisation d'algorithmes de recherche heuristique.

1.3.3.2 Méthodes approchées :

Les méthodes approchées fournissent une solution approchée au problème traité. Elles sont en général conçues de manière à ce que la solution obtenue puisse être proche de la valeur optimale : de telle méthodes permettent d'obtenir des bornes inférieures ou supérieures de la valeur optimale tel que :

- Méthodes Heuristiques ;
- Méthodes Méta heuristiques.

1.3.3.2.1 Heuristiques :

Du grec *heuriskein* : trouver/découvrir (heureka). Une heuristique est plutôt une méthode qui cherche (une stratégie) sans garantir le résultat. Destiné à un problème spécifique, le temps de calcul est raisonnable sans garantir la faisabilité ou l'optimalité [2,10].

En recherche opérationnelle, les heuristiques sont des règles empiriques simples qui ne sont pas basées sur l'analyse scientifique (différents algorithmes). 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 [9], parmi les heuristiques on peut citer l'algorithme glouton).

Algorithme glouton : Les algorithmes gloutons sont ceux qui trouvent une solution parmi plusieurs possibles, comparables selon un critère global. A chaque fin d'étape de l'algorithme, on choisit l'étape suivante en fonction des informations locales. La méthode gloutonne ne trouve pas forcément la meilleure solution [9]. L'idée générale est :

- 1) Générer une solution initiale
- 2) Modifier légèrement la structure de cette solution pour en obtenir une meilleure
- 3) Répéter, tant qu'on trouve des améliorations

1.3.3.2 Meta heuristiques :

Méta est un préfixe qui provient du grec *μετά* (*meta*) (après, au-delà de, avec). Il exprime, tout à la fois, la réflexion, le changement, la succession, le fait d'aller au-delà, à côté de, entre ou avec [9]. **Méta** signifie aussi à propos.

Ceux sont des algorithmes d'optimisation (généralement de type stochastique) combinant plusieurs approches heuristiques. Elles tentent de trouver l'optimum global (plusieurs solutions) ou l'optimum local (une seule solution) d'un problème d'optimisation difficile.

1.4.2.1. Meta heuristique à une solution :

a. Recuit simulé (Simulated Annealing):

L'idée est d'effectuer un mouvement selon une distribution de probabilité qui dépend de la qualité des différents voisins [9]:

- Les meilleurs voisins ont une probabilité plus élevée.
- Les moins bons ont une probabilité plus faible.

On utilise un paramètre, appelé la température (notée T) :

- T élevée : tous les voisins ont à peu près la même probabilité d'être acceptés.
- T faible : un mouvement qui dégrade la fonction de coût a une faible probabilité d'être choisi.
- $T=0$: aucune dégradation de la fonction de coût n'est acceptée.

La température varie au cours de la recherche : T est élevée au début, puis diminue et finit par tendre vers 0.

b. Recherche Tabou (Tabu Search):

L'idée de base de la liste tabou consiste à mémoriser les configurations ou régions visitées et à introduire des mécanismes permettant d'interdire à la recherche de retourner trop rapidement vers ces configurations [9].

c. Recherche locale (Local Search):

La recherche locale peut être utilisée sur des problèmes de recherche d'une solution maximisant un critère parmi un ensemble de solutions candidates [9].

1.4.2.2. Meta heuristique à une population :

a. Algorithmes génétiques :

Inspiré de la génétique, ils consistent à trouver une solution optimale à partir d'une population qui évolue de génération en génération [10].

b. Colonies de fourmis (ACO pour Ant Colony Optimisation) :

Est une méthode évolutive inspirée du comportement des fourmis à la recherche de nourriture. Cet algorithme a été proposé par Dorigo en 1992 [11].

c. Essaim de particules (Algorithme des Abeilles):

Les particules sont les individus et elles se déplacent dans l'hyperespace de Recherche. Le processus de recherche est basé sur deux règles [9]:

1. Chaque particule est dotée d'une mémoire qui lui permet de mémoriser le meilleur point par lequel elle est déjà passée et elle a tendance à retourner vers ce point.
2. Chaque particule est informée du meilleur point connu au sein de son voisinage et elle va tendre à aller vers ce point.

1.4 Formulation Mathématique du problème de lot sizing :

En 1958 **WAGNER** et **WITHIN** [6] ont mis au point une méthode qui a révolutionné le monde de la planification de la production avec leur algorithme de lot sizing (WWP)[12].

Ceci est le modèle de base, la structure étudiée se compose d'une unité de production, d'un dépôt central et d'un client. Les auteurs ont supposé que le problème de lot sizing est sans contrainte de capacité et avec un coût de setup (figure1.7). Ils ont proposé comme méthode de résolution la programmation dynamique.

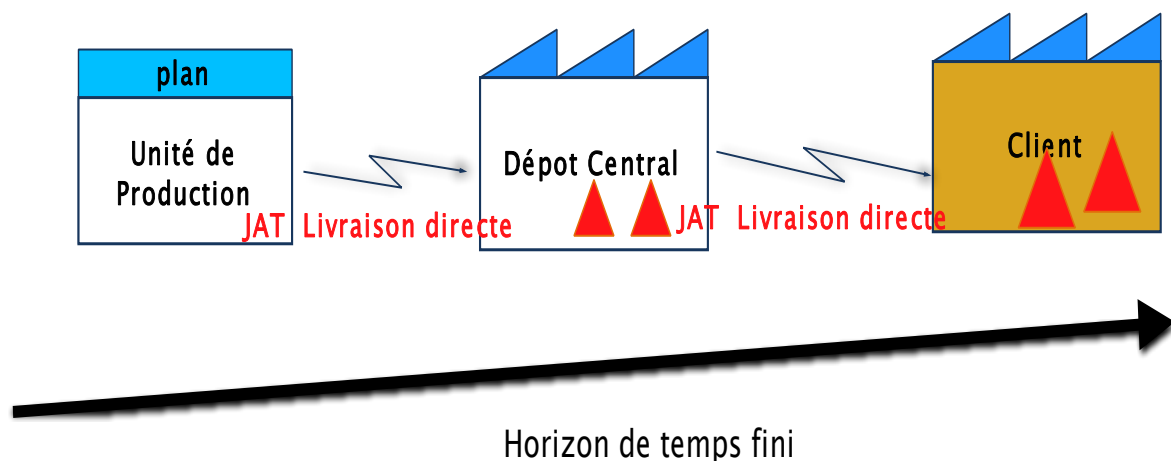


Figure 1.7.La structure du modèle de WAGNER et WITHIN

a. La fonction objective :

La fonction objectif est :

$$\text{Min } f(x, I) = \sum_{t=1}^T (f_t x_t + h_t I_t)$$



b. Les notations

Constantes

T Nombre de périodes

Ensemble des indices

$t = 1, \dots, T$ Intervalle des périodes

Paramètres

d_t Demande du produit dans la période.

h_t Coût de stockage dans la période t .

I_0 Niveau de stock.

f_t Coût de lancement.

Variables de décision

x_t Quantité à produire à la période t .

I_t Le niveau de stock à la fin de la période t

$y_t = 1$ *il ya fabrication du produit*

$y_t = 0$ *pas de fabrication*

c. Les contraintes :

$I_t = I_{t-1} + x_t - d_t$ Le stock à la période $t =$ stock à la période $t-1$ plus la quantité à produire moins la demande.

$I_0 = 0$ Stock initial vide.

$x_t \leq \left(\sum_{t'=1}^t d_{t'} \right) * y_t$ La quantité produite est inferieure ou égale au majorant de la quantité demandée, pour lancer la production.

$x_t, I_t \geq 0$ Les variables doivent être positives ou nulles

Conclusion

Ce chapitre est une présentation des notions de base de la chaine logistique ; la production et le lot sizing. Dans le chapitre suivant nous présentons la modélisation mathématique de notre problème et nous proposons une solution exacte en utilisant le solveur CPLEX.

Chapitre 2 :

Problème de lot sizing avec Transport

L'objectif de ce chapitre est de présenter le problème que nous allons étudier dans ce mémoire. Ce problème consiste à définir les quantités optimales de produit à transporter et à stocker dans une chaîne logistique. Nous donnons la description, la modélisation de la structure et nous présentons une solution en utilisant le solveur *CPLEX*. Un exemple de simulation nous montre la nécessité de développer une méthode heuristique pour résoudre le problème.

Introduction :

Nous considérons la structure d'une chaîne logistique, composée d'une usine qui fabrique un seul type de produit possédant une capacité de production limitée et qui expédie son produit dans un dépôt central (DC). Les produits sont livrés à différents clients disposant chacun d'un stock de capacité limitée.

Le but de ce chapitre est de proposer un plan de production sur un horizon fini, tout en prenant en considération les contraintes de production de transport et de stockage. Ce problème peut être vu comme un problème de lot sizing à un produit et plusieurs clients avec contrainte de capacité et transport.

Aussi nous présentons, dans ce chapitre, en premier, la description du problème considéré puis sa modélisation mathématique en programmation linéaire en nombre entier (PLNE) et par la suite sa résolution avec le logiciel GAMS.

2.1 Description du problème

Nous supposons une structure constitué d'une unité de production (UP) qui fabrique un seul type de produit avec une capacité de production fini et constante dans toutes les périodes t de l'horizon nt de planification. Cette unité de production se caractérise par un coût de setup de production et un coût unitaire de production qui sont fixe parce qu'elle produit un seule type. Le produit est expédié vers un dépôt central, lui-même a un coût de setup et un coût unitaire de stockage.

D'une autre part, on a ni clients qui disposent de stock de capacité limitée avec un coût de setup et un coût unitaire de stockage. Ces clients ont des demandes déterministes et variables sur l'horizon de temps. Le transport se fait sans contrainte de temps, on trouve une flotte de nv véhicules hétérogène disposant chacun d'une capacité finie, d'un coût de setup et un coût unitaire de transport. Suite à cette description la structure considérée est une mono usine mono dépôt (figure 2.1).

Nous avons aussi défini un ensemble d'hypothèses. Commençant par la quantité à produire x_t qui ne doit pas dépasser la capacité de production dans cette période. Dans chaque période nous avons ni demandes de ni clients qui peuvent se passer en même temps. Le nombre de véhicule est largement supérieur au nombre de client $ni \gg nv$. Le DC à une capacité illimitée.

Il ne faut livrer un que s'il y a suffisamment d'espace dans son stock. Enfin chaque véhicule ne doit visiter qu'un seul client dans une période. Initialement on considère que les stocks des clients et le DC sont vides.

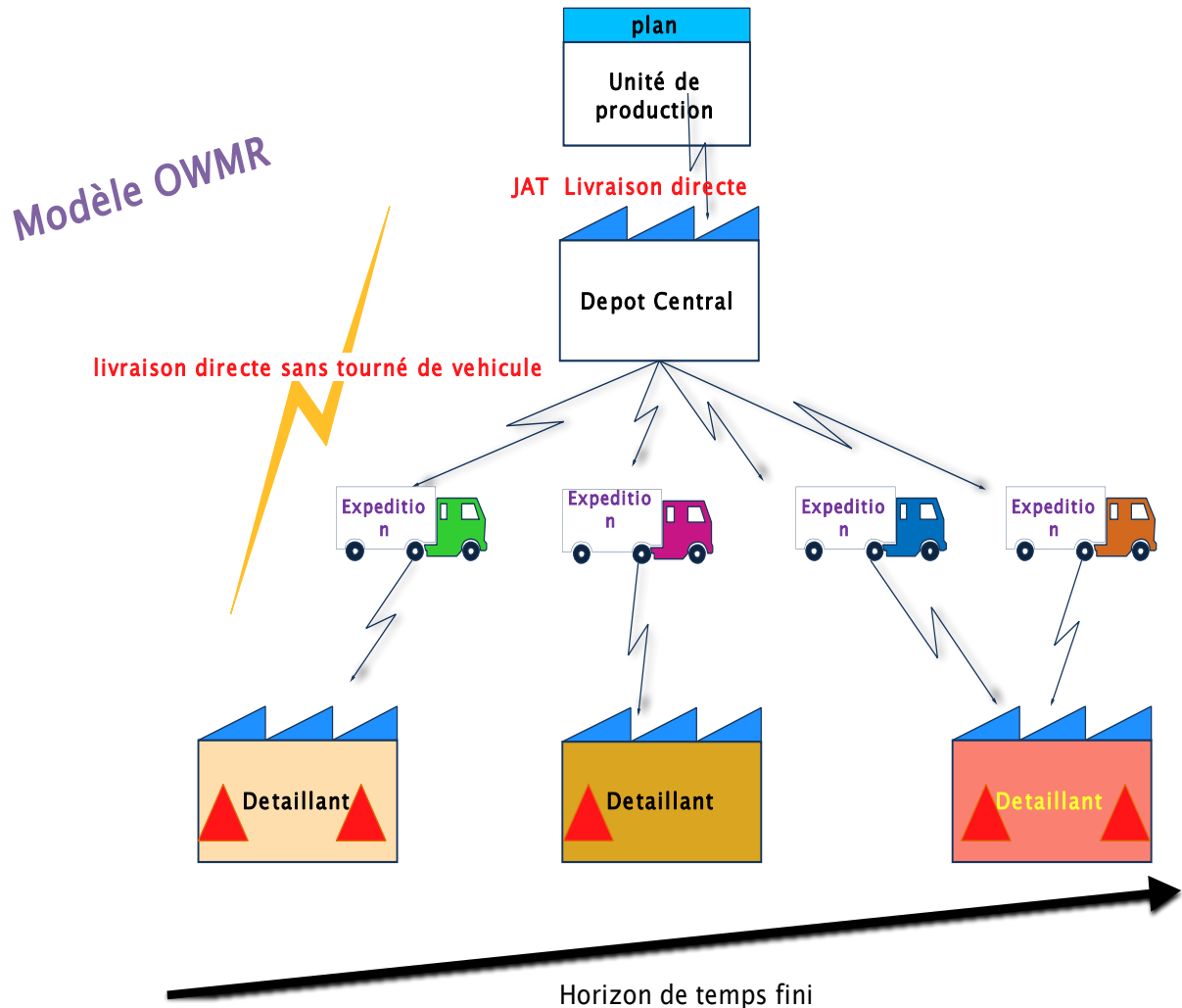


Figure.2.1. Schéma mono usine mono dépôt et multi client et stock.

2.2 Notations

Dans cette partie, nous présentons les notations utilisées dans la modélisation mathématique en programmation linéaire en nombre entier (PLNE).

2.2.1. Unité de production :

Les coûts de productions sont constants.

P : Capacité de production de l'unité de production.

Pf : Coût de setup de production.

Pu : Coût unitaire de production de l'unité de production.

2.2.2. Dépôt central :

De même pour les paramètres du DC, les coûts ne dépendent pas du temps. De plus la capacité du dépôt est supposée illimitée.

Df : Coût fixe de stockage dans le dépôt.

Du : Coût unitaire de stockage dans le DC.

2.2.3. Transport :

Les paramètres de transport sont Tf, Tu, W qui dépendent de chaque véhicule.

Tf_v : Coût fixe de transport payé pour le véhicule v expédié du DC vers un client.

Tu_v : Coût unitaire de transport pour une unité de produit transporté par le véhicule v

W_v : La capacité du véhicule v .

2.2.4. Stock client :

Le stock client dispose des paramètres de coût fixe de stockage, de coût unitaire et de capacité de stockage.

Sf_i : Coût fixe de stockage dans le stock du client i .

Su_i : Coût unitaire de stockage dans le stock du client i .

Cap_i : Capacité du stock du client i .

2.2.5 Les indices :

Dans le modèle mathématique nous manipulons les variables selon trois indices le premier " t " pour indiquer le temps, le second " i " indique les clients et le troisième " v " pour indiquer les véhicules.

2.2.6 Les ensembles :

Les indices du paragraphe précédent (t, i, v) prennent leurs valeurs des trois ensembles nt, ni, nv respectivement.

nt : Ensemble des périodes de l'horizon de planification.

ni : Ensemble des clients.

nv : Ensemble des véhicules.

2.2.7 Les variables de décision :

Notre modèle dispose de huit variables de décision, qui sont :

x_t : La quantité à produire à la période t .

- y_t : Variable binaire qui signifie s'il y a production à la période t ou non
- II_t : La quantité stockée au DC au début de la période t .
- Z_t : Variable binaire représentant l'état (vide ou non) du DC à la période t .
- III_{it} : La quantité stockée dans le stock du client i au début de la période t .
- ZZ_{it} : Variable binaire représentant l'état (vide ou non) du stock du client i à la période t .
- qt_{vit} : La quantité transportée vers le stock du client i par le véhicule v à la période t .
- vv_{vt} : Variable binaire représentant l'état du véhicule v (occupé ou non) à la période t
- ql_{it} : La quantité livrée au client i à la période t (égale à la somme des quantités transporté dans la même période)

2.3 Modèle mathématique en PLNE

Avant de donner la formulation globale, on a donné indépendamment les trois formules. Celle du coût de production, puis la formule du coût de stockage et enfin le coût de transport. Après on a assemblé le tout dans une formule globale.

$$\min(\text{coût Production} + \text{coût stockage depot} + \text{coût de transport})$$

2.3.1 Coût de production :

$$\text{Min} \sum_{t=1}^{nt} (Pf \cdot y_t + Pu_t \cdot x_t)$$

Le coût de production est égal à la somme du coût de setup (coût de lancement de production) plus le coût des quantités produites.

2.3.2 Coût de stockage du DC

$$\text{Min} \sum_{t=1}^T (Df_t \cdot Z_t + Du_t \cdot II_t)$$

Le coût de stockage du DC est égal au coût de lancement plus le coût de stockage des quantités stockées.

2.3.3 Coût de stockage du client

$$\text{Min} \sum_{t=1}^{nt} \sum_{i=1}^{ni} (Sf_t \cdot ZZ_{it} + Su_i \cdot III_{it})$$

De même pour le coût de stockage des clients, il est égal au coût de lancement plus le coût de stockage de quantités stockées.

2.3.4 Coût de transport

$$\text{Min} \left(\sum_{t=1}^{nt} \sum_{v=1}^{nv} (Tf_v \cdot VV_{iv}) + \sum_{t=1}^{nt} \sum_{i=1}^{ni} \sum_{v=1}^{nv} (Tu_t \cdot qt_{vit}) \right)$$

Le coût de transport est composé d'une part du coût de setup et du coût de transport des produits. Dans le coût de transport la quantité transportée est en fonction du véhicule, du client et de la période.

La formulation ci-dessous contient les contraintes. Elle est basée sur l'équilibrage des stocks et le respect des capacités.

$$\text{Min} \sum_{t=1}^{nt} ((Pf \cdot y_t + Pu \cdot x_t) + (Df \cdot Z_t + Du \cdot II_t)) + (\sum_{i=1}^{ni} Sf_i \cdot ZZ_{it} + Su_i \cdot III_{it})) + (\sum_{v=1}^{NV} (Tf_v \cdot VV_{vt} + \sum_{i=1}^{ni} Tu_t \cdot qt_{vit})) \dots\dots\dots(1)$$

$$y_t : \begin{cases} 1 & : \text{Si } x_t > 0 \\ 0 & : \text{si non} \end{cases} \dots\dots\dots(2)$$

$$z_t : \begin{cases} 1 & : \text{Si } II_t > 0 \\ 0 & : \text{si non} \end{cases} \dots\dots\dots(3)$$

$$zz_{it} : \begin{cases} 1 & : \text{Si } III_{it} > 0 \\ 0 & : \text{si non} \end{cases} \dots\dots\dots(4)$$

$$vv_{it} : \begin{cases} 1 & : \text{Si } qt_{vit} > 0 \\ 0 & : \text{si non} \end{cases} \dots\dots\dots(5)$$

$$II_t = II_{t-1} + x_t - \sum_{i=1}^{ni} ql_{it} \dots\dots\dots(6)$$

$$III_{it} = III_{i(t-1)} + ql_{it} - d_{it} \dots\dots\dots(7)$$

$$III_{it} \leq Cap_i \dots\dots\dots(8)$$

$$x_t \leq P_t \dots\dots\dots(9)$$

$$qt_{vit} \leq W_v \dots\dots\dots(10)$$

$$\sum_{v=1}^{nv} qt_{vti} = ql_{it} \dots\dots\dots(11)$$

$$\forall v, i, t \geq 0 \dots\dots\dots(12)$$

La fonction objective (1) minimise le coût total de production, de transport et de stockage. Les contraintes (2), (3), (4) et (5) définissent les domaines des variables de décision. Elles imposent que y_t , z_t , zz_{it} et vv_{it} soient des variables binaires. La contrainte (6) représente

l'équilibre du stock du dépôt central, le stock à une période donnée t est égale au stock jusqu'à $t - 1$ ajouté à la quantité produite à cette période moins le cumule des quantités à livrer à tous les clients dans cette même période. Pour la contrainte (7), l'équilibre des stocks des clients à la période t est égal au stock jusqu'à $t - 1$ ajouté à la quantité à livrer au client à la période t moins la demande de la même période. Les contraintes (8), (9) et (10) représentent les contraintes de respect de capacité, la contrainte (8) vérifie le niveau de stock du client i à la période t qui ne doit pas dépasser la capacité de stockage de ce stock client. Dans la contrainte (9), on trouve que la quantité à produire ne doit pas dépasser la capacité de production. Dans (10) la quantité à transporter ne dépasse pas la capacité du véhicule. (10) montre que la quantité livrée au client i est égale à la somme des quantités transportées par les véhicules v .

Une fois notre fonction objectif définie, nous avons opté, pour une méthode exacte en utilisant le solveur CPLEX du langage GAMS.

2.4. Résolution du problème par GAMS-CPLEX:

Nous avons utilisé GAMS (General Algebraic Modeling System) pour résoudre le modèle mathématique du problème avec le solveur CLPEX (référence au langage C et à l'algorithme du SIMPLEX),

2.4.1 Description des instances

On propose la génération des instances par la combinaison des trois paramètres suivants:

1. (v, i, t) : Nombre de véhicule v , de client i à la période t .
2. P : La capacité de production.
3. $C = (PU, SU(i), TU(v))$: Les coûts unitaires de production, de stockage client et de livraison

Pour chaque paramètre on prend 3 valeurs, ce qui nous donne $3^3 = 27$ possibilités (tableaux 2.1 et 2.2). Chaque instance sera générée 10 fois pour calculer le temps moyen d'exécution. Par contre le reste des paramètres du modèle doivent rester invariables.

1. $d(i, t) = U(5,25)$.
2. $PF = 10$
3. $DU = 0.5$
4. $DF = 1.5$
5. $W(v) = 0.1 * U(1,10) * dmax$ tel que $dmax$ est la demande max
6. $b = dmax * ni$.

- 7. $TF(v) = 0.1 * U(1,10)$
- 8. $CAP(i) = U(2,3) * dmaxcli(i)$. tel que $dmaxcli(i)$ demande max du client i.
- 9. $SF(i) = U(1,5)$;

$U(a, b)$: La loi uniforme discrète sur l'intervalle $[a, b]$.

	Valeur1	Valeur2	Valeur3
(V, I, T)	(10,20,10)	(20,40,20)	(40,60,30)
P	$1 * b$	$2 * b$	$3 * b$
Coût	Coût1	Coût2	Coût3

Tableau 2.1. Configuration des paramètres (V, I, T, P, Pu, Tu, Su) .

	Pu	$Tu(v)$	$Su(i)$
Coût1	$10 * U(1,5)$	$1 * U(2,5)$	$1 * U(1,5)$
Coût2	$1 * U(1,5)$	$10 * U(2,5)$	$2 * U(1,5)$
Coût3	$1 * U(1,5)$	$1 * U(2,5)$	$10 * U(1,5)$

Tableau 2.2. Différentes structures des coûts.

- Coût1 : le coût de production est élevé.
- Coût2 : le coût de livraison est élevé.
- Coût3 : le coût de stockage est élevé.

2.4.2 Tableau des instances et des résultats

Le tableau 2.3 présente une simulation du système étudié avec des paramètres générés aléatoirement. On forme toutes les possibilités du (tableau 2.2) ensuite on calcule le coût et le temps de calcul.

ion	Itérat	(nv, ni, nt)	P	Coûts			Temps moyen d'exécution (s)
				Pu	$Tu(v)$	$Su(i)$	
1		(10,20,10)	1xb	10xU(1,5)	1xU(2,5)	1xU(1,5)	0.169
2		(10,20,10)	1xb	1x U(1,5)	10xU(2,5)	2xU(1,5)	0.189
3		(10,20,10)	1xb	1x U(1,5)	1xU(2,5)	10xU(1,5)	0.176
4		(10,20,10)	2xb	10xU(1,5)	1xU(2,5)	1xU(1,5)	0.189

5	(10,20,10)	2xb	1xU(1,5)	10xU(2,5)	2xU(1,5)	0.183
6	(10,20,10)	2xb	1x U(1,5)	1xU(2,5)	10xU(1,5)	0.198
7	(10,20,10)	3xb	10xU(1,5)	1xU(2,5)	1xU(1,5)	0.186
8	(10,20,10)	3xb	1x U(1,5)	10xU(2,5)	2xU(1,5)	0.215
9	(10,20,10)	3xb	1* U(1,5)	1xU(2,5)	10xU(1,5)	0.192
10	(20,40,20)	1xb	10xU(1,5)	1xU(2,5)	1xU(1,5)	3.353
11	(20,40,20)	1xb	1xU(1,5)	10xU(2,5)	2xU(1,5)	3.640
12	(20,40,20)	1xb	1x U(1,5)	1xU(2,5)	10xU(1,5)	3.073
13	(20,40,20)	2xb	10xU(1,5)	1xU(2,5)	1xU(1,5)	4.184
14	(20,40,20)	2*b	1xU(1,5)	10xU(2,5)	2xU(1,5)	5.513
15	(20,40,20)	2xb	1x U(1,5)	1xU(2,5)	10xU(1,5)	4.527
16	(20,40,20)	3xb	10xU(1,5)	1xU(2,5)	1xU(1,5)	4.649
17	(20,40,20)	3xb	1xU(1,5)	10xU(2,5)	2xU(1,5)	4.370
18	(20,40,20)	3xb	1x U(1,5)	1xU(2,5)	10xU(1,5)	5.167
19	(40,60,30)	1xb	10xU(1,5)	1xU(2,5)	1xU(1,5)	18.782
20	(40,60,30)	1xb	1xU(1,5)	10xU(2,5)	2xU(1,5)	18.843
21	(40,60,30)	1xb	1xU(1,5)	1xU(2,5)	10xU(1,5)	18.647
22	(40,60,30)	2xb	10xU(1,5)	1xU(2,5)	1xU(1,5)	19.191
23	(40,60,30)	2xb	1xU(1,5)	10xU(2,5)	2xU(1,5)	19.681
24	(40,60,30)	2xb	1xU(1,5)	1xU(2,5)	10xU(1,5)	20.230
25	(40,60,30)	3xb	10xU(1,5)	1xU(2,5)	1xU(1,5)	22.939
26	(40,60,30)	3xb	1xU(1,5)	10xU(2,5)	2xU(1,5)	25.906
27	(40,60,30)	3xb	1xU(1,5)	1xU(2,5)	10xU(1,5)	30.802
28	(50,70,40)	1xb	10xU(1,5)	1xU(2,5)	1xU(1,5)	50.874
29	(60,80,50)	1xb	10xU(1,5)	1xU(2,5)	1xU(1,5)	104.450
30	(100,120,70)	1xb	10*U(1,5)	1xU(2,5)	1xU(1,5)	650.590

Tableau 2.3. Les instances et les résultats du solveur CPLEX.

2.4.3 Interprétation des résultats

La figure 2.3 présente le temps moyen d'exécution en fonction des (nv, ni, nt) , on remarque que le temps augmente avec l'augmentation des (nv, ni, nt) , cela signifie que le solveur CPLEX consomme plus de temps pour trouver la solution du problème. La courbe est bien uniforme et croissante, ce qui indique que ces trois valeurs ont une grande influence sur le temps

de calcul. Avec les grandes itérations le calcul devient très lent voir impossible d'exécuter le programme sur une machine ordinaire.

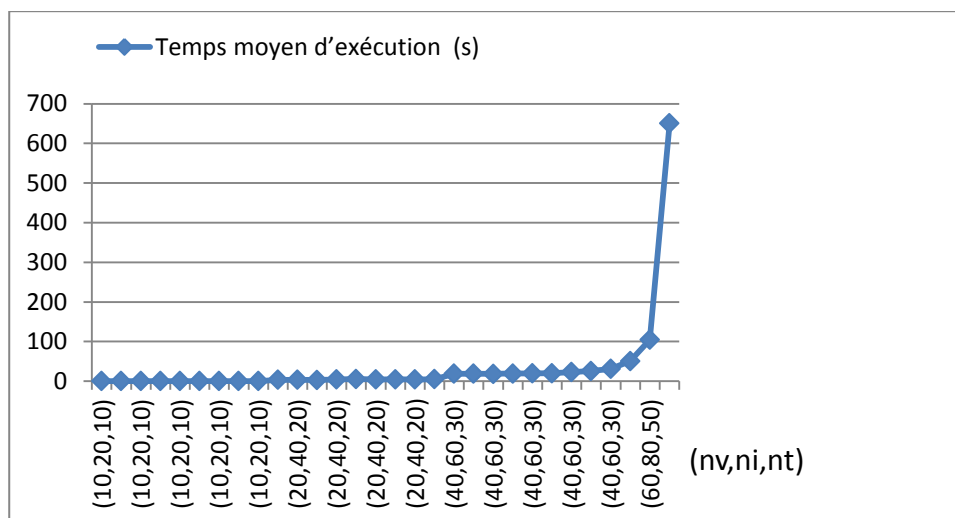


Figure 2.2 Graphe du temps de calcul en fonction du nombre de véhicule, de client et de période.

Pour voir l'influence de la capacité de production, on fixe (nv, ni, nt) à $(10, 20, 10)$ (les 9 premières itérations tableau 4) et on trace la courbe de temps de calcul (figure 2.3).

Itération	(nv, ni, nt)	P	Coûts			Temps moyen d'exécution (s)
			Pu	Tu(v)	Su(i)	
1	(10, 20, 10)	1*b	10*U(1,5)	1*U(2,5)	1*U(1,5)	0.169
2	(10, 20, 10)	1*b	1* U(1,5)	10*U(2,5)	2*U(1,5)	0.189
3	(10, 20, 10)	1*b	1* U(1,5)	1*U(2,5)	10*U(1,5)	0.176
4	(10, 20, 10)	2*b	10*U(1,5)	1*U(2,5)	1*U(1,5)	0.189
5	(10, 20, 10)	2*b	1* U(1,5)	10*U(2,5)	2*U(1,5)	0.183
6	(10, 20, 10)	2*b	1* U(1,5)	1*U(2,5)	10*U(1,5)	0.198
7	(10, 20, 10)	3*b	10*U(1,5)	1*U(2,5)	1*U(1,5)	0.186
8	(10, 20, 10)	3*b	1* U(1,5)	10*U(2,5)	2*U(1,5)	0.215
9	(10, 20, 10)	3*b	1* U(1,5)	1*U(2,5)	10*U(1,5)	0.192

Tableau 2.4. Les instances pour $(nv, ni, nt) = (10, 20, 10)$.

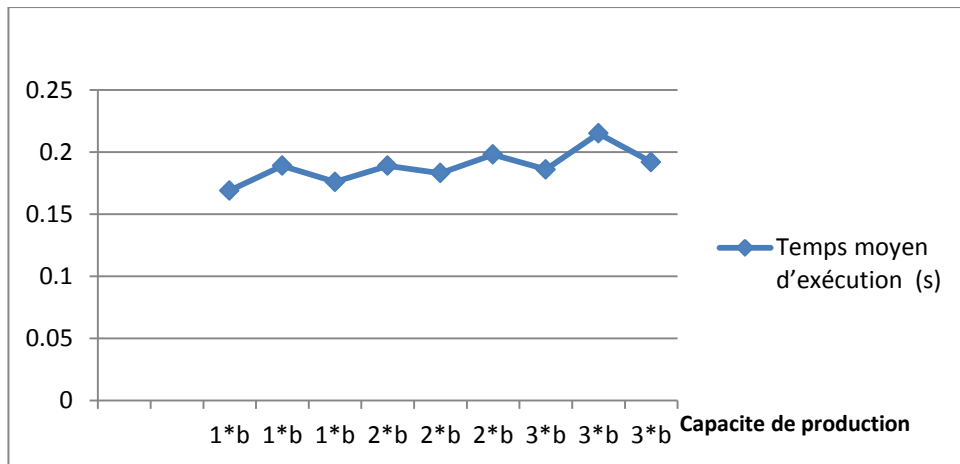


Figure 2.3. Graphe du temps de calcul en fonction de la capacité de production pour $(nv, ni, nt) = (10, 20, 10)$.

La figure 2.3 montre que le temps de calcul augmente avec l'augmentation de la capacité de production. Cette progression n'est pas uniforme ce qui indique que la capacité de production n'a pas une grande influence sur le temps de calcul. On remarque que dès que le nombre de variable augmente, le système devient de plus en plus lent il peut même arriver à des heures de calcul, ce qui n'est pas pratique pour les grands systèmes. Aussi afin d'aboutir à une solution dans ce cas, nous allons utiliser une méthode approchée. Nous proposons l'utilisation des algorithmes génétiques qui sont les plus adaptés pour ce type de problèmes. Dans le chapitre suivant nous expliquons en détail le principe des algorithmes génétiques, et leur application au problème de lot sizing avec transport.

Conclusion

Dans ce chapitre nous avons pu faire la modélisation mathématique du problème de lot sizing avec transport. Nous avons utilisé le solveur *CPLEX* pour résoudre ce problème, nous avons abouti à des résultats satisfaisants mais à une certaine limite de nombre des variables (tableau 2.3.). Mais dans le cas où le nombre de variables augmente il faudra chercher une heuristique qui économise plus de temps.

Dans le chapitre suivant nous allons voir comment résoudre ce problème en utilisant les algorithmes génétiques.

Chapitre 3 :

Résolution du problème par les algorithmes génétiques

Ce chapitre présente une approche de résolution du problème par les algorithmes génétiques. Nous commençons d'abord par les terminologies et le principe des algorithmes génétiques, nous décrivons ensuite le découpage de la structure en sous structure pour simplifier l'utilisation de l'algorithme génétique avec le codage binaire du multi chromosome. Aussi nous introduisons l'algorithme du zéro Switch pour calculer les quantités à livrer et les quantités à produire et l'algorithme glouton pour calculer la quantité à transporter par les véhicules. Le tout est assemblé dans l'algorithme génétique. Nous implémentons et nous simulons des exécutions avec des valeurs arbitraires des paramètres, ce qui rend notre approche valable pour toutes les chaînes logistiques de ce genre. Pour vérifier le degré de validité de notre approche, nous comparons notre approche avec les résultats de la méthode exacte élaborée au chapitre 2. Nous terminons par une interprétation des résultats.

Introduction

La principale motivation d'une approche intégrée est de minimiser le coût total généré tout au long de la chaîne logistique. Dans ce chapitre qui présente le noyau de notre étude, nous allons expliciter la démarche suivie pour résoudre le problème de lot Sizing avec transport.

Afin d'optimiser la structure intégrale on a eu recours à une méthode approchée. Nous proposons comme méthode de résolution d'utiliser les algorithmes génétiques à codage binaire, qui sont des méta-heuristiques. Les algorithmes génétiques sont les mieux adaptés à ce type de problèmes. L'application de l'algorithme génétique dans l'état brut du problème devient alors compliquée à cause des contraintes. On a opté pour une approche modulaire pour notre structure afin de faciliter l'implémentation. Il a fallu utiliser d'autres heuristiques, comme le zéro Switch pour calculer les quantités à produire et à livrer d'un côté et l'algorithme glouton pour calculer la quantité à transporter. Ces algorithmes nous ont permis de ramener les variables de décisions à un type binaire seulement. Ceci nous a facilité l'implémentation, accéléré l'exécution et a permis d'aboutir à des résultats. Aussi pour donner une chronologie à cette approche, nous présentons dans ce chapitre la terminologie et le principe des algorithmes génétiques. Par la suite, on utilise le PLNE (Modélisation mathématique) du chapitre 2 pour implémenter l'algorithme génétique. Nous terminons par une simulation et une comparaison des résultats de notre approche avec ceux du CPLEX.

3.1 Principes des Algorithmes Génétiques :

En 1859 les bases de l'évolution étaient posées par C. Darwin avec son idée de la sélection naturelle (« *dans tout espèce les meilleurs sont sélectionnés* »). En 1901 ceux sont les bases de la génétique qui étaient posés par De-Vries suite à sa théorie du mutationnisme. En 1975 Jhon Holland proposa l'Algorithme Génétique. En 1989 Goldberg exposa les fondements mathématiques des algorithmes génétiques.[13]

Les algorithmes génétiques consistent à faire évoluer une population initiale tiré au hasard vers l'optimum par des opérations simples qui sont : la sélection le croisement et la mutation sur un certain nombre de générations.

3.1.1 Concepts des algorithmes génétiques :

Les algorithmes génétiques sont basés sur cinq niveaux d'organisation comme présenté sur la figure 3.1.

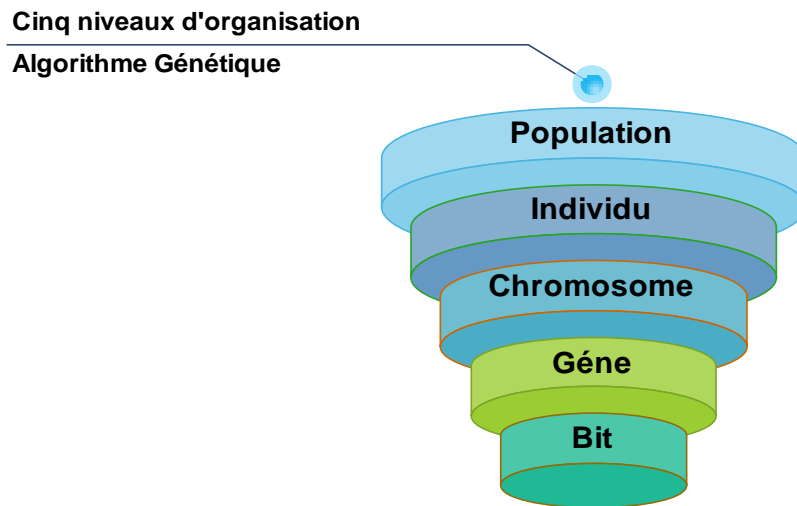


Figure.3.1. Les cinq niveaux d'organisation dans les algorithmes génétiques [10].

Bit : Est la plus petite unité pour présenter une donnée.

Gène : A chaque variable d'optimisation x_i , nous faisons correspondre un gène.

Chromosome : Un chromosome est un ensemble de gènes, (figure 3.2). Chaque dispositif est représenté par un chromosome.

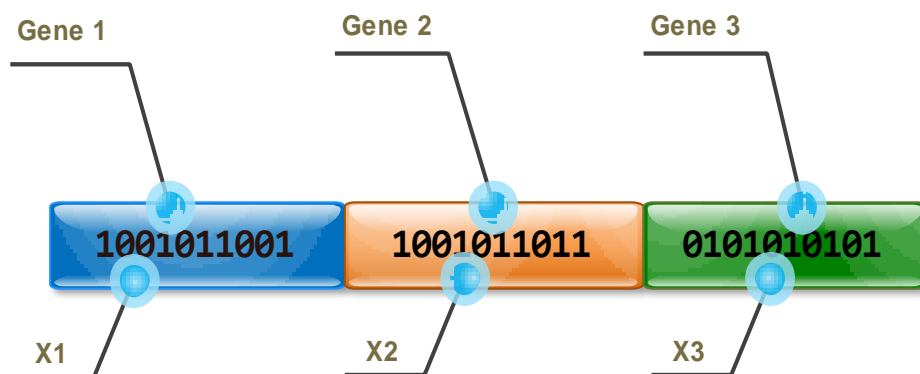


Figure.3.2. Codage des variables [10]

Individu : Un individu est constitué d'un ou plusieurs chromosomes

Population : Une population est un ensemble de N individus qui vont évoluer.

Génération : C'est la population à l'étape i .

Une génération peut être représentée par la figure 3.3.

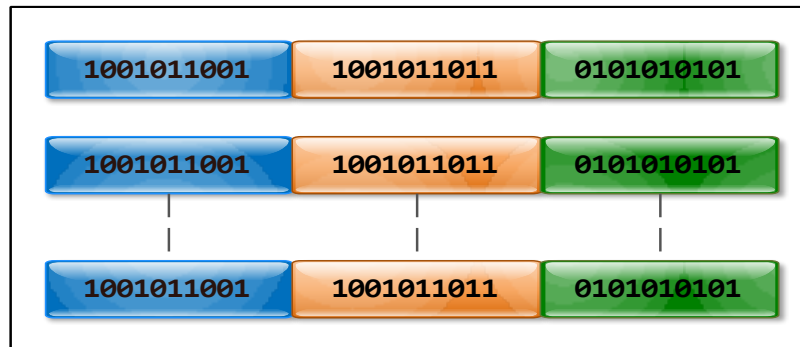


Figure 3.3. Présentation de la population à une génération i [10].

3.1.2 Opérations des algorithmes génétiques :

Les algorithmes génétiques sont une méthode basée sur un ensemble d'opérations. La succession de ces opérations est donnée par la figure 3.4.

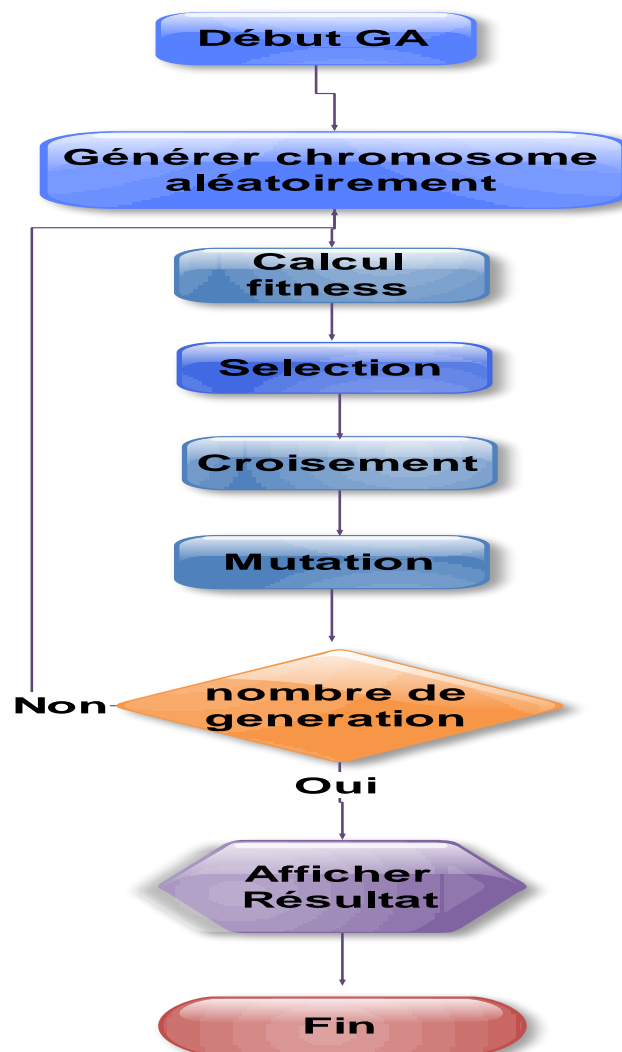


Figure 3.4. Les opérations des algorithmes génétiques [14]

3.1.2.1 . Sélection : c'est une étape très importante, c'est elle qui détermine les individus qui survivent de ceux qui vont être éliminés. La sélection se fait à partir de la fonction objective. Seuls les individus passant l'épreuve de sélection peuvent accéder à la génération suivante et se reproduire. Mahdi [13], a cité présenté plusieurs méthodes de sélection.

- **Elitisme** : les individus sont triés selon leur fonction d'adaptation (fitness) par ordre décroissant. Seule les n premiers seront sélectionnés. Autrement dit ceux correspondant aux meilleurs composants. Cependant cette méthode induit à une convergence prématurée de l'algorithme, parce que même les individus ayant une valeur moins importante peuvent faire naître des individus enfant de bonne qualité.
- **Roulette de Wheel (Roue de la Fortune)** : Elle est basée sur des probabilités. Pour cette méthode il faut appliquer un ensemble d'étapes :
 - Calculer le fitness de chaque individu.
 - Calculer la probabilité proportionnelle $Pr(i)$ pour chaque individu.
 - Calculer la probabilité proportionnelle cumulée pour chaque individu $cum(i)$ (figure 3.5).
 - A chaque individu générer aléatoirement une probabilité $r(i)$.
 - Garder les individus qui ont une probabilité $r(i) \leq cum(i)$.

Les différentes étapes de son application sont

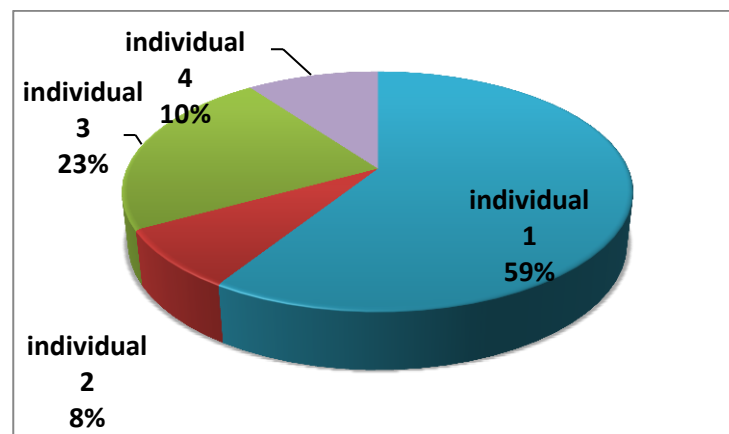


Figure.3.5. Roulette de Wheel. [10]

Si $r(i) \leq cum(i)$ alors l'élément i est sélectionné. C'est-à-dire chaque fois que le cumule est plus grand l'individu a plus de chance d'être sélectionné pour la prochaine population.

D'après la (figure 3.5), la chance de sélectionner l'individu 1 est 59% par contre l'individu 4 a 10% de chance d'être sélectionné. De cette façon les mieux adaptés survivent et forment la nouvelle population et obéissent aux opérations de croisement et de mutation.

- **Sélection par tournoi** : Cette technique consiste à prendre avec remise deux individus, puis choisir parmi ces paires l'individu qui a le meilleur score d'adaptation.
- **Sélection uniforme** : La sélection se fait aléatoirement, uniformément et sans intervention de la valeur d'adaptation. Chaque individu a donc une probabilité $1/P$ d'être sélectionné, où P est le nombre total d'individus dans la population.

3.1.2.2. Croisement : Une fois la génération intermédiaire sélectionnée, les individus sont aléatoirement répartis en couples, les individus parents sont alors copiés et recombinaison de façon à former deux descendants possédant des caractéristiques héritées des deux parents. Il faut alors choisir un ou plusieurs points de croisement aléatoirement et permuter les données entre les deux parents à partir des points de croisement. Les figures (3.6) et (3.7), présentent respectivement un exemple de croisement à un point et un croisement multipoint [14]. Les parents sont pris au hasard de la nouvelle population, les individus enfants remplacent les parents. L'intervalle de la probabilité est que nous avons utilisé est [0.7, 0.9].

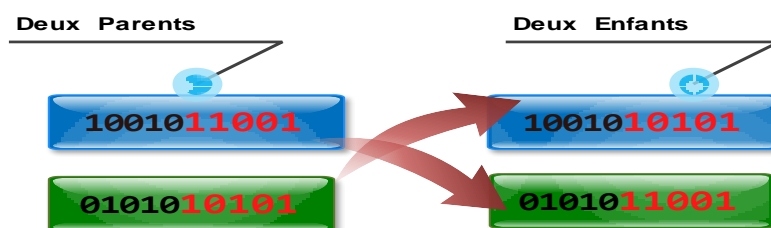


Figure.3.6. Opération de croisement à un point [10].

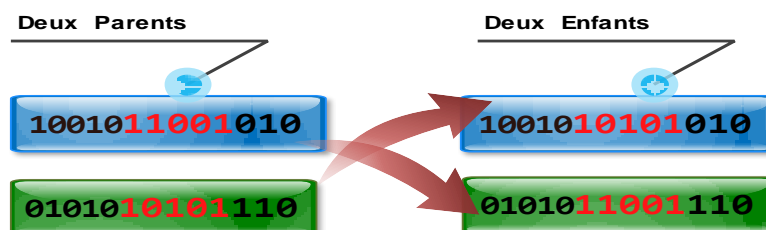


Figure.3.7. Opération de croisement multipoint [10].

3.1.2.3 . Mutation : est la dernière opération de la procédure, elle a pour objectif de modifier les gènes des parents. Cette opération consiste à inverser la valeur d'un bit. Si le bit est à 1 le rendre 0 et inversement (figure 3.8). Cette opération est aussi basée sur la probabilité pm pour choisir les individus à muter. (Il existe la méthode de probabilité de mutation pm fixée entre 0,001 et 0,01. Nous pouvons prendre également $pm = 1/l$ où l est le nombre de bits composant un chromosome). Sans cette opération la population ne peut jamais avoir de nouveaux individus pour évoluer [10].



Figure.3.8. Opération de mutation [10].

Après ce qui précède, nous donnons dans ce qui suit la structure générale des algorithmes génétiques, avec comme critères d'arrêt le nombre de génération.

3.1.3 Structure générale des algorithmes génétiques

Dans l'algorithme suivant nous assemblons l'ensemble des opérations dans l'ordre logique des instructions.

Algorithme 1 Algorithme Génétique

Début

Coder le chromosome

Générer la population initiale

Répéter *% nombre de génération*

Calculer le fitness

Sélection

Croisement

Mutation

Fin Jusqu'au nombre de génération

Fin

3.2 Application des Algorithmes génétiques pour la résolution du problème de lot Sizing :

L'application des AG, pour la résolution du problème de lot sizing considéré, sur le modèle dans sa globalité est très difficile, aussi nous avons opté pour décomposer le modèle en trois sous structures (figure 3.9).

- La productions : qui présente l'unité de production et le stock du dépôt central avec les variables de décisions $x(t)$, $y(t)$, $\Pi(t)$, $z(t)$.
- Le transport : qui présente les véhicules avec la variables de décisions $qt(v,i,t)$ $vv(v,t)$.
- Le client : qui est présenté par le stock client avec les variables de décisions $ql(i,t)$ $\text{III}(i,t)$ $zz(i,t)$.

Ainsi notre fonction objectif est constituée de 8 variables de décisions qui sont $x(t)$, $y(t)$, $z(t)$, $\Pi(t)$, $qt(v,i,t)$, $vv(v,t)$, $\text{III}(i,t)$ et $zz(i,t)$. Nous remarquons que la variable $ql(i,t)$ est une variable qui ne fait pas partie de la fonction objectif mais qu'on a utilisé comme variable intermédiaire pour calculer d'autres variables (voir algorithme 5). L'application de l'algorithme génétique sur le modèle dans son état brut est difficile si ce n'est impossible. Nous utiliserons le codage binaire et nous déduirons le reste des variables de décision avec les algorithmes de réparations. La démarche à suivre est dans l'organigramme de la figure (3.10).

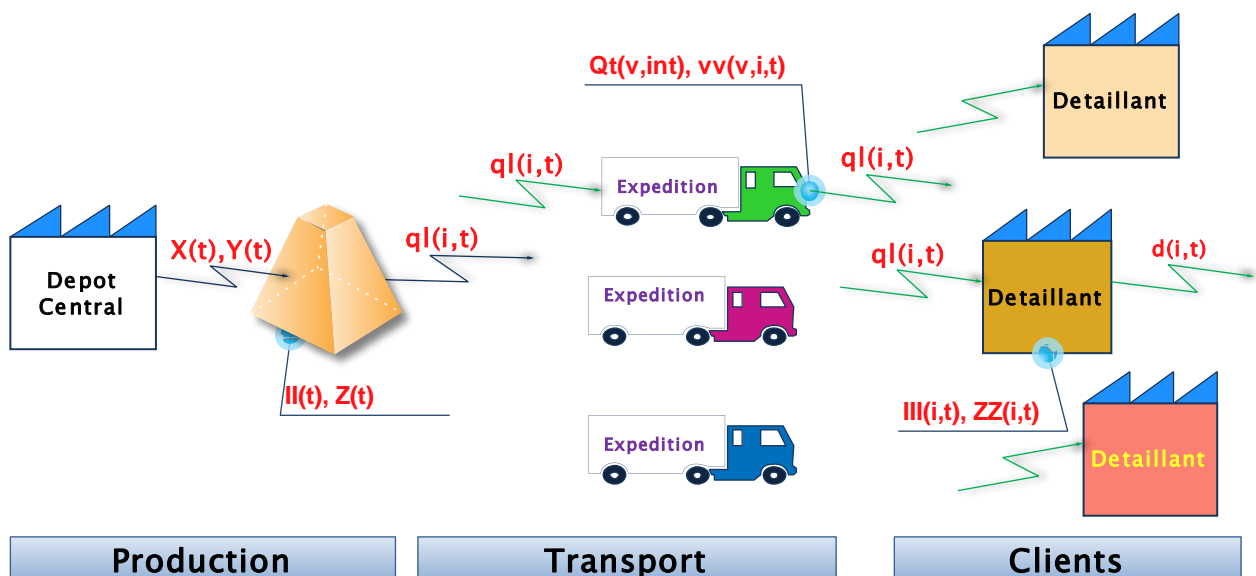


Figure .3.9. Décomposition du système étudié en 3 sous structures.

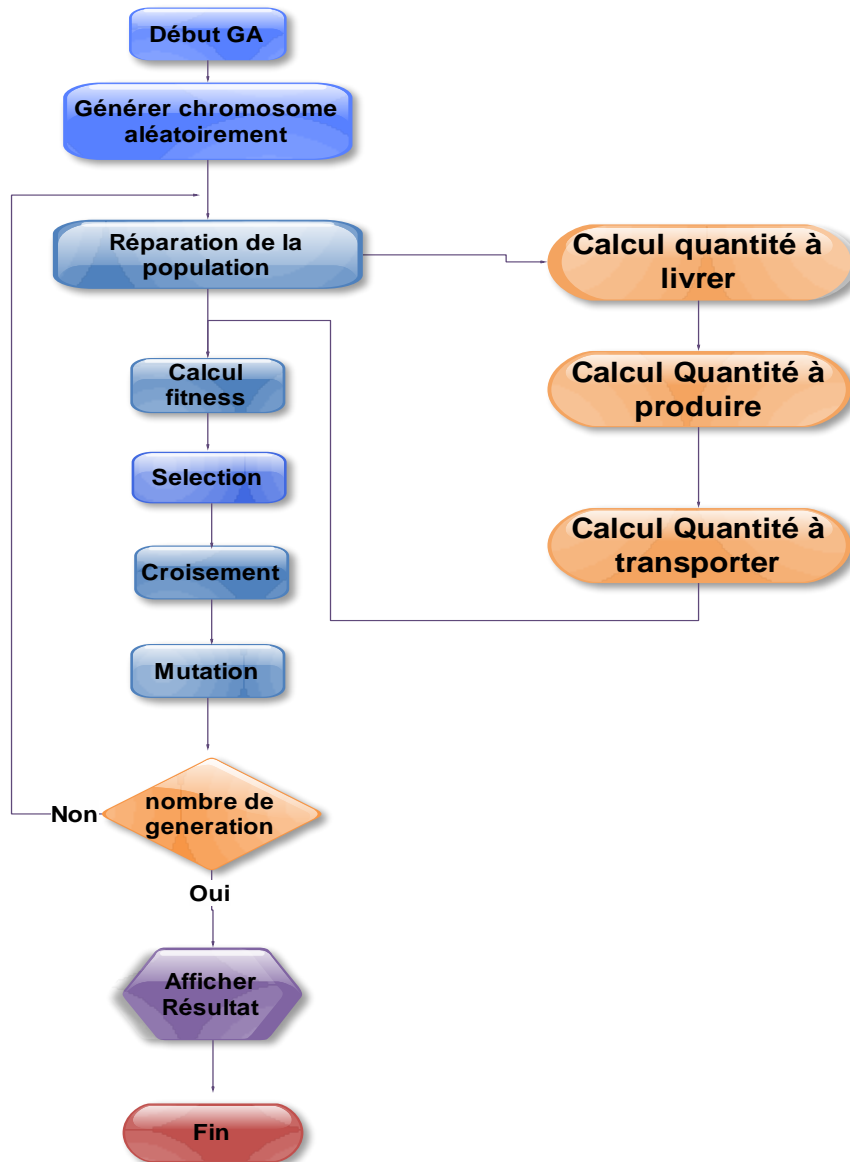


Figure 3.10. Organigramme de l’algorithme génétique.

3.2.1 Codage du chromosome :

Dans le cadre de notre application, nous avons opté pour un codage binaire qui limite les variables de décision à deux valeurs. Ainsi le multi chromosome à la structure suivante :

$$\begin{bmatrix}
 [y(1) & y(2) & \dots & \dots & \dots & \dots & y(nt)] \\
 [zzz(1,1) & zzz(1,2) & \dots & \dots & \dots & \dots & zzz(1, nt)] \\
 [zzz(2,1) & zzz(2,2) & \dots & \dots & \dots & \dots & zzz(2, nt)] \\
 [zzz(ni, 1) & zzz(ni, 2) & & & & & zzz(ni, nt)]
 \end{bmatrix}$$

Figure .3.11. Structure du multi chromosome.

Le multi chromosome, est une matrice binaire de $ni+1$ lignes et nt colonnes formant deux parties, la première partie occupe la première ligne présente par la variable $y(t)$ signifiant s'il y a production ou non. La seconde partie est une matrice de ni lignes et nt colonnes représentée par la matrice $zzz(i,t)$ (figure 3.11), signifiant s'il y a livraison au client i à la période t ou non. Puisque le multi chromosome est binaire et nous avons les variables de décision de type entier et réel, nous avons appliqué le principe zéro Switch pour déduire le reste des variables de décision à partir du multi chromosomes.

3.2.2 Principe de zéro Switch :

La méthode du « zero switch » [15] et [6] donne une solution optimale au problème de lot sizing sans capacité.

Le principe est le suivant :

- $x(t)$: quantité produite à l'instant t .
- $y(t)$: variable binaire 0/1 qui indique s'il y a production ou non.
- $I(t)$: le niveau de stock à l'instant t .

$$x(t).I(t) = 0 \rightarrow \begin{cases} x(t) = 0 \text{ et } I(t) = 0 \\ \text{ou} \\ x(t) = 0 \text{ et } I(t) \neq 0 \\ \text{ou} \\ x(t) \neq 0 \text{ et } I(t) = 0 \end{cases}$$

Ce qui signifie assurer le zéro stock ou zéro quantité produite. L'exemple suivant dans la (figure 3.12) illustre le principe. Si y est nul alors la quantité à produire est aussi nulle. Mais dans le cas contraire, il faut produire la quantité demandée dans cette période et les périodes suivantes (la somme des demandes), jusqu'à ce que y redevienne nul. L'algorithme suivant illustre ce principe :

Algorithme 2 zéro Switch

Début

Si $y(t)=0$ *alors* $x(t)=0$

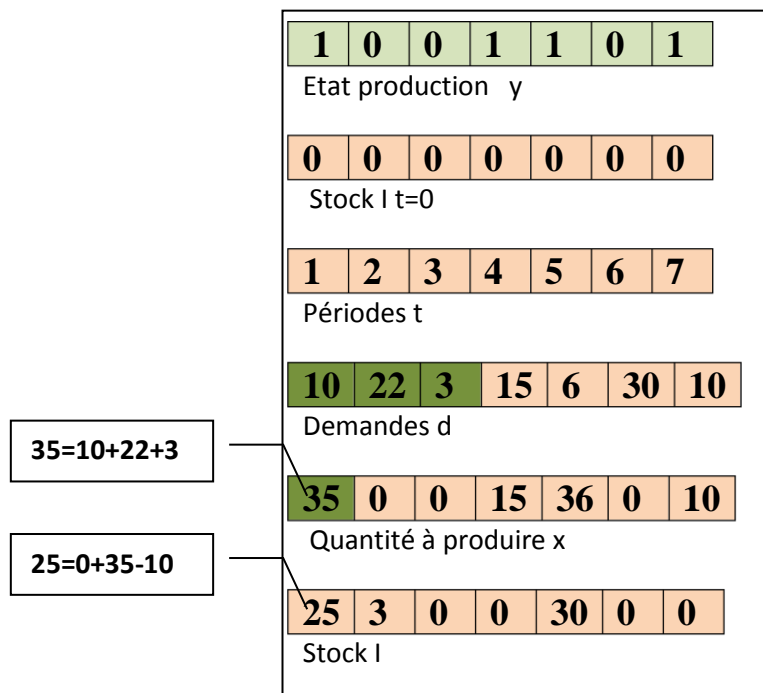
Sinon

tant que $y(t)=0$ *faire*

$$x(t)=x(t)+x(t-1)$$

$$I(t)=I(t-1)+x(t)$$

$$t=t+1$$

*Fin TQ**Finsi**Fin***Exemple d'application :****Figure.3.12.** Principe du zéro Switch.

Dans les périodes 2 et 3 $y=0$, donc on doit produire pour les trois périodes 1,2,3 $x(1)=10+22+3=35$. Aussi cette même quantité part au stock mais on doit réduire la quantité demandée dans la période 1 $I(1)=I(0)+x(1)-d(1)=0+35-10=25$.

Nous avons constaté qu'au niveau de la production et du client, nous pouvons appliquer le principe de zéro Switch, mais nous sommes confrontés à deux problèmes.

1. Les contraintes qui sont l'équilibre du stock du DC, l'équilibre du stock client, la capacité de stock du client i à la période t , la capacité de production et la capacité des véhicules v .
 2. Pour calculer les valeurs des variables de décision qui restent, on utilise les contraintes elles-mêmes. A partir des contraintes des égalités nous déduisons le reste des variables de décision, et à partir des contraintes des inégalités nous vérifions les faisabilités ou les capacités.
- Dans la contrainte $III_{it} = III_{i(t-1)} + ql_{it} - d_{it}$, nous calculons III si nous avons ql .

- Dans la contrainte $III_{it} \leq Cap_i$, nous pouvons contrôler la capacité de stock.

Nous allons maintenant détailler ce procédé.

3.2.3 Les algorithmes de réparation

L'opération de réparation consiste à faire des modifications sur le multi chromosome à l'aide de la vérification des contraintes de capacité pour le DC, le stock client et les capacités des véhicules.

Nous avons pu développer trois algorithmes de réparation. Le premier calcule les quantités livrées aux clients, le second calcule les quantités produites et le troisième calcule les quantités transportées par chaque véhicule.

3.2.3.1 Calcul de la quantité à livrer à chaque client

Cet algorithme calcule les quantités livrées pour chaque client à chaque période. Nous avons comme données d'entrée le chromosome $zzz(i,t)$ qui signifie s'il y a une livraison pour le client i à la période t ou non. L'exemple suivant illustre ce principe (figure 3.13) :

Supposons que pour le client $i=1$, nous avons les demandes $d(1,t)$ et la capacité de stock client $cap(i)$, si nous appliquons l'algorithme du zéro Switch sans correction nous trouvons dans la période $t=1$ la quantité à transporter égale à 70 unités alors que la capacité est seulement de 50 (figure 12). Il faut donc réparer cette contrainte en faisant la comparaison avec la capacité à chaque période. Dès que la condition n'est pas vérifiée, on répare le chromosome (modifier le 0 à 1) c'est-à-dire refaire une autre livraison à cette période ainsi de suite tout au long de l'horizon de planification (Figure 3.14).

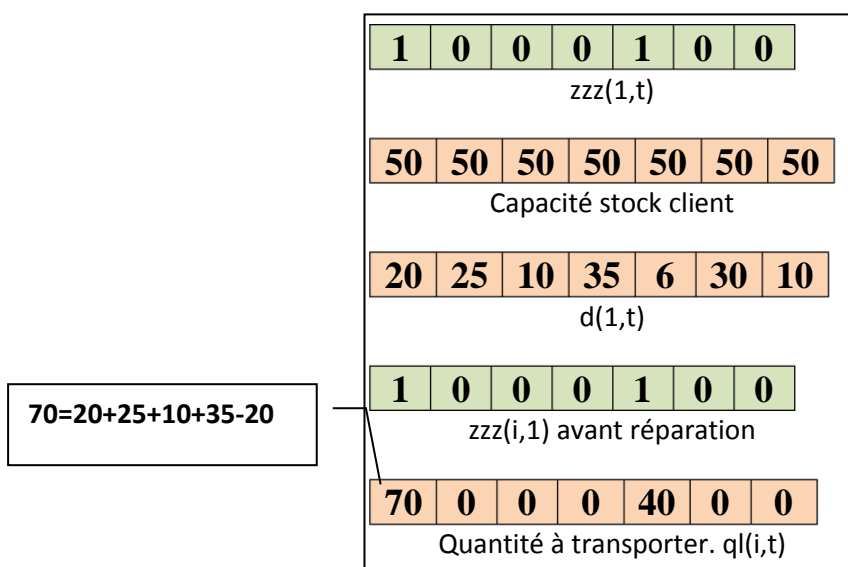


Figure .3.13. Calcul de la quantité à livrer sans réparation.

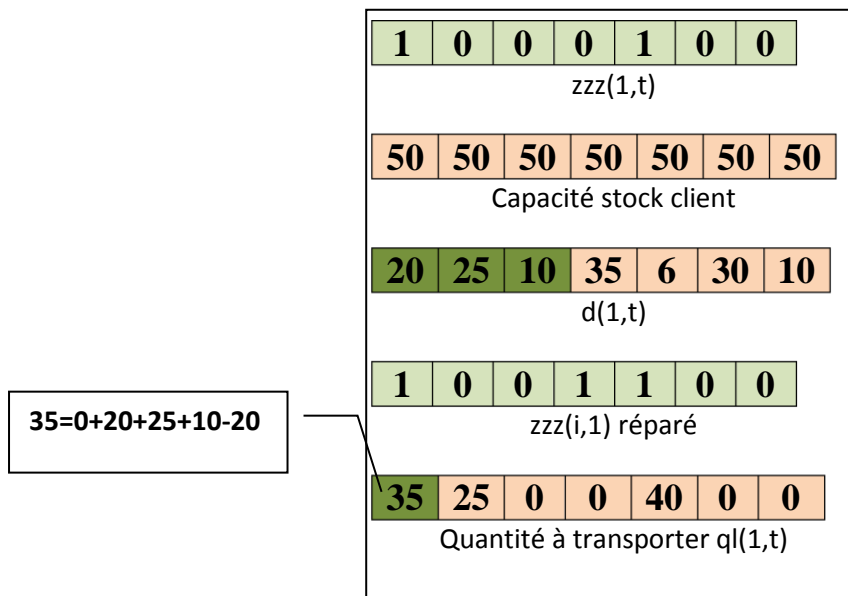


Figure .3.14. Calcul de la quantité à livrer avec réparation.

Ce principe est traduit par l'algorithme 3.

Algorithme 3 calcul de la quantité à livrer aux clients

Début

$t=1$;

$zzz(i,1)=1$; /* livré ou non =0 ou non

$ql(1,0)=0$

$III(i,0)=0$

$zzz=0$

$ql=0$

$III=0$

Pour chaque client i faire

Pour chaque instant t faire

Si $zzz(i,t)=0$ **alors**

$ql(i,t)=0$

Sinon

$t1=t$

$t=t+1$

Tant que $zzz(i,t)=0$ **faire**

$ql(i,t1)=ql(i,t1)+d(i,t)-d(t1)$

Si $ql(i,t1) <$ la capacité du stock client

$ql(i,t1)=ql(i,t1)-d(i,t)$

$zzz(i,t)=1$

sortir boucle tant que

fin si

$t=t+1$

fin tant que

```

Fin si
   $III(i,t)=III(i,t-1)+ql(i,t)-d(i,t)$ 
Fin pour
Fin pour
Fin

```

3.2.3.2 Calcul de la quantité à produire :

Le principe de calcul de la quantité à produire pour chaque période est le même que pour la quantité à livrer à chaque client. La quantité à produire n'est pas en fonction de la demande du client, mais en fonction de la somme des quantités à livrer aux clients. L'algorithme 4 présente le principe.

Algorithme 4 calcul de la quantité à produire

```

Début
   $T=1$  ;
   $y(1)=1$  ; /* produire ou non
   $X(1)=0$ 
   $II(0)=0$ 
  Pour chaque instant t faire
    Si  $y(t)=0$  alors
       $X(t)=0$ 
    Sinon
       $t1=t$ 
       $t=t+1$ 
      Tant que  $y(t)=0$  alors
         $X(t1)=X(t1)+\sum_{i=1}^{ni} ql(i, t)-\sum_{i=1}^{ni} ql(i, t1)$ 
        Si  $X(t1) <$  la capacité de production
           $X(t1)=X(t1)-\sum_{i=1}^{ni} ql(i, t)$ 
           $y(t)=1$ 
           $t=t-1$ 
        Sortir boucle tant que
      Fin si
       $t=t+1$ 
    fin tant que
  Fin si
   $II(t)=II(t-1)+X(t)-\sum_{i=1}^{ni} ql(i, t)$ 
Fin pour t
Fin

```

3.2.3.3 Calcul de la quantité à transporter pour chaque véhicule

Une fois que les quantités à livrer et les quantités à produire sont calculées, il reste maintenant à calculer la quantité à transporter par chaque véhicule à une période donnée, et à quel client vat-on être livrer tout en minimisant les coûts.

Suivant les hypothèses de notre cas, on ne considère pas de tournées. Par conséquent, en respectant les capacités des véhicules, un client peut être servi (ou visité) par plusieurs véhicules pour avoir sa commande satisfaite (figure.3.15).

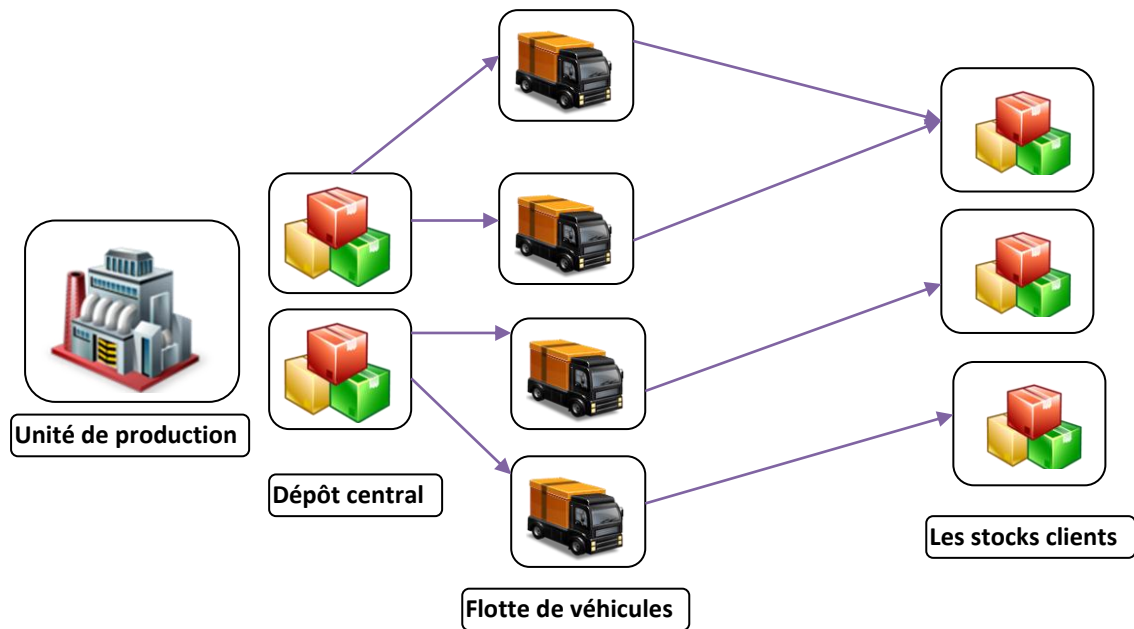


Figure .3.15. Affectation des véhicules aux différents clients.

Pour mieux illustrer ce problème de quantité à livrer (ou d'affectation de véhicule), nous présentons l'exemple suivant (figure 3.16). Pour notre cas on suppose que toutes les demandes sont satisfaites à la même période et le nombre de véhicules est suffisant pour satisfaire toutes les demandes.

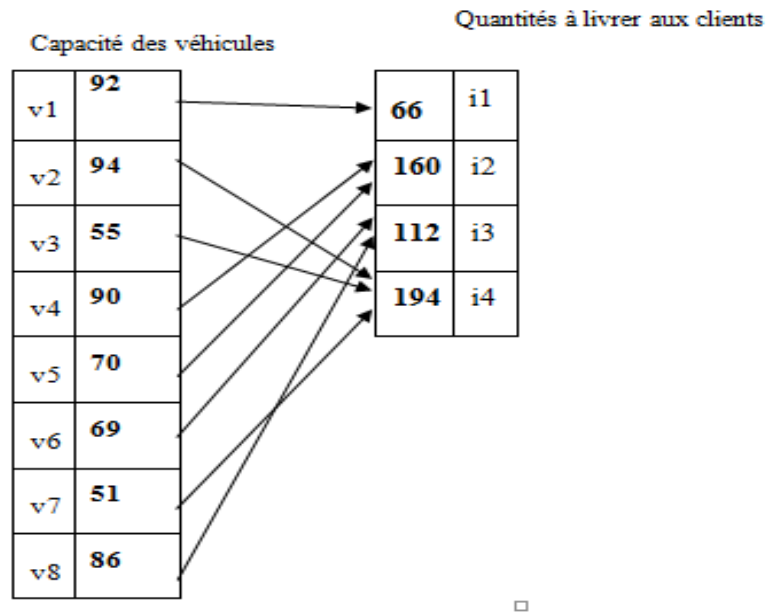


Figure.3.16. Affectation des véhicules aux clients à un instant t

A partir de cet exemple on constate deux cas de figure.

- Le client i_1 voit sa demande satisfaite par un seul véhicule v_1 .
- Le client i_4 par contre sa demande est honorée suite à la visite des véhicules v_3 , v_2 et v_7

Ce problème est proche du problème de sac à dos généralisé (Multiknapsak) où le sac est la quantité à livrer et les objets sont les véhicules. Plusieurs heuristiques ont été utilisées pour résoudre un tel problème de type NP-difficile, comme les colonies de fourmis, les algorithmes génétiques, l'algorithme glouton etc...

Dans le cadre de ce travail nous avons opté pour l'algorithme glouton.

Le principe des algorithmes gloutons consiste à trier les objets (qui sont dans notre cas les véhicules) par ordre décroissant de la capacité pondérée $Cp(v)$ qui est égale à la somme du coût fixe $TF(v)$, et coût unitaire $Tu(v)$ pondéré par la capacité du véhicule $W(v)$.

Autrement dit
$$CP(v) = \frac{TF(v)+TU(v)}{w(v)}$$

Tel que :

- CP : capacité pondérée.
- TF : coût fixe de transport.

- TU : coût unitaire de transport.
- W : capacité du véhicule.
- v : véhicule v .
- ql : quantité à livrer.

L'exemple suivant illustre le principe.

v	TF	TU	W	CP
1	1.00	3.00	92	0.0435
2	0.10	4.00	94	0.0436
3	0.20	5.00	55	0.0945
4	0.40	4.00	90	0.0489
5	0.90	5.00	70	0.0843
6	0.30	5.00	69	0.0768
7	1.00	3.00	51	0.0784
8	0.10	3.00	86	0.0360

Tableau 3.1. Calcul du coût pondéré

Maintenant nous trions les CP et les ql par ordre décroissant et affectons le maximum de CP au maximum de ql pour minimiser le coût de transport.

V	TF	TU	W	CP	client	ql
3	0.20	5.00	55	0.0945	4	194
5	0.90	5.00	70	0.0843	2	160
7	1.00	3.00	51	0.0784	3	112
6	0.30	5.00	69	0.0768	1	66
4	0.40	4.00	90	0.0489		
2	0.10	4.00	94	0.0436		
1	1.00	3.00	92	0.0435		
8	0.10	3.00	86	0.0360		

Tableau.3.2. Tri du CP et ql

Quatre possibilités de comparaison sont possibles dans ce cas

1. Le Cp croissant et ql croissante.
2. Le CP croissant et ql décroissante.
3. Le CP décroissant et ql croissante.
4. Le CP décroissant et ql décroissant.

On va donner une explication pour chaque cas

a. CP croissant et ql croissant :

		i1	i2	i3	i4
V	W	66	160	112	194
8	86	66 fin			
1	92			112-92=20	
2	94			20 fin	
4	90		160-90=70		
6	69		1 fin		
7	51				194-51=143
5	70				143-70=73
3	55				73-55=18

Tableau .3.3. Affectation de la plus petite valeur du ql à la plus petite valeur de CP.

Dans le tableau 3.3, nous trouvons la quantité 18 qui ne va pas être livré. La quantité à livrer pour le client i_4 n'est pas totalement satisfaite.

b. Le CP croissant et ql décroissante.

		i1	i2	i3	i4
V	W	66	160	112	194
8	86				108
1	92		68		
2	94			18	
4	90				18
6	69		68 fin		
7	51	15			
5	70			18 fin	
3	55				18 fin

Tableau .3.4. Affectation de la plus grande valeur du ql à la plus petite valeur de CP.

Aussi dans ce cas le véhicule v_6 part quasiment vide, alors que i_4 n'est pas livré (tableau 3.4).

c. Le CP décroissant et ql croissante.

		i1	i2	i3	i4
V	W	66	160	112	194
3	55	11			
5	70	11 fin			
7	51			61	

6	69			61 fin	
4	90		70		
2	94		70 fin		
1	92				102
8	86				16

Tableau .3.5. Affectation de la plus petite valeur du ql à la plus grande valeur de CP.

De même nous voyons que i_4 n'est pas complètement livré, et v_5 part quasiment vide (tableau 3.5).

d. Le CP décroissant et ql décroissante :

CP est grand veut dire que la capacité du véhicule est petite (occuper les véhicules de petite taille à coût élevé en premier pour minimiser les coûts). De cette façon les véhicules "chers" partent avec la charge complète pour ne pas dépenser le coût de setup de transport avec une demi charge.

		i1	i2	i3	i4
V	W	66	160	112	194
3	55				139
5	70		90		
7	51				88
6	69			43	
4	90		90 fin		
2	94				88 fin
1	92	66 fin			
8	86			43 fin	

Tableau .3.6. Affectation de la plus grande valeur du ql à la plus grande valeur de CP.

Dans ce cas, nous voyons que toutes les quantités à livrer vont être transportées, et l'affectation des différents véhicules est comme suit :

Capacité w	92	94	95	90	70	69	51	86	
	v1	v2	v3	v4	v5	v6	v7	v8	ql
i1	66	0	0	0	0	0	0	0	66
i2	0	0	0	90	70	0	0	0	160
i3	0	0	0	0	0	69	0	43	112

<i>i4</i>	0	88	55	0	0	0	51	0	194
<i>Quantités à livrer qt</i>	66	88	55	90	70	69	51	43	

Tableau.3.7. Quantité transportée.

Pour résumer ces tableaux nous avons élaboré l'algorithme calcul quantité à transporter.

Algorithme 5 Calcul Quantité à Transporter

Début

$$CP(v) = (TF(v) + TU(v)) / W(v)$$

Trier CP par ordre décroissant

Ndxw index des capacités des CP

Pour *chaque période*

Trier QL par ordre décroissant

Pour $v=1, nv$ % nombre de véhicules

Calculer maxql = max(ql)

si *quantité max < capacité véhicule*

affecter la quantité au véhicule

$vv(v, i, t) = 1$ % le véhicule est occupé

sinon

affecter au véhicule sa capacité

$vv(v, i, t) = 1$

soustraire la capacité du véhicule de la quantité à livrer

fin si

fin pour

fin pour

Une fois que les quantités à produire, les quantités à livrer et les quantités à transporter sont calculées, nous pouvons dire que notre chaîne logistique est complète. Ainsi il ne nous reste qu'à l'optimiser. Nous avons appliqué les algorithmes génétiques comme indiqué au début de ce chapitre, et intégré ces trois derniers algorithmes.

3.2.4 Algorithme génétique multi chromosome:

Nous plaçons les algorithmes de réparation après la génération d'une population pour effectuer les modifications sur le multi chromosome et d'en déduire les valeurs du reste des variables. En suite nous calculons le fitness, la sélection, le croisement et enfin la mutation.

Algorithme génétique multi chromosome.

Début

Coder le chromosome

Générer la population initiale

Répter % nombre de génération

Appel calcul quantité à livrer

Appel quantité à produire

Appel quantité à transporter

Calculer le fitness

Sélection

Croisement

Mutation

Fin

Jusqu'à la fin

Fin

3.3 Implémentation et résultats numériques de l'algorithme génétique intégral :

Afin d'implémenter cet algorithme, nous avons utilisé le langage MATLAB version 9 (2011), sur une machine à processeur 2.3 GH et 4 GO de RAM. Les paramètres sont générés aléatoirement, les résultats d'une Simulation seront présentés dans ce qui suit.

3.3.1 Résultats numériques:

Dans cette section, nous présentons des exemples de calcul par la méthode exacte et notre heuristique. Nous avons développé une GUI (Graphic User Interface) avec MATLAB la (figure 3.17) montre une image écran de l'exécution.

Après l'exécution de l'application on a eu le graphe de la figure.3.18 qui présente la régression du coût en fonction des générations de l'algorithme génétique. Notre problème est un problème de minimisation, on voit bien que la courbe diminue, on déduit que l'algorithme fonctionne dans le bon sens.

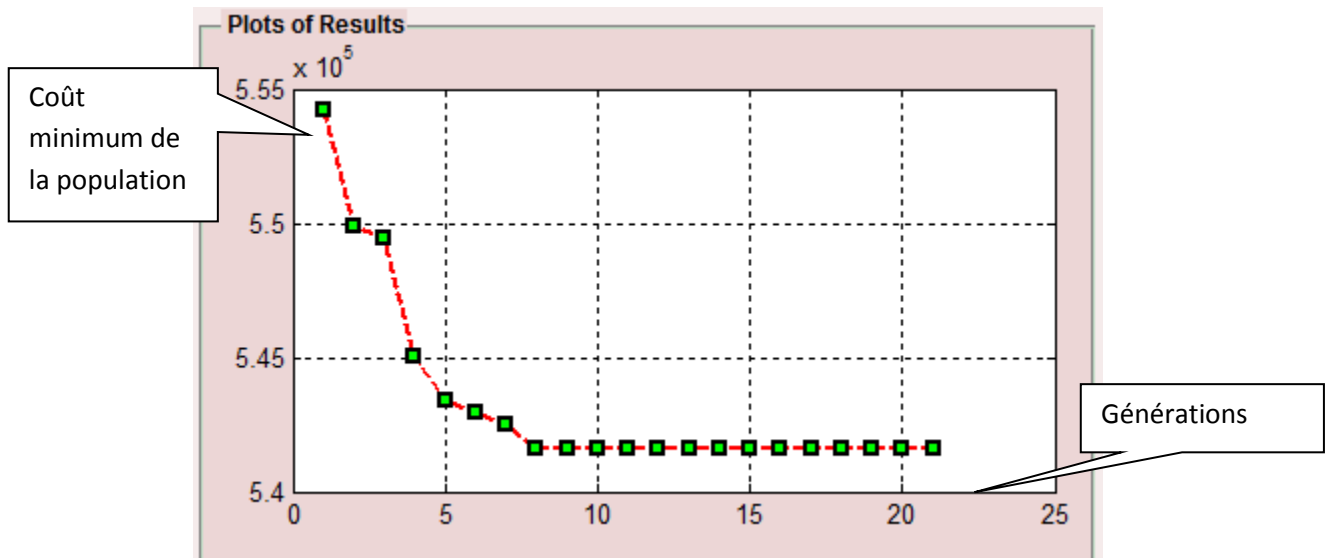
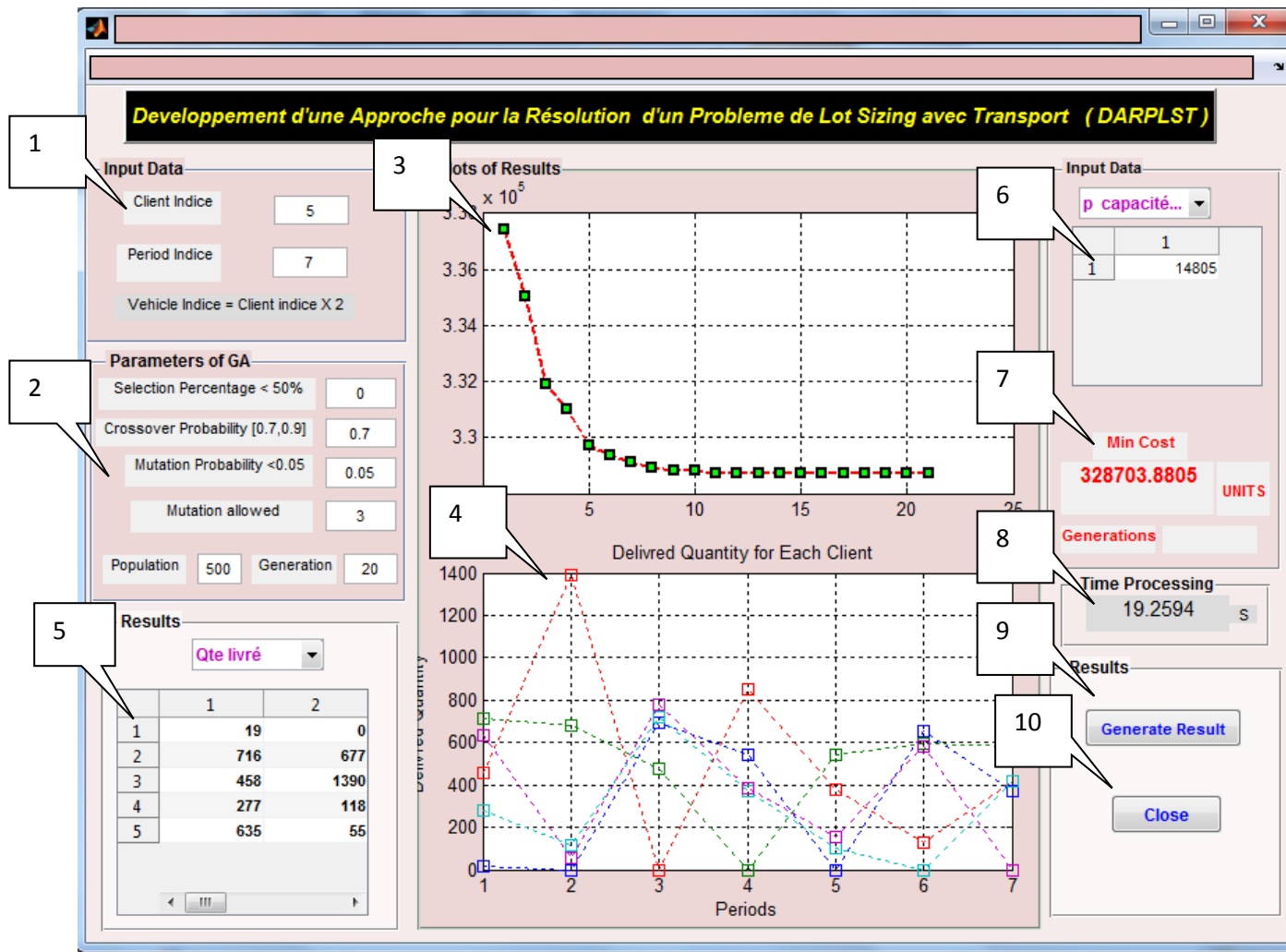


Figure.3.18. courbe du coût en fonction des générations.



1. Les variables du problème
2. Les paramètres de l’algorithme génétique.
3. Graphe de l’exécution de l’algorithme génétique.
4. Graphe des valeurs des variables de décision.
5. Table des valeurs des variables de décision.
6. Table des paramètres d’entrées.
7. Valeurs du coût minimum.
8. Temps d’exécution de l’algorithme génétique.
9. Bouton de lancement de l’algorithme génétique.
10. Bouton fermer.

Figure.3.17. Interface de l’algorithme génétique pour le problème de lot sizing avec transport.

L'étape suivante est voir le degré de validité de notre approche en faisant une étude comparative avec la solution du solveur *CPLEX*.

3.3.2 Comparaison de l'heuristique avec GAMS (Méthode exacte):

Notre recherche est une simulation qui permet de faire le plan de production d'un système semblable au notre. Afin de vérifier le degré de validité de l'algorithme évolutionnaire il a fallu basculer les mêmes paramètres vers GAMS afin d'avoir les mêmes données d'entrée pour les deux solutions. On a utilisé les mêmes paramètres pour la méthode exacte.

En utilisant les paramètres de l'algorithme génétique qui sont présentés dans le tableau 3.8 on a pu faire la simulation.

Paramètres du GA	valeurs
Population	200
Génération	20
Probabilité de mutation	0.05
Probabilité de croisement	0.7
Nombre de mutation	10

Tableau.3.8. propriétés de l'algorithme génétique.

Itér	i	t	MATLAB		GAMS		Erreur %
			<i>Coût</i>	<i>Tmp</i>	<i>Coût</i>	<i>Tmp</i>	
1	2	3	36 787.00	2 .20	35 125.00	0.25	4.73
2	4	4	475 321.62	6.50	464 892.00	0.26	2.24
3	8	5	924 298.64	6.81	890 483.00	0.26	3.80
4	10	6	756 863.03	7.12	726 382.00	0.20	4.20
5	12	7	1 809 077.23	7.60	1 741 985.00	0.30	3.85
6	14	8	3 015 339.59	9.09	2 889 644.00	0.50	4.35
7	16	9	3 874 523.31	11.00	3 795 595.93	0.70	2.08
8	18	10	2 332 398.57	15.32	2 296 532.71	0.89	1.56
9	20	11	3 885 962.24	16.65	3 803 995.41	1.23	2.15
10	22	12	7 526 902.34	18.21	7 436 158.91	2.10	1.22
11	24	13	5 488 154.36	19.83	5 313 347.75	3.10	3.29
12	26	14	7 922 471.64	25.21	7 896 458.78	3.00	0.33

13	28	15	9 302 135.66	30.10	9 141 618.41	4.05	1.76
14	30	16	8 489 162.32	34.36	8 142 114.32	4.50	4.26
15	32	17	9 578 227.58	36.00	9 127 856.37	5.97	4.93
16	34	18	1 737 943.52	38.25	1 709 840.37	9.28	1.64
17	36	19	1 557 960.00	40.58	1 530 737.36	11.00	1.78
18	38	20	1 722 353.21	55.28	1 668 987.96	13.04	3.20
19	40	21	1 882 930.00	60.52	1 817 528.00	15.38	3.60
20	50	50	6 876 659.12	80.58	6 759 755.02	94.00	1.73
21	60	60	6 525 364.14	100.00	6 230 809.52	124.00	4.73
22	70	70	113 460 100.25	480.00	108 284 790.68	870.00	4.88

Tableau.3.9. Comparaison entre la méthode heuristique et la méthode exacte.

Dans le tableau 3.9 on trouve la simulation et la comparaison entre notre approche et la méthode exacte.

3.3.3 Interprétation des résultats :

Afin de faire l'interprétation des résultats, nous nous sommes basés sur deux axes, le premier est le temps de calcul, et le deuxième est la marge d'erreur entre les deux approches. Nous avons pu constater que (tableau 3.9):

1. L'erreur entre notre approche et la méthode exacte est dans l'intervalle $[0\%, 5\%]$. Si on calcule l'erreur moyenne on trouve : 3.01%
2. Le temps de calcul est petit pour la méthode exacte et élevé pour la méthode approchée, jusqu'à la 20^{ème} itération $(ni, nt) = (60, 60)$. Au delà on trouve l'inverse (le temps de calcul de la méthode exacte devient plus important que celui de l'algorithme génétique). Pendant la simulation nous constatons la lenteur ou le blocage de l'exécution de la méthode exacte (nécessité de plus de mémoires (trop de variables intermédiaires pour résoudre le problème)).

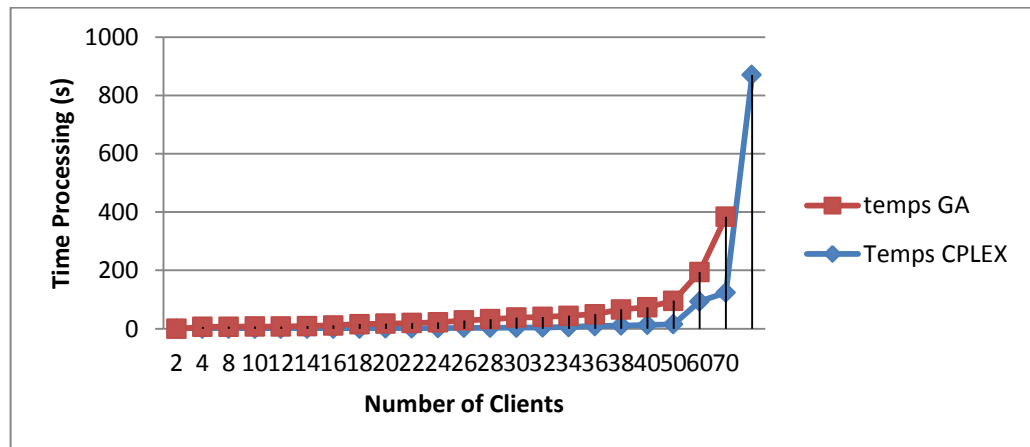


Figure 3.19. Graphe représentant le temps de calcul des deux méthodes en fonction du nombre de clients.

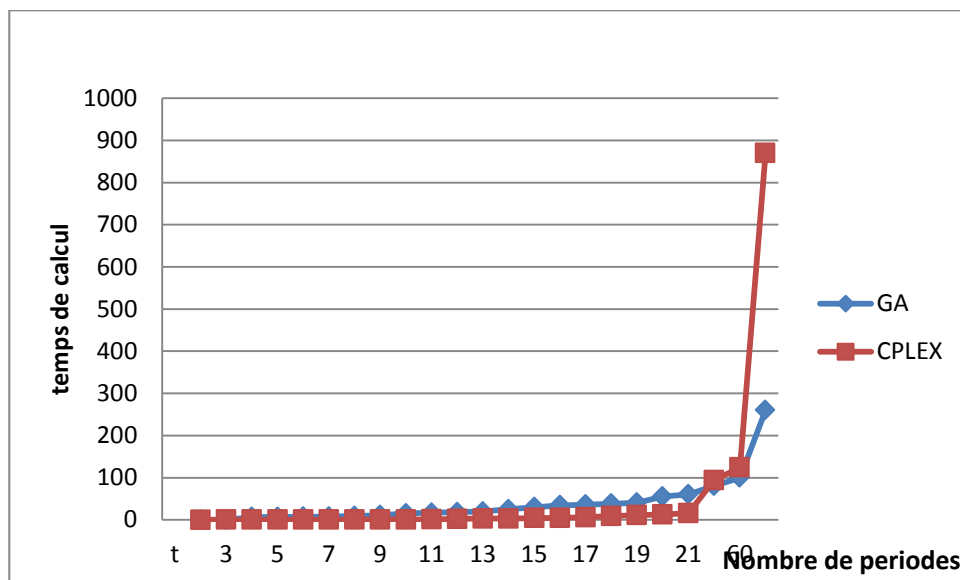


Figure 3.20. Graphe représentant le temps de calcul des deux méthodes en fonction du nombre de périodes

D'après les graphes des figures (3.19) et (3.20), le temps de calcul de la méthode exacte augmente de plus de 200% par rapport à la méthode approchée ce qui rend notre approche valable et applicable dans les entreprises de telle structure (OWMR).

Conclusion

Nous avons appliqué l'algorithme génétique pour résoudre notre problème avec un codage binaire des chromosomes, de plus il nous a fallu se servir de trois autres heuristiques pour satisfaire les contraintes, et calculer les variables de décision de la fonction objective. Ces trois heuristique sont le zéro Switch pour calculer la quantité à livrer, le zéro Switch pour calculer la

quantité à produire et en fin l'algorithme glouton pour calculer la quantité à transporter par chaque véhicule.

A l'aide de l'étude comparative entre notre approche et la méthode exacte, nous constatons que les résultats obtenus montrent que notre approche devient la mieux adaptée pour résoudre le problème de lot sizing avec transport..

Conclusion générale

Le travail présenté dans ce mémoire concerne le développement d'une approche en utilisant les algorithmes génétiques pour résoudre un problème de lot sizing avec transport à capacité fini. Ce travail est divisé en trois chapitres.

Le premier chapitre est une vision générale des concepts de base relatifs au problème de lot sizing.

Le second présente d'une part la structure particulière de la chaîne logistique, qui est formée d'une unité de production qui dépose son produit dans un dépôt central à capacité infini. Ce produit est ensuite expédié vers des clients dispersés géographiquement à l'aide d'une flotte de véhicule hétérogène et d'autre part sa modélisation mathématique et l'utilisation d'une solution exacte à l'aide du solveur CPLEX sous GAMS. Avec la méthode exacte on constate que le temps de calcul devient trop important si le nombre de client et de période augmente.

Aussi nous avons présenté dans le troisième chapitre un algorithme génétique à codage binaire comme heuristique à la solution de ce problème. Vu la complexité de la structure utilisée, une décomposition modulaire et une utilisation des algorithmes de réparations a été nécessaire pour nous faciliter l'utilisation des algorithmes génétiques.

Nous avons fait des simulations de l'AG sur une population de 200 individus et 20 générations avec une sélection par la roulette de wheel, un croisement à un point avec une probabilité de 0.7 et une mutation multipoint (10 points de mutation) avec une probabilité de 0.05. Les paramètres de la fonction objectif sont générées aléatoirement ce qui rend notre approche applicable pour les structures de ce genre. Les résultats obtenus ont montré que le temps de calcul dépend du nombre des clients, des périodes et des véhicules. De plus on a comparé notre approche avec la méthode exacte. On a remarqué qu'aux premières itérations le temps de calcul est considérablement grand pour l'AG. Une fois que $(n_v, n_i, n_t) = (120, 60, 60)$ le traitement pour CPLEX devient gourmand en temps et en mémoire, par rapport à l'AG qui a donné des résultats acceptables arrivant à une erreur de 0.3% et un temps d'exécution très réduit par rapport à celui de la méthode exacte.

Notre approche peut être améliorée en ajoutant d'autres heuristiques pour diminuer l'erreur et s'approcher de l'optimum comme la recherche locale. De plus on peut enrichir le modèle, en ajoutant des hypothèses comme la capacité du dépôt central, produire plusieurs produits au lieu d'un seul produit, varier les coûts de setup et les coûts unitaires de

production, de stockage et de transport, dans chaque période. De plus utiliser une flotte de véhicule homogène. Utiliser une livraison avec tourné de véhicule au lieu d'une livraison directe, et intégrer le problème VRP avec le problème de lot sizing.

Références Bibliographique

- [1] K.Ghdira, «Logistique de la production: Approches de modélisation et de résolution», Edition Technip 2007.
- [2] C.P.Vincent, « Heuristique - Création, intuition, créativité et stratégies d'innovation », BOD - Books on Demand France, 2012.
- [3] V.Cerny, « Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm », J. of Optimization Theory and Applications, 1985.
- [4] C.Wolosewicz, « Approche intégrée en planification et ordonnancement de la production ». Thèse doctorat de l'Ecole Nationale Supérieure des Mines de Saint Etienne, 2008.
- [5] S.E.Merzoug . « Problème de dimensionnement de lot de livraison: application au cas d'une chaîne logistique ». Thèse de doctorat Université de Technologie de Belfort-Montbéliard et de l'Université de Franche-Comte, 2007.
- [6] H.M.Wagner, T.M.Within, « Dynamic version of the economic lot size model», Management Science, 1958.
- [7] N.Brahimi, « Planification de la production: Modèles et algorithmes pour le problème de dimensionnement de lot » , Thèse de doctorat , Université de Nantes, 2004.
- [8] E.Arkin, D.Joneja, R.Roundy, « Computational complexity of uncapacitated multi-echelon production planning problems ». Operations Research Letters,1989.
- [9] E.Taillard, «Introduction aux méta-heuristiques », Cours 3^{ème} cycle informatique, Haute Ecole d'Ingénierie et de Gestion du Cantone De Vaud, Genève, 2007.
- [10] S. Kirkpatrick, C. Gelatt , M. Vecchi, «Optimization by simulated annealing», JSTOR, 1983.
- [11] S.Amédée, R. F.Gérard, « les Algorithmes génétiques» TE de fin d'année, Université Nice Sophia Antipolis, 2004.
- [12] M. Dorigo, T. Stützle, «Ant colony optimization », The Massachusetts Institute of Technology, 2004.
- [13] Harvey M.Wagner, Thomson M. Within, «Dynamic version of the Economic Lot size Model», Management Science, 1958.
- [14] S.Mahdi,«Optimisation multi objectif par un nouveau schéma de coopération méta/exacte » , Mémoire de magister université Mentouri Constantine ,2007.

- [15] T. Baeck, D. B. Fogel, Z. Michalewicz, «Evolutionary computation» », Institute of Physics Publishing, 2000.
- [16] A.Akbalik, « Optimisation de la gestion intégrée des flux physiques dans une chaîne logistique ; extension du problème de dimensionnement de lot», Thèse de doctorat. Institut National Polytechnique de Grenoble 2006.
- [17] M. Elkhamri, « L'intégration de la planification et l'ordonnancement», Projet de fin d'étude l'École des Mines de St-Étienne & l'université Jean Monnet , 2009-2011.
- [18] N.Nananukul. «Lot-sizing and inventory routing for a production-distribution supply chain ». Thèse de doctorat. the university of texas at austin may 2008.
- [19] N.Grangéon, S.Norre , O.G.Guechon, M.suon, «Un problème de planification stratégiques de type production distribution avec économie d'échelle et technologie de production». 8^{ème} Conférence Internationale de Modélisation et Simulation MOSIM 10 -10 au 12 Mai 2010 Hammamet Tunisie.
- [20] N.Brahimi, B.Nadji, S.Dauzere-Peres and A.Nordli, «Etat de l'art sur les problèmes de dimensionnement des lots avec contraintes de capacité ». 4^{ème} Conférence Francophone de Modélisation et simulation. Organisation et Conduite d'Activités dans l'Industrie et les Services, MOSIM 03 du 23 au 25 avril 2003 Toulouse France.
- [21] N.Brahimi, S.Dauzere perese, N.Najid, and A.Nordli. «Single lot sizing problems», European Journal of Operational Research, 2006.
- [22] J.Morana, «de la logistique d'entreprise au supply chain management (scm) : Vers une integration des processus» . Reference E-Theque, 2003.
- [23] Harris, W.Ford « How Many Parts to Make at Once ». The Magazine of Management, 1913
- [24] B. Karimi, S.M.T. Fatemi Ghomi, J.M. Wilson, «The capacitated lot sizing problem: A review of models and algorithms », OMEGA 2003.
- [25] R.Levi, R.Roundy, D.Shmoys, M.Sviridenko, «A Constant Approximation Algorithm for the One-Warehouse Multi retailer Problem », Management Science, 2008.
- [26] J.-F.Scheid, «Graphe et recherche opérationnelle, chapitre 9 : Introduction aux méthodes heuristiques », ESIAL , 2012-2013.

- [27] J.C. Lang, «Production and Inventory Management with Substitutions », Springer Edition 2010.
- [28] O.Solyalı, H.Süral, «The one-warehouse multi-retailer problem: reformulation, classification, and computational results », Annals of Operations Research Springer Science Business Media, 2011.
- [29] Y.Jin, A.Muriel, «Single-warehouse multi-retailer inventory systems with full truckload shipments », Naval Research Logistics (NRL), 2009.
- [30] M.Hariga , M.Gumus , A.Daghfous , S.K.Goyal, «A vendor managed inventory model under contractual storage agreement », Computers & Operations Research, 2013.
- [31] C.Monthatipkul, P.Yenradee, «Inventory/distribution control system in a one-warehouse/multi-retailer supply chain », Int. J. Production Economics, 2008.
- [32] S.H.R.Pasandideh, S.T.A.Niaki, N.Tokhmehchi, « A parameter-tuned genetic algorithm to optimize two-echelon continuous review inventory systems», Expert Systems with Applications , Elsevier 2011.
- [33] P.Kamesky, D.Simshi « Production and distribution lot sizing in a two stage supply chain», IIE Transactions , 2003.
- [34] W.Jaruphongsaa, S.Cetinkayab, C.Yee Leec, «Warehouse space capacity and delivery time window considerations in dynamic lot-sizing for a simple supply chain », International Journal of Production Economics , 2004.
- [35] P.Chandra, « A Dynamic Distribution Model with Warehouse and Customer Replenishment Requirements », Operational Research Society Ltd, 1993.
- [36] F.T.Abdelmaguid, M.M.Dessouky, «A genetic algorithm approach to the integrated inventory distribution problem » , International Journal of Production Research, 2006.
- [37] Jonathan F.Bard , Narameth Nananukul, « A branch-and-price algorithm for an integrated production and inventory routing problem», Journal of Computers and Operations Research, 2010.
- [38] H.Liberalino, C.Duhamel, A.Quilliot, S.Kedad-Sidhoum , P.C. Crossref, «The integrated lot sizing and vehicle routing problem », Computational Intelligence In Production And Logistics Systems (CIPLS), 2011.
- [39] W. Yang , Felix T.S. Chan, V. Kumar, «Optimizing replenishment polices using genetic algorithm for single-warehouse multi-retailer system», Journal of Expert Systems with

Application, 2012.

- [40] C.Guo , X.Yang , «A programming of genetic algorithm in matlab7.0», Modern Applied Science, 2011
- [41] R. L. Haupt, S.E. Haupt, « Practical genetic algorithms », 2^{ème} Edition, Wiley-Interscience, 2004.
- [42] J.H Lorie et J.Savage. «Three problems in capital rationing», Journal of Business, 1955

DEVELOPPEMENT D'UNE APPROCHE POUR LA RESOLUTION D'UN PROBLEME DE LOT SIZING AVEC TRANSPORT

Résumé :

Le travail présenté dans le cadre de ce mémoire est relatif à un problème de lot sizing. L'objectif consiste à optimiser le coût total de la chaîne, constituée d'un coût de production, de stockage et de transport. Le problème a été d'abord résolu par une méthode exacte ; qui a montrée ses limites si le nombre de client et la période augmentent. Aussi nous avons proposé l'utilisation des algorithmes génétiques comme heuristique à la solution de ce problème. Les résultats obtenus sont satisfaisants.

Mots clefs : Lot sizing, Chaîne logistique, Algorithme génétique.

Development Of An Approach To Solve A Lot Sizing Problem With Transport

Abstract :

The work presented is about a lot sizing problem. The objective is to optimize the total cost of the supply chain, consisting of a production cost, storage cost and transport cost. First the problem was solved by an exact method, which has shown its limits if the number of customers and periods increase. Also we proposed the use of genetic algorithms as a heuristic to solve this problem. The results are satisfactory.

Key words : Lot Sizing, Supply chain, Genetic Algorithm.

تطوير منهجية من اجل حل مسألة تحجيم دفعات الإنتاج مع النقل

ملخص :

العمل المقدم في هذه المذكرة حول مسألة تحجيم دفعات الإنتاج. الهدف هو تحسين التكلفة الإجمالية للسلسلة اللوجيستية ، والتي تتألف من تكلفة الإنتاج والتخزين والنقل. في البداية تم حل المشكلة بواسطة طريقة دقيقة ، التي أظهرت حدودها إذا زاد عدد العملاء و فترات الإنتاج. كما اقترحنا استخدام الخوارزميات الجينية كطريقة مقربة ومهجنة لحل هذه المسألة. وكانت النتائج مرضية.

الكلمات الدلالية: تحجيم دفعات الإنتاج . سلسلة لوجيستية . خوارزمية جينية.