

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE MATHÉMATIQUES ET INFORMATIQUE

DEPARTEMENT INFORMATIQUE

N° :



DOMAINE : Mathématiques Et Informatique

FILIERE : Informatique

OPTION : Informatique Décisionnelle Et Optimisation

**Mémoire présenté pour l'obtention
Du diplôme de Master Académique**

Par: Aouadj Ayache

Intitulé

**Résolution du problème de tournées de
véhicules avec fenêtre de temps**

Soutenu devant le jury composé de :

Mousaoui Adel

Université de M'sila

Président

Bouzaroura Ahlem

Université de M'sila

Rapporteur


Bounif Mohamed El Hadi

Université de M'sila

Examineur

Année universitaire : 2018 /2019

Remerciements

 Je tiens tout d'abord à remercier Dieu le tout-puissant et Le Miséricordieux, qui m'a donné la vie et la force ainsi que le courage et l'audace pour dépasser toutes les difficultés, d'avoir suivis et accomplis toutes les étapes de mes études .

En second lieux je tiens à remercier mon encadreur Mme. AHLEM BOUZAROURA pour m'avoir assisté et m'aides par leurs précieux conseils et de m'avoir offerts toutes les possibilités telle que suivent et prestation de documentations nécessaires pour la réalisation de mon mémoire de fin d'étude .

Enfin je tiens aussi à remercier tous mes amis et collègues d'étude qui mon aider de près ou de loin d'avoir réalisé ce travail.

A mes chères collègues d'études et frères de cœur.

Table des matières

INTRODUCTION GENERALE	1
------------------------------------	---

CHAPITRE 01 : LES CHAINES LOGISTIQUES

1	Introduction.....	4
2	Historique de la chaîne logistique.....	4
3	Définitions de la chaîne logistique.....	5
4	Gestion de la chaîne logistique.....	6
5	Structure de la chaîne logistique.....	7
	• Chaîne logistique divergente.....	7
	• Chaîne logistique convergente.....	7
	• Chaîne logistique réseau.....	7
	• Chaîne logistique séquentielle (ou linéaire).....	7
6	Les flux de la chaîne logistique.....	7
	6.1 Flux physiques.....	8
	6.2 Flux financiers.....	8
	6.3 Flux d'information.....	8
7	Rôles d'une chaîne logistique.....	9
	7.1 L'approvisionnement de matières.....	10
	7.2 La production de biens ou de services.....	10
	7.3 Le stockage de matières.....	10
	7.4 Distribution et transports.....	11
	7.5 La vente.....	11
8	Conclusion.....	11

CHAPITRE 02 : L'OPTIMISATION COMBINATOIRE

1	Introduction.....	13
2	Définition d'un problème d'optimisation combinatoire :.....	13
3	La complexité.....	14
	3.1 Les complexités d'un algorithme.....	14
	3.2 La complexité problématique.....	14
4	Quelques problèmes classiques d'optimisation combinatoire.....	16
	4.1 Problème du sac-à-dos.....	16

4.2	Problème d'affectation.....	16
4.3	Problème d'ordonnancement.....	16
4.4	Problème du voyageur de commerce :.....	17
5	Les méthodes d'optimisation combinatoire :.....	17
5.1	Les méthodes exactes.....	18
5.2	Les méthodes approchées.....	19
5.2.1	Les méthodes Heuristiques.....	19
5.2.2	Les méthodes méta-heuristiques.....	19
6	Conclusion.....	30

CHAPITRE 03 : PROBLEME DE TOURNEES DE VEHICULES AVEC FENETRE DE TEMPS

1	Introduction.....	32
2	Le problème de tournées de véhicules.....	32
2.1	Formulation du VRP.....	33
3	Le Problème de tournées de véhicules avec fenêtre de temps.....	35
3.1	Formulation du PTVFT.....	37
4	Problème de tournées de véhicules et ses variantes.....	38
4.1	PVC (Problème Du Voyageur De Commerce).....	38
4.2	PTVC (Problème De Tournées De Véhicules Avec Capacité).....	39
4.3	PTVD (Problème De Tournées Des Véhicules Dynamique).....	39
4.4	VRPB (Prolongation Du VRP Classique).....	39
4.5	PTVMP (Problème De Tournées De Véhicules Multi Périodique).....	39
4.6	PTVS (Problème De Tournées De Véhicules Stochastique).....	40
5	Conclusion.....	40

CHAPITRE 04 : IMPLEMENTATION ET RESULTATS EXPERIMENTAUX

1	Introduction.....	42
2	L'objectif du travail.....	42
3	Description de l'approche proposée.....	42
3.1	Contraintes du problème de tournée de véhicule avec fenêtre de temps.....	42
3.2	Paramètres du problème.....	43
3.3	Paramètres d'algorithme génétique.....	43
4	Environnement matériel.....	45

5 Environnement de développement.....	45
6 Expérimentation.....	46
7 Quelques tests sur l'implémentation.....	48
8 Discussion des résultats	50
9 Conclusion	50
Conclusion générale	51
Bibliographie.....	52

Liste des Figures

CHAPITRE 01 : LES CHAINES LOGISTIQUES

Figure 1.1: Exemple d'une chaîne logistique	6
Figure 1.2: Structure de la chaîne logistique	7
Figure 1.3: Flux de la chaîne logistique	9

CHAPITRE 02 : L'OPTIMISATION COMBINATOIRE

Figure 2.1: La relation entre la complexité des problèmes	15
Figure 2.2: Les méthodes de résolution des problèmes d'optimisation combinatoire.....	17
Figure 2.3: Principe général d'un algorithme génétique.....	26
Figure 2.4: Codage des solutions.....	26
Figure 2.5: Sélection par roulette	27
Figure 2.6: Sélection par tournoi	28
Figure 2.7: Croisement a un point	28
Figure 2.8: Une mutation.....	29

CHAPITRE 03 : PROBLEME DE TOURNEES DE VEHICULES AVEC FENETRE DE TEMPS

Figure 3.1: Présentation graphique du problème de tournées de véhicules VRP.....	33
Figure 3.2: Exemple de fenêtre de temps sur un sommet.....	36

CHAPITRE 04 : IMPLEMENTATION ET RESULTATS EXPERIMENTAUX

Figure 4.1: Exemple d'une solution	44
Figure 4.2: Le code source de la fonction fitness	44
Figure 4.3: Langage de programmation Python	45
Figure 4.4: L'interface de l'application console	46
Figure 4.5: L'exécution de l'application console	47
Figure 4.6: Fenêtre graphique de l'exécution	47
Figure 4.7: L'influence de la taille de population sur la solution optimale.....	49
Figure 4.8: L'influence de nombre d'itérations sur la solution optimale.....	50

Liste des tableaux

CHAPITRE 04 : IMPLEMENTATION ET RESULTATS EXPERIMENTAUX

Tableau 4.1: Expérimentation pour mesurer l'influence de nombre d'itérations..... 48

Tableau 4.2: Expérimentation pour mesurer l'influence de la taille de populations 49

Liste des algorithmes

CHAPITRE 02 : L'OPTIMISATION COMBINATOIRE

Algorithme 2.1: L'algorithme de la méthode de descentes 21

Algorithme 2.2: L'algorithme de recuit simulé 22

Algorithme 2.3: L'algorithme de la recherche tabou..... 23

Algorithme 2.4: L'algorithme de colonies de fourmis pour le TSP 24

Liste des abréviations

AG	Algorithme Génétique
IA	Intelligence Artificielle
IDE	Integrated Development Environment
KSP	KnapSack Problem
OC	Optimisation Combinatoire
PL	Programmation Linéaire
PTVFT	Problème De Routage De Véhicule Avec Fenêtre De Temps
RO	Recherche Opérationnelle
SCM	Supply Chain Management
STW	Soft Time Window
TIC	Technologie De L'information Et De La Communication
TW	Time Window
VRP	Vehicule Routing Problem
VRPTW	Vehicule Routing Problem With Time Window

INTRODUCTION GENERALE

L'être humain est confronté à différents problèmes dans toutes les sphères de la société et dans tous les domaines. Un problème donné peut être défini par l'ensemble des propriétés que doivent vérifier ses solutions. Il peut être un problème de décision ou un problème d'optimisation. Un problème de décision peut se ramener à un problème d'existence de solution. Par contre, un problème d'optimisation peut se ramener à un problème d'existence de solution de bonne qualité. Il consiste à parcourir l'espace de recherche afin d'en extraire une solution optimale parmi un ensemble fini de solutions, la résolution d'un problème d'optimisation nécessite l'utilisation d'un procédé algorithmique permettant la maximisation ou la minimisation d'une ou de plusieurs fonctions « objectif » en respectant les contraintes posées par le problème.

Les méthodes de résolution de problèmes ont été classées en deux catégories : les méthodes exactes et les méthodes approchées. Et ces dernières sont classées en deux catégories : les heuristiques et les méta-heuristiques. Une heuristique est une méthode approchée spécifique à un problème donné. Elle forme un ensemble de règles empiriques ou des stratégies qui fonctionnent, les méta-heuristiques sont des méthodes approchées polyvalentes, elles peuvent être appliquées sur de nombreux problèmes.

Le domaine dans lequel s'inscrit notre travail est l'optimisation combinatoire, il concerne l'élaboration des tournées de véhicules avec fenêtre de temps PTVFT (VRPTW : Vehicle Routing Problem With Time Window). C'est un problème très important qui consiste à déterminer les routes d'une flotte de véhicules afin de satisfaire un ensemble de clients en tenant compte d'un ensemble de contraintes et en réduisant les coûts (distance, temps...) pour que les véhicules puissent visiter exactement une fois chaque client géographiquement distribué dans un intervalle de temps bien déterminé.

Le PTVFT est un sous problème important dans le domaine des systèmes de distribution. Pour la résolution de ce problème on a choisi les algorithmes génétiques comme solution.

Le mémoire présenté est donc divisé en quatre chapitres. Le premier chapitre décrit le domaine d'étude des chaînes logistiques. Nous définissons tout d'abord les notions de chaîne logistique et de Supply Chain Management (SCM). A travers cela, nous définissons les types de la chaîne logistique.

Dans le second chapitre, nous présentons l'optimisation combinatoire et nous définissons l'historique et la notion de base d'optimisation, Ensuite Les étapes de la méthode générale de

résolution des problèmes jusqu'à les méthodes de résolution des problèmes d'optimisation (exacte et approchée).

Le troisième chapitre on introduit le problème de tournée des véhicules et le PTV avec fenêtre de temps et ses mécanismes. Le quatrième chapitre présent la partie d'implémentation de notre problème ainsi des résultats expérimentaux.

Finalement, nous clôturons ce mémoire par une conclusion générale.

CHAPITRE 01
LES CHAINES LOGISTIQUES

1 Introduction

De nos jours, les entreprises doivent s'adapter à la dynamique du marché pour espérer survivre dans un environnement très concurrentiel et très compétitif. Les clients sont beaucoup plus exigeants en raison des nombreux choix qui s'offrent à eux. La satisfaction des clients combinés à la réduction des coûts et produits moins chers et de haute qualité livrés dans les délais et avec un excellent service après-vente sont difficiles à réaliser.

La supply chain est un terme anglo-saxon désignant la chaîne logistique globale, celle qui va du fournisseur au client et où la production est tirée par la demande ; c'est-à-dire le marché. Son objectif est : « le bon produit en quantité correcte et dans de bonnes conditions, au bon endroit, au bon moment, au bon client, au coût le plus juste ». Elle désigne en réalité, l'ensemble des entreprises ou organisations qui interviennent pour qu'un produit donné soit fabriqué, acheminé et réceptionné dans de bonnes conditions par le client final. [1]

2 Historique de la chaîne logistique

Le dictionnaire de l'Académie française donne «la science du calcul» comme premier sens au mot «logistique». Étymologiquement, le terme «logistique» provient du grec *logistikos*, ce qui est «relatif à l'art du raisonnement». Platon est notamment cité comme le premier à avoir utilisé le mot *logistikos* pour opposer le calcul pratique (logistique) à l'arithmétique théorique. La logistique est encore largement empreinte de cette acception puisque de nombreux travaux académiques ainsi que de nombreuses préoccupations des entreprises se centrent sur l'optimisation (de tournées, de chargement de palette, des coûts...).

Le terme «logistique» trouve ensuite son origine dans le milieu militaire et provient du grade d'un officier en charge du «logis» des troupes, lors du combat. Napoléon le premier met en place un encadrement de l'approvisionnement en vivres et en munitions. Ainsi, le grade de «major général des logis» fut donné à «un officier qui avait la fonction de loger ou de camper les troupes, de diriger les colonnes, de les placer sur le terrain». Le logisticien militaire avait en charge le transport, le ravitaillement et le campement des troupes.

Le terme «logistique» est entré dans le langage courant à partir du début des années quarante, lorsque la première guerre du Golfe a éclaté. La presse a démocratisé le mot en parlant de «soutien logistique» dans le cadre d'actions militaires ou humanitaires. Il est classiquement reconnu qu'une des principales organisations logistiques à caractère militaire du XXe siècle fut la coordination du débarquement des troupes alliées en Normandie en juin 1944. Le savoir-faire acquis s'est alors diffusé dans les entreprises, d'abord aux États-Unis,

ensuite dans les pays européens. Le développement de la fonction logistique au sein des entreprises européennes est également lié à un contexte économique qui en a amené l'émergence. [2]

3 Définitions de la chaîne logistique

Plusieurs définitions de la chaîne logistique et de gestion de la chaîne logistique sont passées en revue de la littérature, classées et synthétisées. Le but de cette section est de présenter une brève des définitions de la chaîne logistique. En effet, Il semble qu'il existe un certain consensus entre les auteurs sur la définition de "chaîne logistique", ce qui n'est pas le cas pour « management de la chaîne logistique ».

Le terme *supply chain* est le plus souvent traduit en français par chaîne logistique. Les termes chaîne logistique étendue et chaîne d'approvisionnement sont également parfois utilisés.

Une chaîne logistique est souvent représentée comme une chaîne reliant le fournisseur du fournisseur au client du client.

En réalité, les chaînes logistiques n'ont de chaîne que leur nom. Elles sont constituées d'un réseau complexe d'organisations dont la figure 1.1 reste une illustration simplifiée. [3]

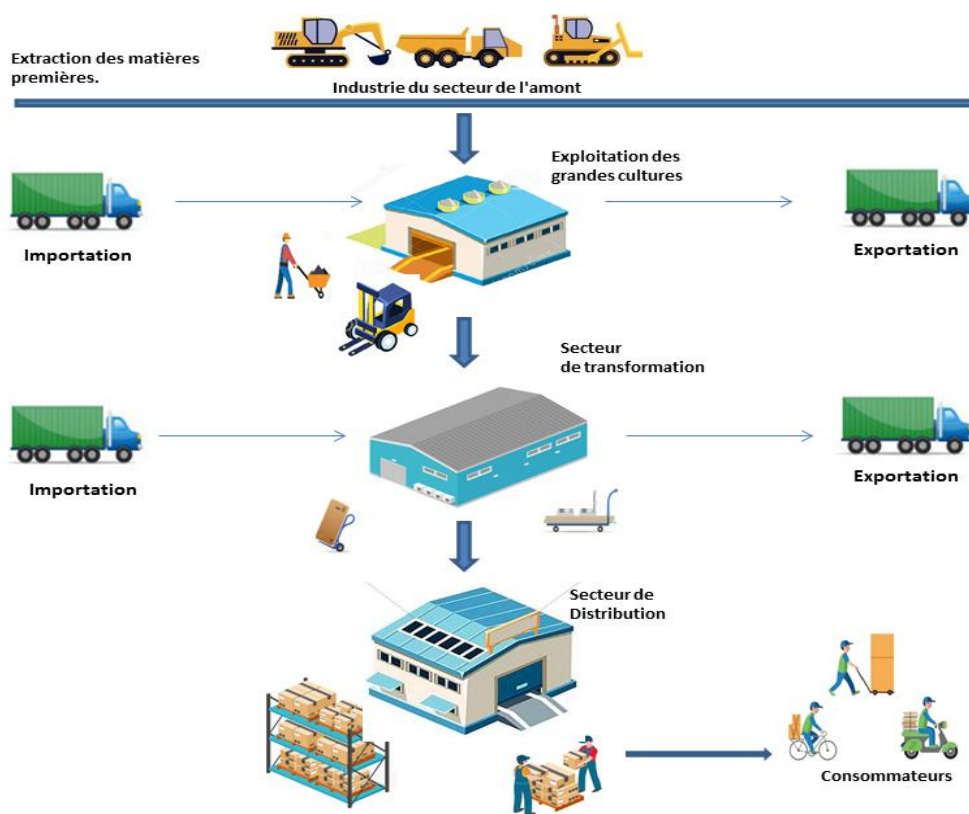


Figure 1.1 Exemple d'une chaîne logistique.

4 Gestion de la chaîne logistique

Une chaîne logistique existe dès lors qu'au moins deux entreprises travaillent sur l'achèvement d'un produit donné. Si et seulement si cette association est délibérément pilotée en vue d'en maximiser la performance, alors on peut parler de gestion de la chaîne logistique. Il existe ainsi une distinction entre la « chaîne logistique » et la « gestion de la chaîne logistique » (SCM-Supply Chain Management) [4].

La gestion de la chaîne logistique englobe la gestion des approvisionnements et des marchandises depuis les fournisseurs de matières premières jusqu'au produit fini (et aussi de son éventuel recyclage). La gestion de la chaîne logistique se focalise sur la façon dont les entreprises utilisent les processus, la technologie et l'aptitude à améliorer la compétitivité de leurs fournisseurs. C'est une philosophie de management qui prolonge les activités classiques intra-entreprise, rassemblant l'ensemble des partenaires commerciaux avec un but commun d'optimisation et d'efficacité [7].

5 Structure de la chaîne logistique

La structure d'une chaîne logistique dépend évidemment de sa nature et des objectifs souhaités. Lors de sa conception, plusieurs architectures ont été développées, du point de vue flux physique, elles peuvent être classées de la forme suivante :

- **Chaîne logistique divergente** : la matière part d'un point unique et se distribue à travers la chaîne. Cela concerne par exemple l'industrie minière.
- **Chaîne logistique convergente** : la matière qui circule entre les sites converge vers un seul et même site qui est logiquement le lieu d'assemblage final. L'industrie navale ou encore aéronautique sont des bons exemples de ce type de chaîne.
- **Chaîne logistique réseau** : C'est la combinaison des deux structures précédentes : est la juxtaposition d'une chaîne convergente et d'une chaîne divergente.
- **Chaîne logistique séquentielle (ou linéaire)** : Chaque entité de la chaîne alimente une seule autre entité en aval. [6]

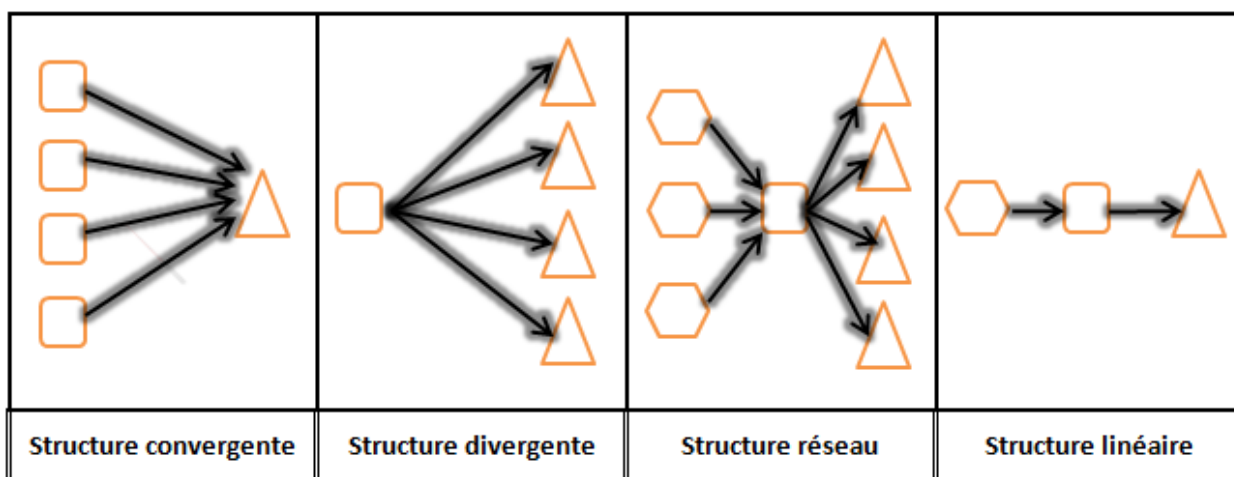


Figure 1.2 Structure de la chaîne logistique. [6]

6 Les flux de la chaîne logistique

Une chaîne logistique efficace repose sur trois types de flux : flux physique, flux d'information, flux financiers.

6.1 Flux physiques

Le flux physique est constitué par le mouvement des marchandises transportées et transformées depuis les matières premières jusqu'aux produits finis en passant par les divers stades de produits semi-finis. Il justifie l'organisation d'un réseau logistique, c'est-à-dire les différents sites avec leurs ressources de production, les moyens de transports pour relier ces sites et les espaces de stockage nécessaires pour pallier les aléas et faire tampon entre deux activités successives. En bref, l'écoulement du flux physique résulte de la mise en œuvre de diverses activités de manutention et de transformation des produits quel que soit leur état.

Le flux physique est généralement considéré comme étant le plus lent des trois flux.

6.2 Flux financiers

Le flux financier concerne toute la gestion pécuniaire des entreprises : ventes des produits, achats de composants ou de matières premières, mais aussi des outils de production, de divers équipements, de la location d'entrepôts, ... et bien sûr du salaire des employés. Le flux financier est généralement géré de façon centralisée dans l'entreprise dans le service financier ou comptabilité, en liaison toutefois avec la fonction production par les services achats et le service commercial. Sur le long terme, il correspond aussi aux investissements lourds tels que la construction de nouveaux bâtiments et de lignes de fabrication. Encore s'agit-il d'échanges avec des organismes bancaires extérieurs au réseau d'entreprises.

6.3 Flux d'information

Le flux d'information représente l'ensemble des transferts ou échanges de données entre les différents acteurs de la chaîne logistique. Il s'agit en premier lieu des informations commerciales, notamment les commandes passées entre clients et fournisseurs. Une commande comprend généralement la référence du produit, la quantité commandée, la date de livraison souhaitée et le prix éventuellement négocié lors de la vente. D'autres éléments peuvent s'ajouter à cette liste : la liste des options désirées pour le produit, la fréquence de livraison si besoin, ... Mais les entreprises s'échangent aussi des informations plus techniques: paramètres physiques du produit, gammes opératoires, capacités de production et éventuellement de transport, informations de suivi des niveaux de stock. Ces dernières sont de plus en plus réclamées par les clients qui souhaitent connaître l'état d'avancement de fabrication de leur produit. De manière plus générale, le principe de traçabilité se traduit par un droit de regard accru du client envers le fournisseur.

Le flux d'information est de plus en plus rapide grâce aux progrès des TIC. Le développement des flux d'information au sein de la chaîne logistique trouve ses limites dans le besoin de confidentialité entre acteurs. Par ailleurs, le problème de la qualité des données véhiculées subsiste, et le risque existe que des décisions soient basées sur des données erronées ou simplement périmées. [4]

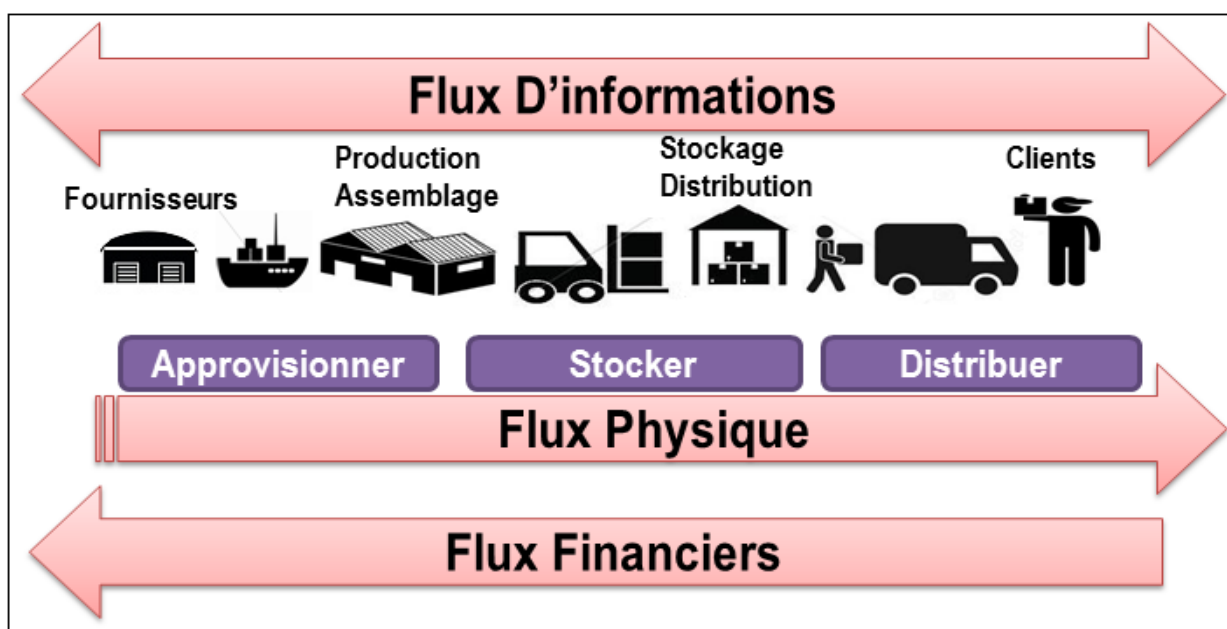


Figure 1.3 Flux de la chaîne logistique.

7 Rôles d'une chaîne logistique

Donnent une définition de la chaîne logistique qui apporte des fonctions de la chaîne logistique : « une chaîne logistique est le réseau des moyens de production et de distribution qui assure les tâches d'approvisionnement en matières premières, la transformation de ces matières premières en produits semi-finis et en produits finis, et la distribution de ces produits finis aux clients » Plus généralement, les fonctions d'une chaîne logistique vont de l'achat des matières premières à la vente des produits finis en passant par la production, le stockage et la distribution. Nous pouvons résumer ces rôles en cinq points :

- L'approvisionnement de matières.
- La production de biens physiques et de services.
- Le stockage de matières.
- La distribution et le transport.
- La vente aux clients.

7.1 L'approvisionnement de matières

Il constitue la fonction amont de la chaîne logistique. Dans une majorité d'entreprises, les matières et les composants approvisionnés constituent de 60% à 70% des coûts des produits fabriqués. Réduire les coûts d'approvisionnement contribue à réduire les coûts des produits finis, et ainsi à avoir plus de marges financières. Les délais de livraison des fournisseurs et la fiabilité de la distribution influent plus que le temps de production sur le niveau de stock ainsi que la qualité de service de chaque fabricant.

7.2 La production de biens ou de services

La fonction de production est au cœur de la chaîne logistique, il s'agit là des compétences que détient l'entreprise pour fabriquer, développer ou transformer les matières premières en produits ou services. La tendance actuelle est de migrer d'une production monosite vers un système de production multi sites. Ainsi, les actions logistiques deviennent de plus en plus prépondérantes dans ces activités industrielles.

7.3 Le stockage de matières

Le stockage inclut toutes les quantités stockées tout au long du processus en commençant par le stock de matières premières, le stock des composants, les stocks des en-cours et finalement le stock des produits finis. Les stocks sont partagés entre les différents acteurs : les fournisseurs, les producteurs et les distributeurs. Ici aussi se pose la question de l'équilibre à

trouver entre une meilleure réactivité et la réduction des coûts. Il est évident que plus on a de stocks, plus la chaîne logistique est réactive aux fluctuations des demandes sur le marché.

Cependant, avoir des stocks engendre des coûts et des risques surtout dans le cas de produits périssables ou bien des produits dont la rapidité d'innovations est telle qu'une nouvelle gamme du même produit mise sur le marché par un concurrent puisse rendre obsolètes les quantités de ce produit en stock et ainsi une perte importante. La gestion des stocks est l'une des clés de la réussite et l'optimisation de toute chaîne logistique.

7.4 Distribution et transports

La fonction transport intervient tout au long de la chaîne, le transport des matières premières, le transport des composants entre les usines, le transport des composants vers les centres d'entreposage ou vers les centres de distribution, ainsi que la livraison des produits finis aux clients.

Les problèmes liés à la distribution et au transport peuvent être vus sous plusieurs angles. On peut chercher à trouver les meilleures routes possibles pour visiter les points de collecte et/ou de distribution (vehicule routing problems, problèmes de tournées de véhicules), ou bien, comme dit plus haut, chercher les meilleurs modes de transports, ou bien les quantités des produits qui doivent être transportés aux clients tout en minimisant le coût global des transports et des stocks

En effet, selon les études les coûts de transport et distributions constituent le tiers des coûts opérationnels globaux d'une chaîne logistique, ce qui rend leur optimisation un défi majeur pour les entreprises.

7.5 La vente

La fonction de vente est la fonction ultime dans une chaîne logistique, son efficacité est interdépendante des performances des fonctions situées en amont. Si elle a bien été optimisée pendant les étapes précédentes, alors on facilite la tâche du personnel chargé de la vente, car il pourra offrir des prix plus compétitifs que la concurrence. Dans le cas contraire, les marges demeureront très étroites et les bénéfices peu importants, voire négatifs. [8]

8 Conclusion

Ce chapitre présente le cadre général de la gestion des chaînes logistiques. Après avoir défini la notion de chaîne logistique au moyen d'une description de ces entités, de ces flux et des activités. Ainsi, on a abordé la gestion de la chaîne logistique et ces différents rôles. Nous focalisons dans les prochains chapitres sur le problème de la distribution et le transport. Ce problème est connu sous le nom de tournée de véhicule, un problème d'optimisation combinatoire. En conséquence, le chapitre suivant détaillera cette notion, en concentrant sur les différentes méthodes existantes pour résoudre ce type de problèmes.

CHAPITRE 02
L'OPTIMISATION COMBINATOIRE

1 Introduction

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. Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes 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.

Étant donnée l'importance de ces problèmes, nombreuses méthodes de résolution ont été développées en recherche opérationnelle (RO) et en intelligence artificielle (IA). Ces méthodes 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, donc dans ce chapitre on parle plus détaillé sur les techniques et les méthodes d'optimisation utilisées pour résoudre notre problématique.

2 Définition d'un problème d'optimisation combinatoire :

Un problème d'optimisation combinatoire (OC) consiste à déterminer un plus grand (petit) élément dans un ensemble fini value. En d'autres termes, étant donné une famille F de sous-ensembles d'un ensemble $E = \{e_1, \dots, e_n\}$ et un système de poids $w = (w(e_1), \dots, w(e_n))$ associé aux éléments de E , un problème d'optimisation consiste à trouver un ensemble $F \in \mathcal{F}$ de poids $w(F) = \sum_{e \in F} w(e) \in \text{maximum (ou minimum)}$, i.e. \max ou $\min \{w(F) \mid F \in \mathcal{F}\}$.

La famille F représente donc les solutions du problème. Elle peut correspondre à un ensemble de très grande taille que l'on ne connaît que par des descriptions ou des propriétés théoriques qui ne permettent pas facilement son énumération. [27]

3 La complexité

Pour un problème donné, chercher un algorithme efficace, veut dire trouver un algorithme où le temps nécessaire à son exécution ne soit pas trop important. Un problème est dit facile si on peut le résoudre facilement, c'est-à-dire s'il ne fait pas trop de temps pour arriver à la solution. Donc, s'il existe un algorithme efficace pour un problème donné, alors ce dernier est dit facile

Un problème pour lequel on ne connaît pas d'algorithme efficace, est ce qu'il est facile ou difficile ?

De nombreux chercheurs se sont penchés sur ce genre de problèmes et ils ont développé une théorie dite de la complexité.

Nous n'allons pas détailler cette théorie, mais nous allons, quand même donner une idée globale du sujet en question.

3.1 Les complexités d'un algorithme

L'objectif de la théorie de complexité est d'analyser les coûts de résolutions surtout en termes de temps de calcul. Elle vise aussi à classifier les problèmes en plusieurs niveaux de difficulté. Une étude a prouvé que les problèmes de tournées de véhicules sont des problèmes difficiles. En général, la complexité algorithmique se mesure par rapport à deux paramètres :

- 1) Le temps alloué pour l'exécution de l'algorithme : il est relatif au nombre d'instructions à exécuter ainsi qu'à la taille des données manipulées.
- 2) Espace mémoire requis : associé à la taille d'instance d'un problème donné. [13]

3.2 La complexité problématique

La complexité problématique est relative au problème à résoudre ainsi que la méthode de résolution adoptée pour élaborer la solution optimale par rapport au critère retenu.

- Un problème de décision comprend deux parties : une partie donnée du problème et un processus binaire ayant « oui » ou « non » comme réponse possible.

- Un problème de recherche est un problème constitué d'un ensemble de données dont chacun représente un ensemble de solutions. Donc, la résolution d'un problème de recherche consiste à trouver pour chaque ensemble de données D des solutions S associées.

- Un problème d'optimisation est un problème de recherche en associant à chaque solution une valeur qualitative. A chaque problème d'optimisation, on peut associer un problème de décision (par exemple l'exclusion ou l'inclusion d'une solution dans les futures générations pour les AG), donc l'étude de la complexité du problème de décision peut donner des indications au problème d'optimisation associé. La théorie de la complexité permet de classer les problèmes en deux classes P et NP.

- La classe P regroupe les problèmes qui peuvent être résolus par des algorithmes polynomiaux. Un algorithme est dit polynomial, lorsque son temps d'exécution est borné par $O(P(x))$ où p est un polynôme et x est la longueur d'entrée d'une instance du problème.

- Les algorithmes dont la complexité ne peut pas être bornée polynomialement sont qualifiés d'exponentiels et correspondent à la classe NP.

- Un problème de décision est dit NP-complet s'il appartient à la classe NP et il résolu, au mieux, en un temps exponentiel.

- Un problème d'optimisation est dit NP-Difficile, si le problème de décision associé est NP-complet [26].

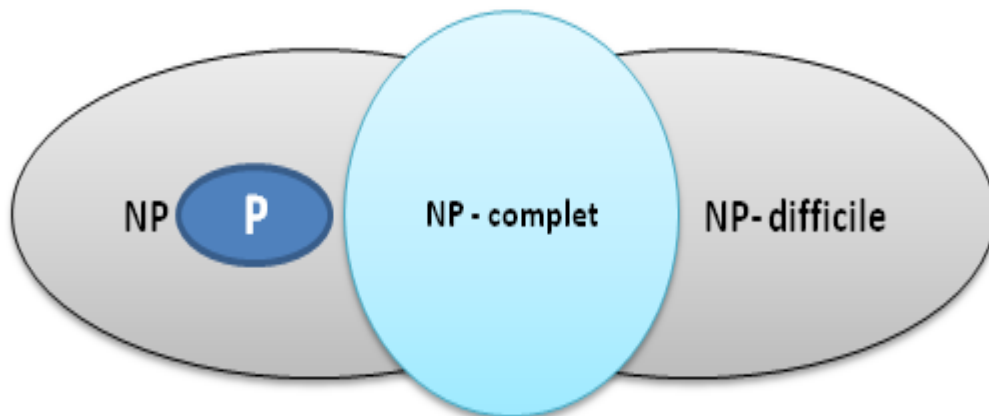


Figure 2.1 La relation entre la complexité des problèmes [26].

4 Quelques problèmes classiques d'optimisation combinatoire

Un problème d'optimisation consiste à chercher une instanciation d'un ensemble de variables soumises à des contraintes, de façon à maximiser ou minimiser un critère. Lorsque les domaines de valeurs des variables sont discrets, on parle alors de problèmes d'optimisation combinatoire.

Nous présentons rapidement ici quatre problèmes classiques d'optimisation combinatoire :

Le problème du sac-à-dos, le problème d'affectation, le problème du voyageur de commerce et le problème d'ordonnancement.

4.1 Problème du sac-à-dos

Le problème du Sac à Dos aussi noté KSP (Knapsack Problem en anglais) étant donné plusieurs objets possédants chacun un poids et une valeur et étant donné un poids maximum pour le sac, quels objets faut-il mettre dans le sac de manière à maximiser la valeur totale sans dépasser le poids maximal autorisé pour le sac. [10]

4.2 Problème d'affectation

Soit $T = \{t_1, \dots, t_n\}$ un ensemble de tâches et $O = \{o_1, \dots, o_n\}$ un ensemble d'ouvriers. On souhaite affecter chaque ouvrier à une seule tâche. Aussi, une tâche ne peut être réalisée que par un seul ouvrier à la fois. On suppose que l'affectation de l'ouvrier o_i à la tâche t_j génère un profit c_{ij} et qu'on lui attribue un budget de réalisation a_{ij} . On dispose d'un budget total b à ne pas dépasser.

Le problème d'affectation généralisé consiste donc à trouver une affectation des ouvriers aux tâches de sorte que le profit généré soit maximum et que le budget soit respecté.[11]

4.3 Problème d'ordonnancement

Le problème d'ordonnancement consiste à organiser dans le temps la réalisation d'un ensemble de tâches, compte tenu de contraintes temporelles (délais, contraintes d'enchaînements, ...) et de contraintes portant sur l'utilisation et la disponibilité des ressources requises. [12]

- Un ensemble de tâches.
- Un environnement de ressources pour effectuer les tâches.
- Des contraintes sur les tâches et les ressources.
- Un (ou plusieurs) critère(s) d'optimisation.
- Déterminer les dates d'exécution des tâches.

4.4 Problème du voyageur de commerce :

Ce problème du voyageur de commerce est un problème de recherche opérationnelle et n'est pas sans rappeler les cycles hamiltoniens. En effet, le nom donné à ce problème du voyageur de commerce a pour origine le fait qu'un circuit hamiltonien est celui que suivrait un voyageur de commerce partant d'une ville pour traverser une seule et unique fois n autres villes et revenir à son point de départ et ce à cause de son travail. Son objectif est rationnellement d'effectuer ces trajets, ce circuit en un temps minimal, une distance minimale ou un coût minimal. Pour parvenir à ses fins, le voyageur commercial peut bien sûr effectuer des essais successifs différents en notant à chaque essai, le temps nécessaire, le coût et la distance.[23]

5 Les méthodes d'optimisation combinatoire :

Les méthodes d'optimisation peuvent être réparties en deux grandes classes de méthodes pour la résolution des problèmes :

- Les méthodes exactes.
- Les méthodes approchées.

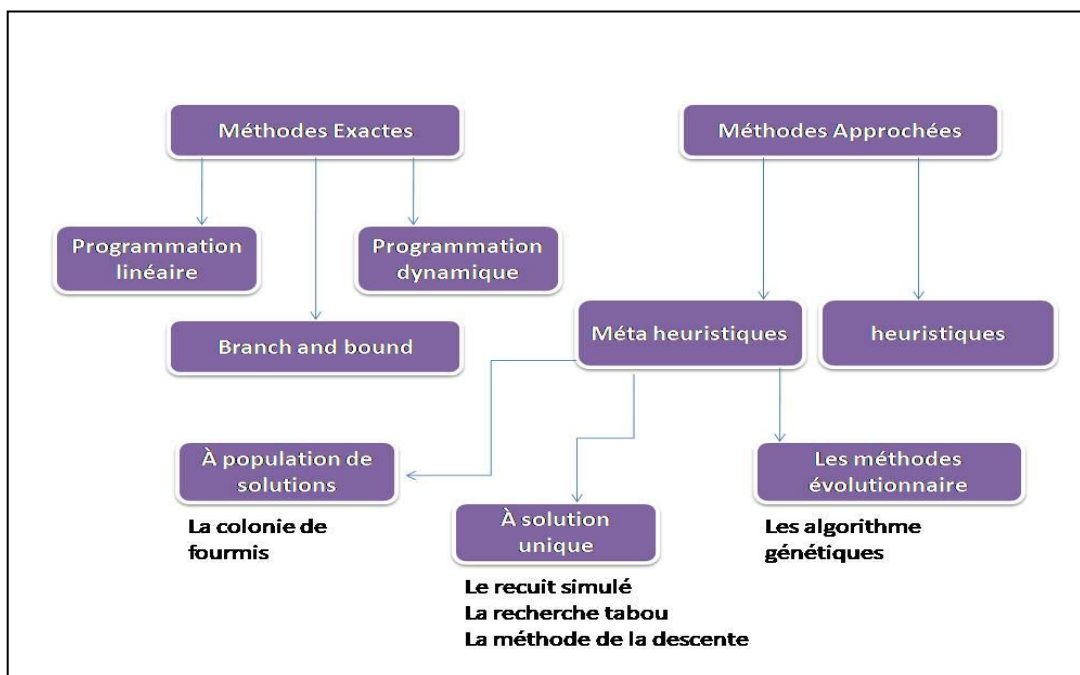


Figure 2.2 Les méthodes de résolution des problèmes d'optimisation combinatoire.[15]

5.1 Les méthodes exactes

Le principe des méthodes exactes consiste à rechercher, souvent de manière implicite, une solution, la meilleure solution ou l'ensemble des solutions d'un problème.

L'optimisation exacte concerne toutes les méthodes permettant d'obtenir un résultat dont on sait qu'il est optimal à un problème précis. Cela va des méthodes du simplexe aux méthodes de Lagrangien en passant par la programmation dynamique. On peut classer les méthodes exactes en quatre grandes classes [27]:

- La programmation dynamique.
- La programmation linéaire.
- Les méthodes de recherche arborescente (Branch & Bound).

1) La programmation dynamique

Est une méthode utilisée pour résoudre des problèmes où une séquence de décisions optimale doit être trouvée. L'idée de base est que l'on peut déduire une ou la solution optimale d'un problème en combinant des solutions optimales d'une série de sous-problèmes consistant à choisir des séquences plus courtes de décisions. Les solutions des problèmes sont calculées de manière ascendante, c'est-à-dire qu'on débute par les solutions des sous-problèmes les plus petits pour ensuite déduire progressivement les solutions de l'ensemble [14].

2) La programmation linéaire

La programmation linéaire (PL) est une branche de l'optimisation permettant de résoudre de nombreux problèmes économiques et industriels. La programmation linéaire désigne la manière de résoudre les problèmes dont la fonction objective et les contraintes sont toutes linéaires [20].

Si l'ensemble de solutions possibles S est formulé comme un ensemble de variables à valeurs dans l'ensemble des réels R et si on a des contraintes à satisfaire des inégalités linéaires et si f est une fonction linéaire en ces variables, on parle alors d'un problème de programmation linéaire (PL). Plusieurs problèmes réels de recherche opérationnelle peuvent être exprimés comme un problème de PL. Pour cette raison, un grand nombre d'algorithmes pour la résolution d'autres problèmes d'optimisation sont fondés sur la résolution de problèmes linéaires [28].

3) La méthode de branch and bound

La méthode de branch and bound (procédure par évaluation et séparation progressive) consiste à énumérer ces solutions d'une manière intelligente en ce sens que, en utilisant certaines propriétés du problème en question, cette technique arrive à éliminer des solutions partielles qui ne mènent pas à la solution que l'on recherche. De ce fait, on arrive souvent à obtenir la solution recherchée en des temps raisonnables. Bien entendu, dans le pire de cas, on retombe toujours sur 'élimination explicite de toutes les solutions du problème. Pour ce faire, cette méthode se dote d'une fonction qui permet de mettre une borne sur certaines solutions pour, soit les exclure soit les maintenir comme des solutions potentielles. Bien entendu, La performance d'une méthode de branch and bound dépend, entre autres, de la qualité de cette fonction [28].

5.2 Les 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 telles méthodes permettent d'obtenir des bornes inférieures ou supérieures de la valeur optimale telles que [21]:

- Méthodes Heuristiques
- Méthodes Méta-heuristiques

5.2.1 Les méthodes Heuristiques

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 [9].

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, 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 [24].

5.2.2 Les méthodes méta-heuristiques

Un algorithme de résolution méta-heuristique est un algorithme heuristique "générique" qu'il faut ajuster à chaque problème.

Une méta-heuristique est une heuristique généraliste, ils peuvent être appliqués à de nombreux problèmes d'optimisation.

Classification habituelle des méta-heuristiques : en fonction du nombre de solution qu'ils traitent

- ✓ Méta-heuristiques à solution unique.
- ✓ Méta-heuristiques à population de solutions.
- ✓ Les algorithmes évolutionnaires.

5.2.2.1 Méta-heuristiques à solution unique

Les méthodes itératives à solution unique sont toutes basées sur un algorithme de recherche de voisinage qui commence avec une solution initiale, puis l'améliore pas à pas en choisissant une nouvelle solution dans son voisinage [25].

Nous présenterons ici les méthodes les plus utilisées et leur utilisation en extraction de connaissances : les méthodes de descente, le recuit simulé et la recherche tabou.

1) Les méthodes de descentes

La méthode de descente décrite de manière générique dans l'algorithme 1 est un exemple de méthode de recherche locale. Une telle méthode progresse au travers de X en choisissant à chaque étape la meilleure solution voisine de la solution courante. Ce procédé est répété aussi longtemps que la valeur de la fonction objective diminue. La recherche s'interrompt dès lors qu'un minimum local de f est atteint.

Historiquement, les méthodes de descente ont toujours compté parmi les méthodes heuristiques les plus populaires pour traiter les problèmes d'optimisation combinatoire. Toutefois elles comportent deux obstacles majeurs qui limitent considérablement leur efficacité: suivant la taille et la structure du voisinage (s) considéré, la recherche de la meilleure solution voisine est un problème qui peut être aussi difficile que le problème (P) initial; une méthode de descente est incapable de progresser au-delà du premier minimum local rencontré. Ou les problèmes d'optimisation combinatoire comportent typiquement de nombreux optima locaux pour lesquels la valeur de la fonction objective peut être fort éloignée de la valeur optimale [16].

```

Début
Initialisation ;
Choisir une solution admissible initiale  $s \in X$  ;
Poser  $s^* := s$  ;
Processus itératif
Tant que le critère d'arrêt n'est pas satisfait faire
Générer  $N(s)$ :
Déterminer  $s' \in N(s)$  telle que
 $f(s') = \min f(s'')$  ;
 $s'' \in N(s)$  ;
 $s := s'$  ;
Si  $f(s) < f(s^*)$  alors  $s^* := s$  ;
Sinon le critère d'arrêt est satisfait ;
Fin Si
Fin Tant que
Fin

```

Algorithme 2.1 L'algorithme de la méthode de descentes. [16]

2) Le recuit simulé

Est une méthode empirique (méta-heuristique) destinée à résoudre au mieux les problèmes d'optimisation. Cette méthode est une technique de recherche locale inspirée du processus utilisé en métallurgie. Ce processus alterne des cycles de refroidissement lents et de réchauffage ou de recuit qui tendent à minimiser l'énergie physique. Le recuit simulé compte sur l'algorithme de Metropolis-Hastings, qui permet la description du développement du système thermodynamique. Comme mesuré par le processus physique, la fonction à minimiser deviendra l'énergie E du système. On introduit également un paramètre fictif, la température T du système.

L'algorithme du recuit simulé commence à partir d'une solution donnée. Et la modifier à plusieurs reprises jusqu'au refroidissement du système. Les solutions trouvées peuvent améliorer le critère que l'on cherche à optimiser, on dit que nous avons réduit l'énergie du système, car il peut être décomposé. Si on accepte une solution qui améliore le critère, on tend ainsi à chercher l'optimum dans le voisinage de la solution de départ. Contrairement autres méthodes de recherche locale, le recuit simulé peut accepter des solutions dont la qualité est moins bonne en fonction de la dégradation de la solution considérée.[22]

```

Début
Commencer avec une solution initiale : Solution courante
Température : TO (* Température initiale*)
Tant_que la condition d'arrêt n'est pas remplie faire
Pour i de 1 à N faire
Calculer nouvelle solution ( perturber la solution )
 $\Delta = \text{cout}(\text{nouvelle solution}) - \text{cout}(\text{perturber solution})$ 
Si  $\Delta < 0$  alors
Conserver nouvelle solution
Sinon  $\text{Random}[0,1] < e^{[-\Delta/\text{Température}]}$  alors
Conserver nouvelle solution
Fin Si
Fin Pour
Température =  $a * \text{Température}$  (*nouveau palier de température*)
Fin Tant que
Fin

```

Algorithme 2.2 L'algorithme de recuit simulé. [5]

3) La méthode tabou

La méthode Tabou est une méthode de recherche proposée par Fred Glover dans les années 1980 et est devenue très conventionnelle dans l'optimisation globale. Il diffère des méthodes de recherche locales simples en recourant à un historique des solutions qui ont été visitées, afin de rendre la recherche "aveugle" un peu moins. Et pour éviter de tomber périodiquement au minimum local, certains les solutions sont bloquées, sont « tabou ».

Contrairement à recuit simulé qui génère aléatoirement une solution proche l'une de l'autre $s' \in (s)$ à chaque itération, Tabou examine un échantillonnage de solutions de (s) et retient la meilleure s' même si $(s') > (s)$. La recherche Tabou ne s'arrête donc pas au premier optimum trouvé.

Alors le risque est de revenir à s immédiatement. Puisque s est meilleure que s' . Pour éviter de tourner ainsi en rond, on crée une liste T qui mémorise les dernières solutions visitées et qui interdit tout déplacement vers une solution de cette liste. Cette liste T est appelée liste Tabou. [28]

```

Début
Initialisation
 $s_0 = \text{une solution initiale } S, s^* = s_0, c^* = f(s_0);$ 
 $T = \emptyset;$ 
Générer un sous-ensemble de solution en voisinage de  $s_0$ ;
 $s' \in N(s_0)$  tel que  $\forall x \in N(s_0), f(x) \geq f(s')$  et  $s' \notin T$ ;
Si  $f(s') < c^*$  alors
 $s^* = s', c^* = f(s');$ 
Fin Si
Mise à jour de T;
Retour à l'étape 2 Si la condition d'arrêt n'est pas satisfaite
Fin

```

Algorithme 2.3 L'algorithme de la recherche tabou. [5]

5.2.2.2 Les Méta-heuristiques à base de population de solutions

Les méta-heuristiques à base de population de solutions débutent la recherche avec une panoplie de solutions. Elles s'appliquent sur un ensemble de solutions afin d'en extraire la meilleure (l'optimum global) qui représentera la solution du problème traité. L'idée d'utiliser un ensemble de solutions au lieu d'une seule solution renforce la diversité de la recherche et augmente la possibilité d'émergence de solutions de bonne qualité. Une grande variété de méthodes basées sur une population de solutions a été proposée dans la littérature, commençant par les algorithmes évolutionnaires, passant par les algorithmes génétiques et arrivant aux algorithmes à base d'intelligence par essaims (l'algorithme d'optimisation par essaim de particules, l'algorithme de colonies de fourmis, l'algorithme de colonies d'abeilles ...).[8]

➤ La Colonie de fourmis

Ces algorithmes inspirés du comportement des fourmis proposés par Marco Dorigo et al dans les années 90.

Une colonie de fourmis ayant le choix entre deux chemins d'inégale longueur menant à une source de nourriture avait tendance à utiliser le chemin le plus court. Le modèle expliquant ce comportement est le suivant :

- une fourmi appelée « éclaireuse » parcourt plus ou moins au hasard l'environnement autour de la colonie

- si celle-ci découvre une source de nourriture, elle rentre plus ou moins directement au nid, en laissant sur son chemin une piste de phéromones
- ces phéromones sont attractives, les fourmis passant à proximité vont avoir tendance à suivre, de façon plus ou moins directe, cette piste
- en revenant au nid, ces mêmes fourmis vont renforcer la piste si deux pistes sont possibles, pour atteindre la même source de nourriture, celle étant la plus courte sera, dans le même temps parcouru par plus de fourmis que la piste longue, la piste courte sera donc plus renforcée et donc de plus en plus attractive la piste longue, finira par disparaître, les phéromones sont volatiles à terme l'ensemble de fourmis a donc déterminé et choisi la piste la plus courte.[31]

```
Début  
Initialiser une population de m fourmis ;  
Tant que la condition d'arrêt n'est pas satisfaite faire  
  Pour i=1 à m faire  
    Construire le trajet de la fourmi i ;  
    Déposer des phéromones sur le trajet de la fourmi i ;  
  Fin pour  
  Evaluer les m fourmis ;  
  Evaporer les pistes de phéromones ;  
Fin Tant que  
Retourner la ou les meilleures solutions ;  
Fin
```

Algorithme 2.4 L'algorithme de colonies de fourmis pour le TSP. [18]

5.2.2.3 Les algorithmes évolutionnaires

➤ Les algorithmes génétiques

Un AG est une méthode d'optimisation dans laquelle un groupe appelé population de solution potentielles, appelées individus, est progressivement mis à jour à travers le mécanisme de sélection et le processus génétiques : Le croisement et La mutation.

La sélection naturelle que Darwin appelle l'élément "propulseur" de l'évolution, favorise les individus d'une population qui sont le mieux adaptés à un environnement. La sélection est

suivie de croisements et de mutations au niveau des individus, constitués d'un ensemble de gènes. Ainsi, deux individus "parents" qui se croisent, transmettent une partie de leur patrimoine génétique à leurs descendants. L'individu enfant fait que celui-ci est plus ou moins adapté à l'environnement. S'il est bien adapté, il a une plus grande chance de procréer dans la génération future. Au fur et à mesure des générations, sont sélectionnés les individus les mieux adaptés, et l'augmentation du nombre des individus bien adaptés fait évoluer la population entière [19].

Principe de l'algorithme

Les algorithmes génétiques sont la combinaison de deux domaines ; la biologie et l'informatique. Pour définir le fonctionnement d'un algorithme génétique, on commence par définir les mots techniques utilisés :

- ♦ Genèse: c'est la première phase de l'algorithme, il s'agit d'une population initiale de taille N.
- ♦ Chromosome : c'est une chaîne représentant les caractéristiques de l'individu
- ♦ Phénotype : c'est un ensemble de paramètres ou une structure décodée.
- ♦ Evaluation : c'est la phase de calcul de la fonction de fitness.
- ♦ Sélection : c'est le choix des individus qui vont se reproduire.
- ♦ Croisement : c'est la phase de production des descendants.
- ♦ Mutation : c'est la modification d'un chromosome dans le but d'améliorer les caractéristiques de l'individu. [17]

La mise en œuvre des algorithmes génétiques nécessite plusieurs étapes à détailler. La première est le codage d'un individu représenté par un chromosome. La seconde, est le calcul de la qualité. La troisième est de définir les opérateurs de reproduction.

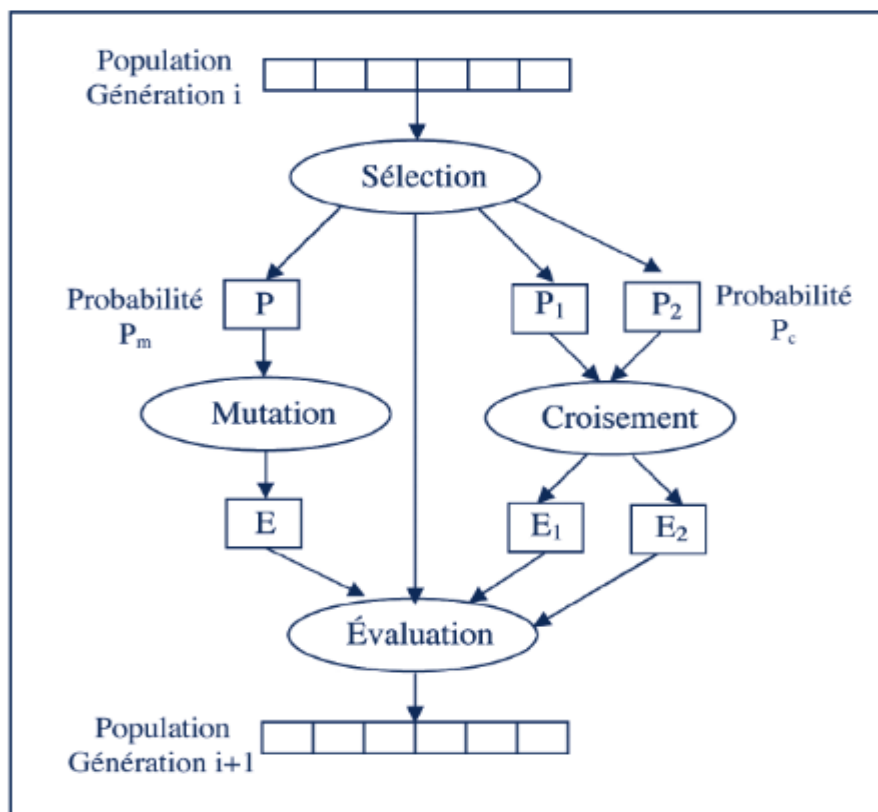


Figure 2.3 Principe général d'un algorithme génétique.

a) Codage des individus :

Cette étape associe à chacun des points de l'espace d'état une structure de données. Elle se place généralement après une phase de modélisation mathématique du problème traité.

La qualité du codage des données conditionne le succès des algorithmes génétiques. Les codages binaires ont été très utilisés à l'origine. Les codages réels sont désormais largement utilisés, notamment dans les domaines applicatifs pour l'optimisation de problèmes à variables réelles [19]. Des exemples du codage sont présentés dans figure suivante :

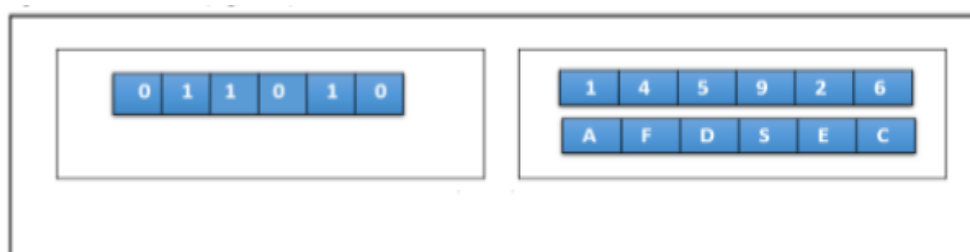


Figure 2.4 Codage des solutions. [19]

Les algorithmes génétiques utilisent trois opérateurs pour générer de nouvelles solutions :

- L'opérateur de sélection qui permet de choisir des solutions parentes sur lesquelles la reproduction va être faite pour générer de nouvelles solutions.

- L'opérateur de croisement qui permet de croiser les deux solutions parentes et créer de nouvelles solutions.
- L'opérateur de mutation qui permet de diversifier les nouvelles solutions afin qu'elles ne ressemblent pas trop aux solutions parentes.

b) L'opérateur de sélection :

La sélection consiste à choisir des individus qui permettront de générer de nouveaux individus. Plusieurs méthodes existent pour sélectionner des individus destinés à la reproduction. On citera les deux méthodes classiques les plus utilisées.

1. La sélection par roulette : la population est représentée comme une roue de roulette, où chaque individu est représentée par une portion qui correspond proportionnellement à sa valeur de fitness. La sélection d'un individu se fait en tournant la roue en face d'un pointeur fixe. Cette procédure est répétée jusqu'à constitution complète de la population des parents. L'un des inconvénients de ce type de sélection est de choisir presque toujours le même individu s'il en existe un bien meilleur que les autres, ce qui cause une perte de diversité dans la population [20]. La figure 2.5 illustre une population de 5 individus dont les performances sont représentées en roulette.

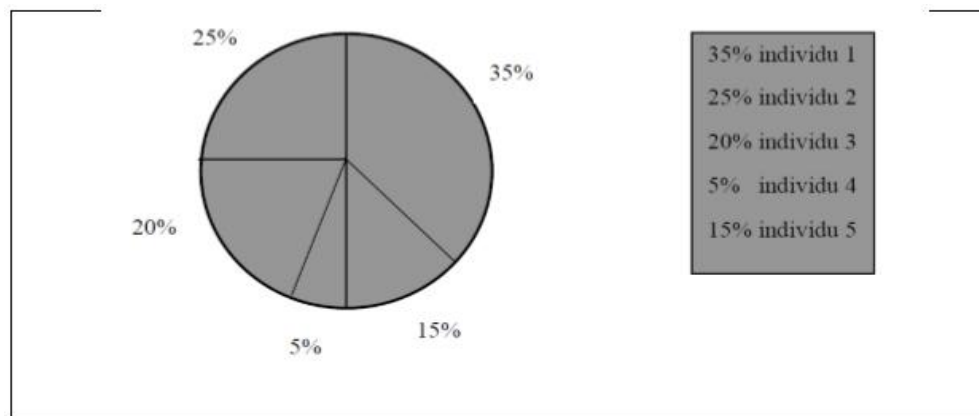


Figure 2.5 Sélection par roulette.

2. La sélection par tournoi : La sélection par tournoi est l'une des sélections les plus utilisées dans les algorithmes évolutionnaires. Le principe consiste à choisir aléatoirement un sous-ensemble d'individus (K individus) dans la population, puis à sélectionner le meilleur individu dans ce groupe en fonction de sa fitness. Ce processus est répété jusqu'à l'obtention du nombre d'individus requis.

Le nombre de participants à un tournoi (K), appelé la taille du tournoi, est utilisé pour faire varier la pression de cette sélection. Si ce nombre est grand, alors la pression sera forte et les faibles individus auront une petite chance d'être choisis. En général, un seul gagnant est choisi parmi les participants à un tournoi. Ce gagnant peut être choisi d'une façon déterministe ou probabiliste. Dans le cas déterministe, qui est pratiquement le plus utilisé, le gagnant est l'individu de meilleure qualité (meilleure fitness). Dans le cas probabiliste, chacun des participants peut être choisi en tant que gagnant avec une probabilité proportionnelle à sa fitness [20]. Cette méthode est en général satisfaisante.

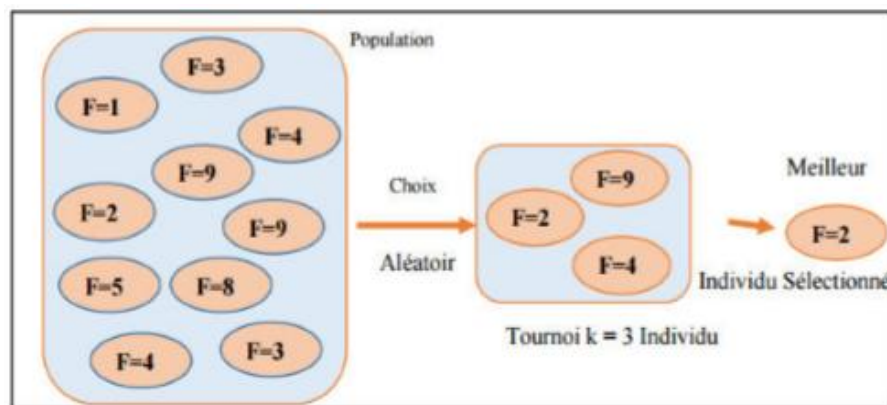


Figure 2.6 Sélection par tournoi. [20]

c) L'opérateur de croisement :

Les croisements permettent de simuler des reproductions d'individus dans le but d'en créer de nouveaux. Il est tout à fait possible de faire des croisements aléatoires. Toutefois, une solution largement utilisée est d'effectuer des croisements multi-points [20].

- Le croisement à un point : Il a été initialement défini pour le codage binaire. Le principe consiste à tirer aléatoirement une position pour chaque parent et à échanger les sous-chaines des parents à partir des positions tirées ; ce qui donne naissance à deux nouveaux individus ind1 et ind2 [20].

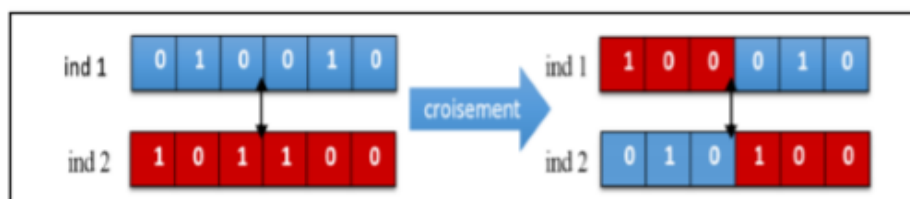


Figure 2.7 Croisement à un point. [20]

- Le croisement multi-points : Cette méthode reprend le mécanisme de la méthode de croisement à un point en généralisant l'échange à 3 ou 4 sous-chaines.

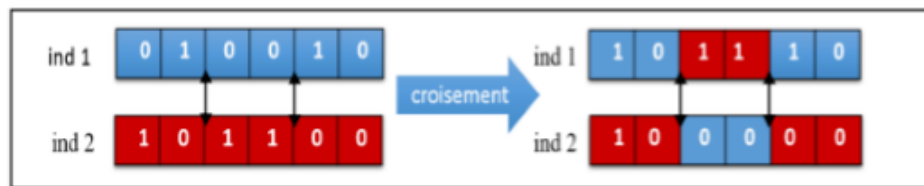


Figure 2.8 Une mutation. [20]

d) Evaluation de la fonction fitness :

On appelle adaptation d'une séquence A une valeur positive que nous noterons $f(A)$. f est la fonction objectif ou fitness du problème à résoudre.

Un critère permettant de déterminer l'adaptation d'un individu par rapport à l'environnement afin de classer les individus entre eux. Dans le vocabulaire des algorithmes génétiques, on dénomme par "fitness" (traduction anglaise du mot adaptation) le critère à optimiser

L'évaluation de la Fitness est généralement l'étape dans laquelle on mesure la performance de chaque individu. Pour pouvoir juger la qualité d'un individu et ainsi le comparer aux autres, il faut établir une mesure commune d'évaluation. Aucune règle n'existe pour définir cette fonction, son calcul peut ainsi être quelconque, que ce soit une simple équation ou une fonction affine. La manière la plus simple est de poser la fonction d'adaptation comme la formalisation du critère d'optimisation.

e) Opérateur de remplacement :

Cet opérateur est le plus simple, son travail consiste à réintroduire les descendants obtenus par application successive des opérateurs de sélection, de croisement et de mutation (la population P') dans la population de leurs parents (la population P). Ce faisant il vont remplacer une certaine proportion de ceux-ci, proportion pouvant bien sûr être choisie. Le rapport entre le nombre d'individus nouveaux allant être introduits dans la population P et le nombre d'individus de cette population est connu sous le nom de génération gap.

On trouve essentiellement deux méthodes de remplacement différentes :

- Le remplacement stationnaire : consiste à remplacer automatiquement les parents par les enfants. Ceci induit une grande diversité de population.
- Le remplacement élitiste : on ne garde que les individus performants en regardant sur la génération complète et non que sur les parents.[21]

6 Conclusion

La programmation mathématique recouvre un ensemble de techniques d'optimisation sous contraintes qui permettent de déterminer dans quelles conditions on peut rendre maximum ou minimum une fonction objective, de nombreux problèmes de l'entreprise peuvent s'exprimer en termes d'optimisation contrainte, aussi rencontre-t-on de multiples applications de la programmation mathématique et ceci dans pratiquement tous les domaines de la gestion.

Dans ce chapitre, nous avons essayé d'aborder quelques notions de base en optimisation. Nous avons présenté des définitions de notions confrontées souvent dans le domaine de l'optimisation en allant de la définition d'un problème d'optimisation, passant par les types des problèmes d'optimisation jusqu'à la définition d'une solution optimale d'un problème donné. Ensuite, nous avons dévoilé les processus d'optimisation et Les méthodes de résolution de problèmes d'optimisation (Exacte et approchée) et nous avons spécifié les méthodes heuristiques et méta-heuristiques.

Dans le chapitre qui suit, nous présentons le problème d'optimisation de tournée de véhicule avec fenêtre de temps (PTVFT).

CHAPITRE 03

PROBLEME DE TOURNEES DE VEHICULES AVEC FENETRE DE TEMPS

1 Introduction

Le problème de tournées avec fenêtres de temps est une variante du problème classique de routage de véhicule, il désigne le cas où les visites aux clients (ou aux fournisseurs) ne sont autorisées que dans des intervalles horaires précis. Ce type de situations est fréquent par exemple dans le cas de la distribution aux hypermarchés, lesquels traitent en général avec plusieurs milliers de fournisseurs différents, et sont obligés d'imposer l'heure de livraison des commandes, pour pouvoir préparer correctement la réception des marchandises. Nous nous intéressons particulièrement à ce problème, à cause de son haut degré de réalisme, et parce qu'il représente une situation souvent rencontrée en milieu industriel [29].

Plus précisément, dans la suite de ce chapitre, Nous commencerons par définir les problèmes de tournée des véhicules PTV et PTV avec fenêtre de temps et exposer la formulation mathématique de base du PTVFT puis nous traiterons les variantes de notre problème, enfin, nous présenterons une conclusion ou nous essaierons de dégager quelques éléments de réflexions concernant les futurs axes de recherche dans ce domaine.

2 Le problème de tournée des véhicules

Le problème de tournées de véhicules (vehicles routing problem VRP) est une extension du problème du voyageur du commerce. Il a été introduit pour la première fois par Dantzig en 1954 sous le nom de (Truck Dispatching Problem) et a depuis fait l'objet d'études intensives pour le modéliser et le résoudre. Dans sa version la plus basique dite (CVRP) ou VRP avec contraintes de capacité, une flotte de véhicules de capacité finie, basée dans un dépôt, doit assurer des tournées entre plusieurs clients (ou villes) ayant demandé chacun une certaine quantité de marchandises. L'ensemble des clients visités par un véhicule désigne la tournée de celui-ci. Chaque client doit être desservi une et une seule fois et chaque tournée commence et se termine au dépôt. L'objectif du CVRP est de minimiser le coût total, la somme des distances ou des temps de parcours des tournées, tout en respectant la contrainte de capacité des véhicules : la quantité de marchandises livrées sur une tournée ne doit pas dépasser la capacité du véhicule qui l'assure.[37]

La figure 3.1 représente un exemple de problème de VRP avec 25 clients, résolu avec 5 véhicules.

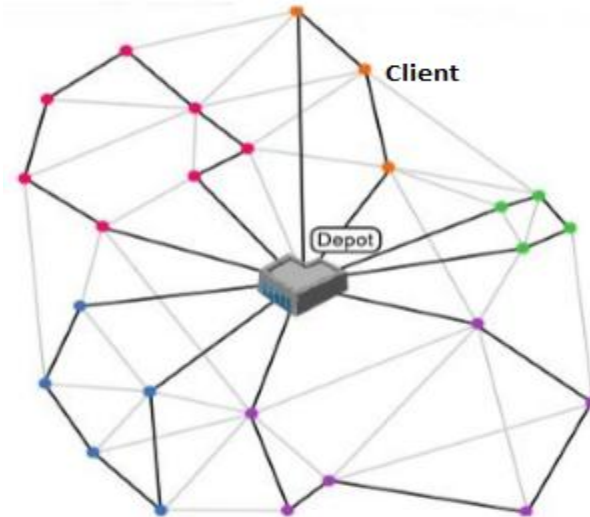


Figure 3.1 Présentation graphique du problème de tournées de véhicules VRP. [37]

2.1 Formulation du VRP

Formulation La version standard du VRP considère une contrainte de capacité (PTVC, Problème De Tournée Des Véhicules Avec Capacité). Il peut être représenté sous la forme d'un graphe orienté et valeur $G = (N, A)$

Où :

- N représente les positions des clients et du dépôt,
- A représente les arcs entre deux clients $i, j \in N$.

Plus spécifiquement, nous avons un ensemble $C = \{1, \dots, n_c\}$ de clients qui doivent obtenir une livraison de marchandise provenant du dépôt. L'ensemble des positions de ces clients ou nœuds est défini par l'ensemble $N = C \cup \{0, n_{c+1}\}$ où 0 et n_{c+1} représentent le dépôt (aller et retour). Une demande positive de produit d_i est associée à chaque client i appartenant à C .

Une flotte de véhicules $V = \{1, \dots, n_v\}$ est disponible au dépôt et chaque véhicule possède la même capacité (flotte homogène) Q telle que $Q \geq \max d_i, \forall i \in N$. Pour tous les clients i et $j, \forall i, j \in N$, nous connaissons le coût c_{ij} de transport direct entre i et j (proportionnel à la distance à parcourir). Pour trouver l'ordre de visite des clients, nous définissons les variables de décisions comme suit :

$$x_{ij}^v = \begin{cases} 1 & \text{Si le véhicule } v \in V \text{ visite le client } j \text{ après le client } i, \\ 0 & \text{sinon.} \end{cases}$$

En définissant y_i comme étant la charge résiduelle du véhicule après avoir desservi le client $i \in C$. Il nous est possible d'écrire formellement le modèle de VRP. Il s'agit d'optimiser la fonction suivante :

$$\text{Min} \sum_{v \in V} \sum_{i \in N} \sum_{j \in N} c_{i,j} x_{i,j}^v \quad (1)$$

Avec les contraintes :

$$\sum_{v \in V} \sum_{j \in N} x_{i,j}^v = 1, \forall i \in C \quad (2)$$

$$\sum_{i \in N} x_{i,h}^v - \sum_{j \in N} x_{h,j}^v = 0, \forall h \in C, v \in V \quad (3)$$

$$\sum_{j \in N} x_{0,j}^v = 1, \forall v \in V \quad (4)$$

$$\sum_{j \in N} x_{j,n+1}^v = 1, \forall v \in V \quad (5)$$

$$x_{i,j}^v = 1 \Rightarrow y_i - d_j = y_j, \forall i, j \in N, v \in V \quad (6)$$

$$y_0 = Q, 0 \leq y_i, \forall i \in C \quad (7)$$

$$x_{i,j}^v \in \{0, 1\}, \forall i, j \in N, v \in V \quad (8)$$

La fonction de coût de la solution $X = (x_{i,j}^v), \forall i, j \in N, \forall v \in V$ est définie par :

$$\text{coût}(X) = \sum_{v \in V} \sum_{i \in N} \sum_{j \in N} c_{i,j} x_{i,j}^v \quad (9)$$

Le nombre de véhicules utilisés par la solution X , est défini par :

$$Nb\ véhicules(X) = \sum_{v \in V} \sum_{j \in C} x_{0,j}^v \quad (10)$$

La fonction objective (équation 1) représente le nombre de véhicules utilisés pour les trajets effectués et la somme des coûts s'y rapportant.

La formulation du problème nécessite de satisfaire certaines contraintes :

- ✓ L'équation 2 assure qu'on part une et une seule fois de chaque client, avec un seul véhicule.
- ✓ L'équation 3 assure que le véhicule qui arrive chez un client est le même que celui qui part de ce client.
- ✓ L'équation 4 assure que chaque véhicule ne sort qu'une seule fois du dépôt.
- ✓ L'équation 5 assure le retour unique au dépôt pour chaque véhicule (ou tournée). Il n'ya pas de restriction sur le nombre de véhicules, mais un coût c_v est affecté à chaque véhicule utilisé. On impose une valeur de c_v suffisamment grande pour réduire principalement le nombre de véhicules et pour réduire au minimum dans un deuxième temps les coûts de transport.
- ✓ Les équations 6 et 8 définissent les contraintes de capacité et d'intégrité.
- ✓ Les équations 9 et 10 sont des fonctions de mesure qui permettent respectivement de quantifier la solution selon la distance totale parcourue, ainsi que le nombre de véhicules utilisés. [36]

Le problème d'élaboration de tournées de véhicules est un problème NP-difficile, c'est-à-dire qu'il n'existe pas à ce jour un algorithme déterministe pouvant résoudre ce problème en temps polynomial. Pour des problèmes comportant un grand nombre de clients (> 100 clients), des méthodes approchées sont nécessaires pour les résoudre.

3 Le problème de tournées de véhicules avec fenêtre de temps

Le PTVFT (Vehicle Routing Problem With Time Windows – VRPTW), constitue une généralisation du VRP dans la mesure où nous introduisons en plus une contrainte temporelle sur le service demandé. Chaque client dispose d'une fenêtre de temps à l'intérieur de laquelle il désire être servi. Le dépôt central possède également une fenêtre de temps que nous désignons couramment comme horizon de service ou temps d'ouverture de la journée. Son rôle est de fixer une plage horaire durant laquelle les véhicules peuvent effectuer leur tournée.

Ces contraintes temporelles vont rendre nécessaire l'utilisation de plusieurs véhicules pour satisfaire l'ensemble des clients sur l'horizon de service. On peut vouloir borner le nombre de véhicules à utiliser et dans ce cas des clients risquent de ne pas être servis [38].

Le problème de tournées de véhicules avec fenêtre de temps. Chaque client doit être servi dans un intervalle de temps défini, connu d'avance par le livreur et toute violation de cette contrainte peut engendrer une pénalité. Lorsque la contrainte de fenêtre de temps n'est pas satisfaite, soit on rejette la solution si on considère le cas rigide ou bien on construit une fonction de pénalité qui sera rajoutée ou combinée avec la fonction objective pour le cas relâché. En réalité, c'est un problème très fréquent. La distribution des produits périssables (le lait, la viande . . .), de journaux, services ambulatoires, . . . sont des exemples pratiques du PTVFT. Dans cette classe de problèmes, on distingue deux sous-classes :

- Le PTVFT rigide où le service doit impérativement être effectué dans la fenêtre de temps.
- Le PTVFT relâché où le retard ou l'avance engendre uniquement une pénalité. [38]

Une fenêtre de temps (TW - Time Window) impose qu'une ou plusieurs requêtes du problème soit traitée en respectant un intervalle pour le début de traitement du véhicule (livraison, collecte). Cet intervalle est défini par une date au plus tôt et une date au plus tard. Cette contrainte peut s'appliquer à des demandes de collecte ou de livraison comme présenté dans La figure 3.2 [30].

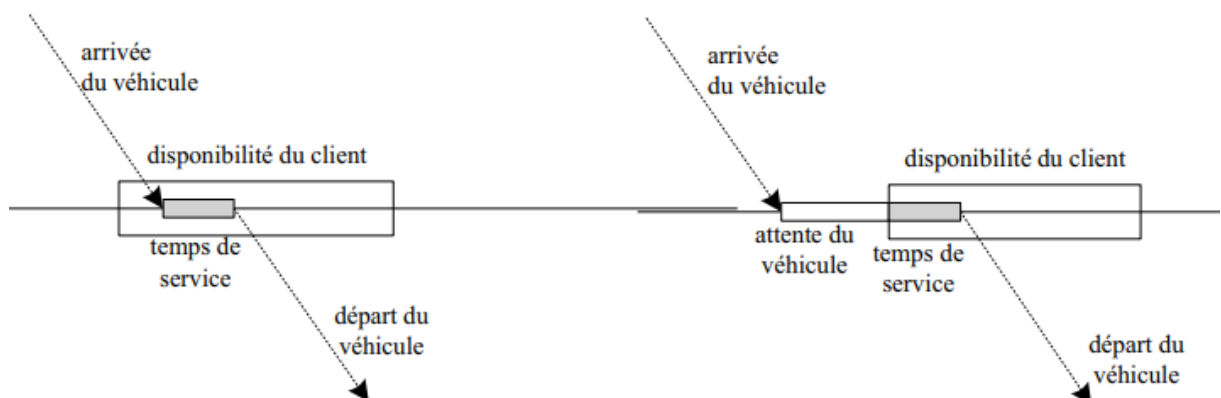


Figure 3.2 Exemple de fenêtre de temps sur un client. [30]

Ce type de contrainte implique le plus souvent d'autoriser le véhicule à arriver en avance, auquel cas, il attend sur le sommet afin de démarrer ses opérations au début de la fenêtre de temps, comme illustré sur la figure (3.2). Cette attente peut être autorisée sur tous les sommets du graphe ou, uniquement sur une partie des sommets. En revanche, le véhicule n'est pas autorisé à arriver après la fin de la fenêtre de temps. Dans la version classique de la contrainte,

si une des fenêtres de temps n'est pas respectée alors la solution n'est pas valide. Il existe des variantes dans la littérature, par exemple la fenêtre de temps souple (STW - Soft Time Window) introduite pour la première fois sur un problème de TSP par (Sexton and Choi, 1986), puis sur un problème de VRP par (Figliozzi, 2008), qui autorise la violation de la fenêtre de temps mais pénalise la fonction objective[30].

Le problème de tournées de véhicules (avec fenêtres de temps) est un des problèmes de routage. Lenstra et Rinnooy Kan ont prouvé que le VRP est un problème NP-DUR. Sa résolution par une méthode exacte s'avère inappropriée pour les instances de grande taille. Il est donc inévitable de procéder à sa résolution par des approches heuristiques, qui fournissent des solutions réalisables et appréciables en un temps raisonnable. [32]

3.1 Formulation du PTVFT

Nous formulons le problème PTVFT en utilisant la formulation de PTV (VRP) classique qui déjà exprimée et en y ajoutant les variables, les constantes et les équations suivantes:

Variables à déterminer :

1. $a_i =$ instant d'arrivée chez le client $i \in C$.
2. $b_i =$ instant de début de service chez le client $i \in C$.
3. $b_0^v =$ instant auquel le véhicule v quitte le dépôt.
4. $b_{n_c+1}^v =$ instant auquel le véhicule v retourne au dépôt.
5. $w_i =$ temps d'attente chez le client $i \in C$.

Constantes connue :

1. $e_i =$ borne inférieure de la fenetre de temps du client $i \in C$.
2. $l_i =$ borne séperieure de la fenetre de temps du client $i \in C$.
3. $c_{ij} =$ cout de déplacement de i à $j : i, j \in C$.
4. $t_{ij} =$ le temps de parcours entre les deux clients i et $j : i, j \in C$.
5. $s_i =$ temps de service chez le client $i \in C$.

L'attente est permise lorsqu'un véhicule arrive trop tôt chez le client j après que le service soit fini chez le client $i \in C$, autrement dit, avant e_j . Le temps de début de service chez le client $j \in C$ se définit comme étant $b_j = \max\{e_j, a_j\}$ ou $a_j = \{b_i + s_i + t_{ij}\}$ et le temps d'attente chez le client j comme étant $w_j = b_j - a_j$. Il est alors possible d'écrire les contraintes supplémentaires de la formulation de VRP pour formuler le problème PTVFT :

$$x_{i,j}^v = 1 \Rightarrow b_i + s_i + t_{i,j} < b_j, \forall i, j \in C, v \in V \quad (11)$$

$$x_{0,j}^v = 1 \Rightarrow b_0^v + t_{0,j} < b_j, \forall j \in C, v \in V \quad (12)$$

$$x_{i,n+1}^v = 1 \Rightarrow b_i + s_i + t_{i,n+1} < b_{n+1}^v, \forall i \in C, v \in V \quad (13)$$

$$e_i \leq b_i \leq l_i, \forall i \in C \quad (14)$$

$$e_0 \leq b_0^v \leq l_0, \forall v \in V \quad (15)$$

$$e_{n+1} \leq b_{n+1}^v \leq l_{n+1}, \forall v \in V \quad (16)$$

Le temps d'utilisation réel des véhicules dans la solution (X) est :

$$\text{temps horaire}(X) = \sum_{v \in V} (b_{n_c}^v - b_0^v) \quad (17)$$

Les contraintes (11 - 16) définissent les contraintes temporelles. La mesure (17) permet d'identifier le temps total d'utilisation des véhicules.

Dans un (PTV) VRP, la définition de distance entre deux clients qui nous vient naturellement à l'esprit est la distance que requiert le parcours direct entre ces deux clients. Cette distance est la même dans les deux sens, autrement dit, la distance entre le client i et j est la même entre j et i , donc la matrice des distances entre les clients est symétrique. [36]

4 Problème de tournées de véhicules et ses variantes

Durant des années de recherches sur le VRP d'autres dérivées de ce problème sont apparues. Ces apparitions sont dues essentiellement aux activités des chercheurs qui travaillent de plus en plus sur les problèmes de transport et de distribution que rencontrent les sociétés. Dans ce qui suit nous allons présenter les principaux problèmes dérivés du PTVFT :

4.1 PVC (Problème du voyageur de commerce)

Le problème du voyageur de commerce consiste à trouver le chemin le plus court pour passer par l'ensemble des villes données une fois et une seule et revenir à la ville de départ. C'est donc un cas particulier du VRP sans contrainte de capacité et avec un seul véhicule. [34]

4.2 PTVC (Problème de tournées de véhicules avec capacité)

Comme défini précédemment, un problème PTVC consiste à affecter chaque client à une tournée effectuée par un seul véhicule de capacité finie. Ce véhicule commence et termine sa tournée au dépôt. [33]

4.3 PTVD (Problème de tournées de véhicules dynamique)

Contrairement au VRP classique où toutes les demandes des clients sont connues et planifiées à l'avance, le PTVD illustre mieux la réalité. Dans les problèmes de transport réels dans la société d'aujourd'hui la composante dynamique est très importante, elle s'exprime surtout par les apparitions de nouveaux clients, de nouvelles demandes, ou de pannes des véhicules de transport. Dans ce genre de situation, le décideur doit réorganiser, en un temps acceptable, la planification des itinéraires de sa flotte de véhicules pour répondre aux nouvelles demandes urgentes. [35]

4.4 VRPB (Prolongation du VRP classique)

Ce problème est une prolongation du VRP classique. Il comporte des clients auxquels des produits doivent être livrés et des fournisseurs dont les marchandises doivent être amenées au dépôt. Donc, pour éviter de réarranger les produits transportés dans le véhicule sur une tournée, les clients doivent être visités avant les fournisseurs. [34]

4.5 PTVMP (Problème de tournées de véhicules multi périodique)

Le problème de tournées de véhicules multi périodique (PTVMP) consiste à livrer pour un ensemble de clients, la quantité demandée d'un ou de plusieurs produits sur un horizon de temps donné. Dans ce problème, la quantité de produits livrée à un client, permet à ce dernier de subvenir à ses besoins en attendant la prochaine visite du véhicule. Le but principal de ce problème se divise en deux parties : la première consiste à planifier les horaires de livraison de chaque client sur un horizon de temps prédéterminé ; la deuxième consiste à organiser les tournées des véhicules afin d'effectuer les livraisons nécessaires tout en optimisant le coût total de transport. Outre la connaissance et l'utilisation des contraintes connues d'un problème de transport, le PTVMP nécessite l'intégration des éléments de gestion de stock afin de respecter les besoins en quantité et en temps des clients. [31]

4.6 PTVS Problème de tournée des véhicules stochastique

Un problème VRP est dit stochastique si au moins un de ses éléments est aléatoire, c'est à-dire avec une certaine incertitude. Ces éléments peuvent être une ou plusieurs demandes des clients, les temps ou les coûts de transport. Le problème avec les demandes stochastiques est celui le plus étudié dans la littérature. La majorité des recherches suppose que les demandes aléatoires suivent une loi de distribution « normale ». [32]

5 Conclusion

Dans ce chapitre nous avons présenté le problème de la tournée de véhicule avec fenêtre de temps. Comme nous l'avant déjà ce problème appartient à la classe NP difficiles. Donc sa résolution n'est possible que d'une manière approchée car de nos jours il n'y a aucun algorithme qui puisse nous permettre leur résolution en un temps polynomial. Alors dans le prochain chapitre, nous introduisons notre approche adaptée pour résoudre ce problème.

CHAPITRE 04
IMPLEMENTATION ET RESULTATS
EXPERIMENTAUX

1 Introduction

Ce chapitre est consacré à la réalisation et l'implémentation de différentes fonctionnalités de notre programme. Nous expliquerons notre approche pour résoudre le problème de tournée de véhicules avec fenêtre de temps. Nous avons choisi d'utiliser les algorithmes génétiques pour résoudre ce problème. Des résultats expérimentaux sont présentés afin de mesurer l'efficacité de notre solution.

2 L'objectif du travail

L'objectif de ce travail est l'utilisation d'une heuristique pour résoudre le problème de la tournée de véhicule avec fenêtre de temps. Dont la fonction objective est la minimisation des coûts de transport, en termes de temps et de distance. Il consiste à visiter un nombre N de villes (clients) en un minimum de distance sans passer deux fois par la même ville, les visites aux leurs ne sont autorisées que dans des intervalles horaires précis. Pour l'heuristique, nous avons opté à choisir les algorithmes génétiques.

3 Description de l'approche proposée

Pour résoudre le problème, nous avons opté à choisir les algorithmes génétiques comme heuristique. La motivation principale est les avantages suivants :

- ✓ Les algorithmes génétiques sont parmi les premières méthodes utilisées
- ✓ Ils sont simples à implémenter
- ✓ Permet de traiter des espaces de recherche important (beaucoup de solution, pas de parcourt exhaustif envisager).
- ✓ Pour un nombre de solution important, les algorithmes génétiques permettent d'élimination les solutions non valides.
- ✓ Relativité de la qualité de la solution selon le degré de précision demandé.

3.1 Contraintes du problème de tournées de véhicules avec fenêtre de temps

Dans notre approche nous proposant les contraintes suivantes :

- ✓ Un centre de distribution logistique fournit des services de distribution pour 25 clients.
- ✓ Les fenêtres de temps sont différentes.
- ✓ Le centre de distribution dispose de 3 véhicules du même type et son autonomie est de 160 km.
- ✓ Le poids de la charge Q est 5.

- ✓ Le temps nécessaire pour que le véhicule quitte le centre de distribution est égal à 0
- ✓ La vitesse de conduite pendant le processus de livraison est constante à 40 Km/ h.
- ✓ Le coût de conduite d'un véhicule est de 50 DA pour le kilomètre.
- ✓ Le coût de la pénalité Epu est de 100 DA pour l'heure en retard
- ✓ Les clients sont représentés comme des sites géographiques
- ✓ Les sites géographiques sont dans une ville quelconque, pour notre algorithme ils sont présentés par des points avec les coordonnées X et Y.

$$\text{Client 1} = (X1, Y1)$$

$$\text{Client 2} = (X2, Y2)$$

- ✓ La distance entre deux clients calculés comme distance euclidienne ou géométrique dans cet espace bidimensionnel. Il est calculé comme :

$$\text{Distance}(X, Y) = \{\sum_i (X_i - Y_i)^2\}^{1/2}$$

3.2 Paramètres du problème

Les paramètres du problème concernent deux axes : les clients et le centre du dépôt

1) La partie qui concerne les attributs des clients :

- n : nombre des clients
- t : est la quantité de service demandée par le client.
- h : est la durée de service chez le client.
- (eh, Ih) : est la fenêtre de temps de client, dans laquelle il souhaite être servi.

2) La partie qui concerne le dépôt :

- m : le centre de distribution.
- (eh, lh) : sont les horaires d'ouverture et de fermeture.
- K = nombre de véhicules.
- c : la capacité du véhicule : elle est servie selon la demande du client.
- v : la vitesse du véhicule pendant la livraison dans la ville
- Cklm : Le coût de 1klm
- Lpu : La valeur de la pénalité du temps (cas où le véhicule arrive chez le client plus tard que la fenêtre horaire).

3.3 Paramètres d'algorithme génétique

L'espace de recherche : est l'ensemble des permutations de {1, 2, ..., n} et la population : est un ensemble des individus, Chaque individu représenté une solution.

La solution : est codée sous forme d'un vecteur, chaque case représente un client dans l'ordre de la tournée. Alors la taille de vecteur représenté la solution (L'ordre des Clients par une tournée des véhicules).

Client 1	Client 2	Client 24	Client 25
-----------------	-----------------	-------	------------------	------------------

Figure 4.1 Exemple d'une solution.

La fonction fitness : ou la fonction d'adaptation qui calcule le coût la distance par kilomètre et le coût de temps (la pénalité de retard) entre les villes pendant la tournée.

Fonction de fitness = minimiser (coût distance + coût de temps)

La figure 4.2 représente une partie de notre code en relation avec la fonction de fitness

```

#La fonction fitness
def getFit(self):
    fit = coût_distance = coût_temps = 0
    dist = []
    # calculer distance
    i = 1
    while i < len(self.data):
        calculateDist = lambda x1, y1, x2, y2: math.sqrt(((x1 - x2) ** 2) + ((y1 - y2) ** 2))
        dist.append(calculerDist(X[self.data[i]], Y[self.data[i]], X[self.data[i - 1]], Y[self.data[i - 1]]))
        i += 1

    # coût de distance
    coût_distance = sum(dist) * CK1m
    # coût temps
    temps = 0
    for i, pos in enumerate(self.data):
        # ignorer le centre
        if i == 0:
            continue
        # nouveau véhicule
        elif pos == centre:
            temps = 0
        # mise à jour le temps de la tournée
        temps += (dist[i - 1] / Vitesse)
        # arrive avant fenetre de temps
        if temps < eh[pos]:
            temps = eh[pos]
        # arrive plus tard que le début de fenetre de temps
        elif temps > lh[pos]:
            coût_temps += ((temps - lh[pos]) * lpu)
        # Mise à jour de temps ( temps de service )
        temps += h[pos]

    fit = coût_distance + coût_temps
    
```

Figure 4.2 Le code source de la fonction fitness.

4 Environnement matériel :

Notre application va être réalisée sur une machine qui comporte les caractéristiques suivantes :

- ✓ Marque : Dell.
- ✓ Modèle : INSPIRON 15.
- ✓ Processeur : Intel® Core™ i5-3337U CPU © 1.80 GHz 1.80 GHz
- ✓ Mémoire installée (RAM): 4.00 Go.
- ✓ Type du système : Système d'exploitation 64bits.
- ✓ Système d'exploitation : Windows 10 édition professionnel.

5 Environnement de développement :

❖ Python :

Le langage de programmation Python a été créé en 1989 par Guido van Rossum, aux Pays-Bas. Le nom Python vient d'un hommage à la série télévisée Monty Python's Flying Circus dont G. van Rossum est fan. La première version publique de ce langage a été publiée en 1991. La dernière version de Python est la version 3. Plus précisément, la version 3.7 a été publiée en juin 2018. La version 2 de Python est désormais obsolète et cessera d'être maintenue après le 1er janvier 2020. Dans la mesure du possible évitez de l'utiliser.

La Python Software Fondation 1 est l'association qui organise le développement de Python et anime la communauté de développeurs et d'utilisateurs. [40]



Figure 4.3 Langage de programmation Python.

❖ PyCharm

PyCharm est un IDE (Integrated development environment ou Environnement de développement intégré) spécifiquement développé pour travailler avec le langage de développement Python et les gabarits de Django, le framework de développement web écrit avec ce langage.

Pour réussir, PyCharm inclut toute une liste d'outils comme :

- La complétion automatique de code.
- Le support pour fenêtres divisées.
- Éditeur de code HTML, CSS et JavaScript.
- Interface VCS pour Mercurial, Perforce,...
- Épurateur de code.
- Développement d'applications pour Google App Engine. [42]

6 Expérimentation

Dans (la figure 4.4) nous affichons l'interface d'accueil de l'application avec quelques lignes de code source.

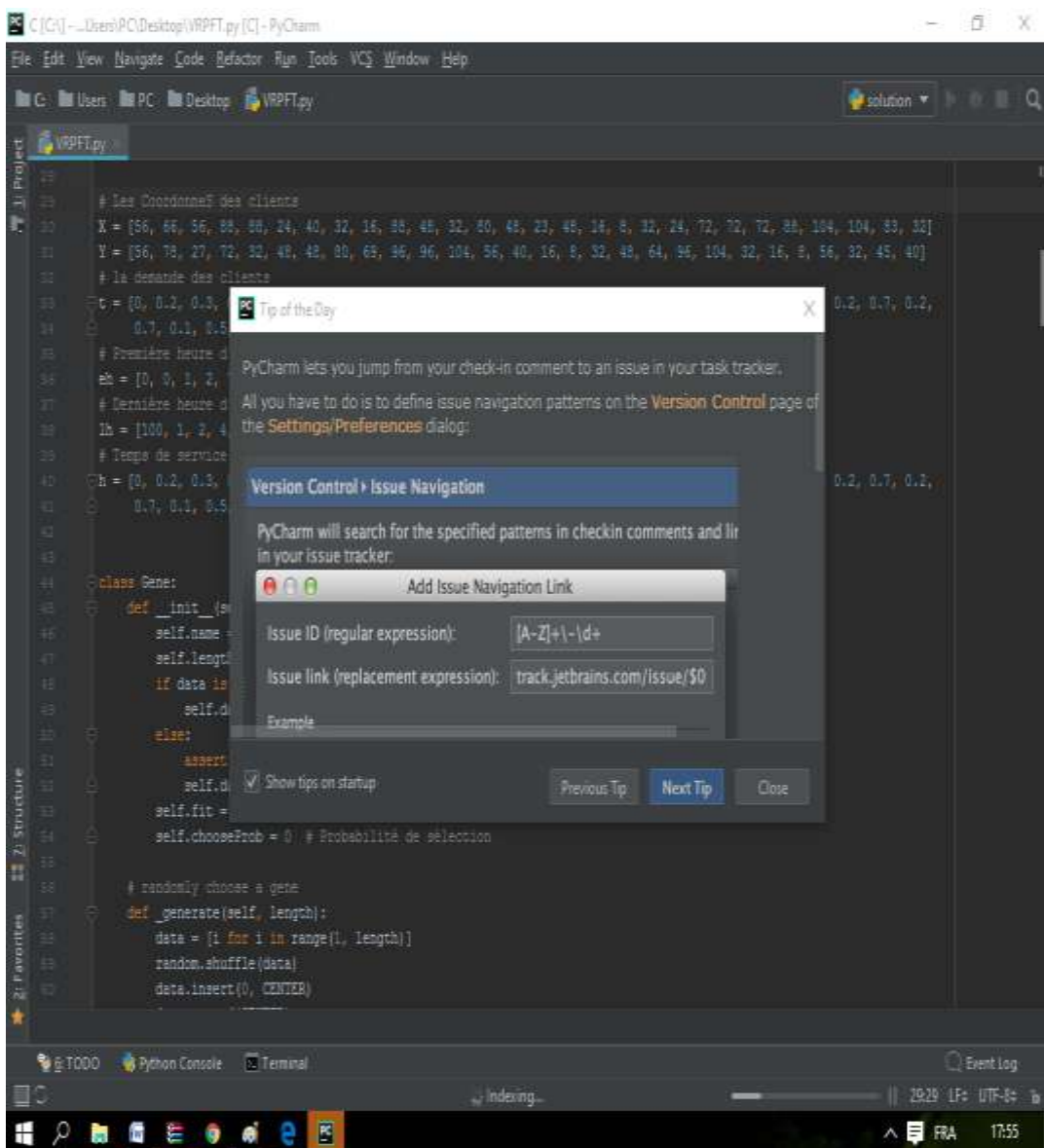


Figure 4.4 L'interface de l'application console.

Après l'exécution de l'application avec les paramètres suivants :

- La taille de population = 1000
- Nombre d'itérations = 3000
- Nombre de véhicules = 3
- Nombre des clients = 25

Nous obtiendrons alors le résultat suivant :



Figure 4.5 L'exécution de l'application console.

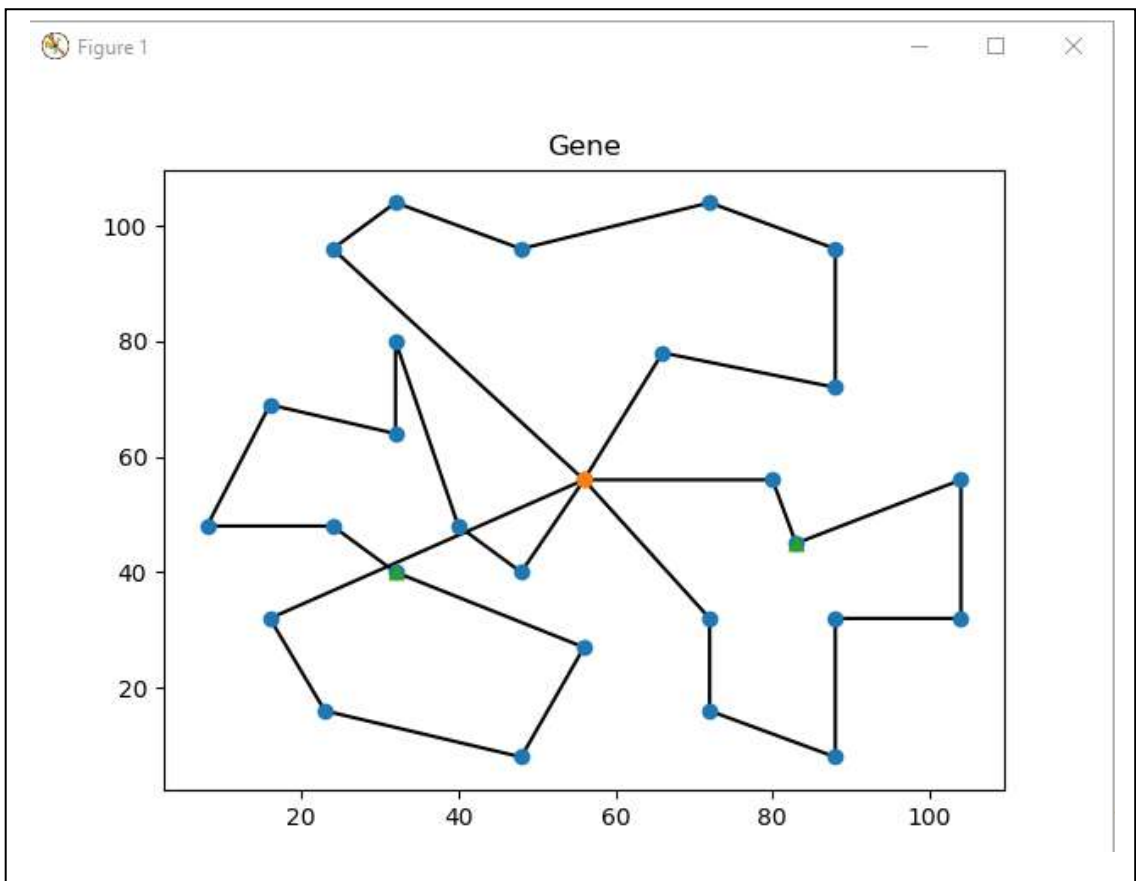


Figure 4.6 Fenêtre graphique de l'exécution.

7 Quelques tests sur l'implémentation

Test 01 :

Le but du test est de distinguer l'influence de la taille de la population sur le coût minimal (la solution optimale) et la performance.

Les paramètres sont choisis comme suit :

- Nombre d'itérations : 3000
- Probabilité de mutation : 0.05
- Probabilité de croisement : 0.4

Les résultats sont reportés sur le tableau suivant (tableau 4.1) et montré sur le graphe figure 4.7 suivant :

Taille de population	Valeur fitness trouvée
50	16452.0012
100	16524.0011
200	16011.2554
300	15821.1256
400	15125.4587
500	14345.1485
600	14001.1254
700	13125.1536
800	12987.3698
900	11011.2548
1000	9654.0008

Tableau 4.1 Expérimentation pour mesurer l'influence de la taille de populations.

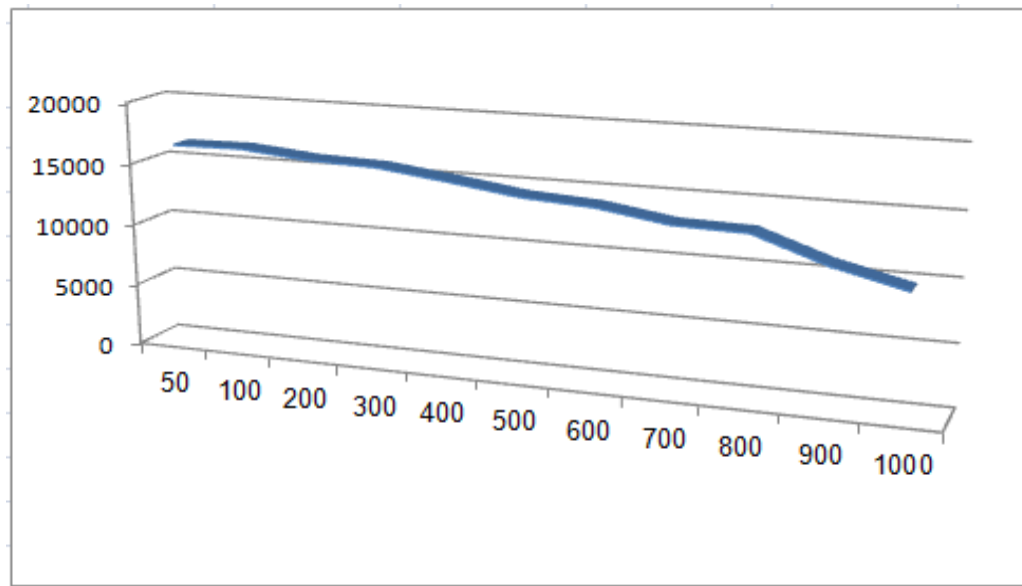


Figure 4.7 l'influence de la taille de population sur la solution optimale.

Test 02 :

Le but du test est de distinguer l'influence du nombre d'itérations sur le coût minimal (la solution optimale) et la performance.

Les paramètres sont choisis comme suit :

- Nombre de population : 500
- Probabilité de mutation : 0.05
- Probabilité de croisement : 0.4

Les résultats sont reportés sur le tableau suivant (tableau 4.2) et montré sur le graphe (figure 4.8)

Le nombre d'itérations	Valeur fitness trouvée
100	21510.2548
200	20025.25
600	19215.3215
1200	17548.0021
2400	11350.0022
2600	11354.1254
2700	11348.1245
2800	11340.1258
2900	11320.1414

Tableau 4.2 Expérimentation pour mesurer l'influence du nombre d'itérations.

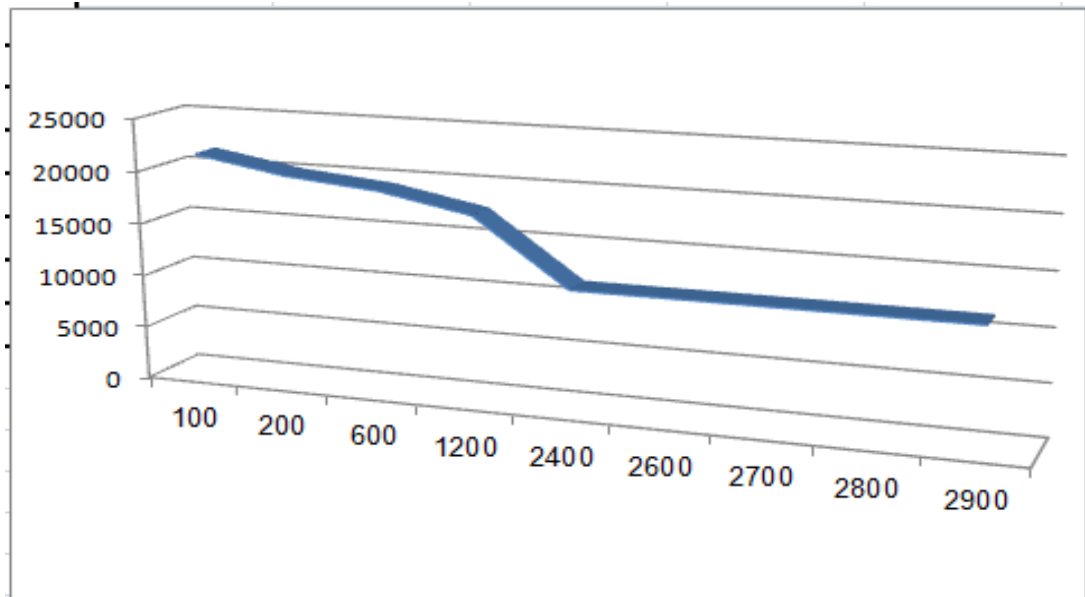


Figure 4.8 l'influence de nombre d'itérations sur la solution optimale.

8 Discussion des résultats :

Les deux tableaux présentés juste au-dessus, sont bien visibles et bien expliqués dans ces deux graphes, où on remarque que :

- La première courbe présente : amélioration légèrement jusqu'à $n=800$. En $n=900$ la courbe stabilise dans la solution optimale
- La deuxième courbe présente : une décroissance accentuée vers une solution qui restera stable car les populations dans la deuxième partie n'ont pas pu la dépasser, et ceci de la taille de la population à $n=300$ à 650, en $n=700$ la solution est de nouveau améliorée et continue légèrement s'améliorée jusqu'à $n=2300$. En $n=2400$ une brusque chute de la courbe améliore nettement la solution optimale.

9 Conclusion

Nous avons mené une étude sur l'influence des paramètres sur la performance de l'algorithme génétique à trouver des solutions optimales au problème considéré qui est le VRPFT. On n'est arrivée à la conclusion que la taille de population et le nombre de génération ont une influence sensible sur la performance de l'algorithme néanmoins leurs valeurs doivent être bien étudiées pour sélectionner les meilleures qui peuvent apporter plus de performance à l'algorithme génétique.

CONCLUSION GENERALE

Parmi les problèmes de la chaîne logistique on cite le problème du transport où les entreprises de distribution doivent livrer et collecter des colis afin de satisfaire leurs clients.

Dans le cadre de ce mémoire, nous sommes intéressés par une variante importante de ces problèmes qui est le problème de tourné de véhicules avec fenêtres de temps (PTVFT) qui le client doit être servi dans un intervalle de temps défini, connu d'avance par le livreur et toute violation de cette contrainte peut engendrer une pénalité. Lorsque la contrainte de fenêtre de temps n'est pas satisfaite, soit on rejette la solution si on considère le cas rigide ou bien on construit une fonction de pénalité qui sera rajoutée ou combinée avec la fonction objective pour le cas relâché. En réalité, c'est un problème très fréquent. La distribution des produits périssables (le lait, la viande, les journaux et les services ambulatoires. . .)

Dans cette classe de problèmes, on distingue deux sous-classes : le VRPTW rigide où le service doit impérativement être effectué dans la fenêtre de temps et le VRPTW relâché où le retard ou l'avance engendre uniquement une pénalité.

Nous avons proposé notre approche pour résoudre le VRPTW principalement basé sur l'algorithme génétique.

Notre algorithme génétique développé a prouvé son efficacité en:

- 1) La minimisation de la route des véhicules.
- 2) La minimisation du temps de transport.

Ce travail est préoccupé par un vendeur considéré comme un administrateur. Dans une perspective d'amélioration nous proposons d'utiliser une méta-heuristique pour avoir meilleur résultat.

Bibliographie

- [1] DIOMANDE Namory, Le transport dans la stratégie de production des grandes entreprises, Diplôme d'Ingénieur des Travaux Publics, Ecole Supérieure des Travaux Publics de l'Institut National Polytechnique Félix Houphouet-Boigny, France, 2007.
- [2] BARBARA Lyonnnet, et Marie-Pascale Senkel , La logistique, Dunod, 2015.
- [3] CHEYROUX Laurent, Sur l'évaluation de performances des chaînes logistiques, Thèse Doctorat en automatique et productique, Institut National Polytechnique de Grenoble-INPG, 2003.
- [4] FRANCOIS Julien, Planification des chaînes logistiques: Modélisation du système décisionnel et performance, Thèse Doctorat en productique, Université Sciences et Technologies-Bordeaux I, France, 2007
- [5] HAFEZ Névine, Conditions d'équilibre et gestion d'unités de transport en libre service avec demandes aléatoire, Thèse Doctorat en informatique, Université de Metz-France, 1999.
- [6] TOUIL, Rima, Assia AOUS, Proposition d'un protocole de transport pour l'Internet physique, Mémoire Master en génie productique. Université Abou Bekr Belkaid-Tlemcen, 2016.
- [7] TROJET Mariem, Planification d'une chaîne logistique: Approche par satisfaction de contrainte dynamiques, Thèse Doctorat en génie industriel, Université de toulouse, 2014.
- [8] A. Gherbouj, Méthodes de résolution de problèmes difficiles académiques, Thèse Doctorat en informatique, université Constantine, 2013.
- [9] SARIKLIS, Dimitrios, and Susan Powell. "A heuristic method for the open vehicle routing problem." Journal of the Operational Research Society, 2000.
- [10] MOUNIR, Mohamed Elhavedh Ould Ahmed, Contribution à la résolution du sac-à- dos à contraintes disjonctives. Thèse Doctorat en informatique, Université de Picardie Jules Verne, 2009.
- [11] MANCEL Catherine, modélisation et résolution de problèmes d'optimisation combinatoire issus d'applications spatiales, Thèse Doctorat en systèmes industriels, l'Institut National des Sciences Appliquées de Toulouse (INSA), 2004.
- [12] BOURAZZA Saïd, Variantes d'algorithmes génétiques appliquées aux problèmes d'ordonnancement, Thèse Doctorat en mathématiques appliquées et informatique, Université du Havre, 2006.

- [13] MEZIANE M. E. A., Optimisation par phases pour les problèmes d'ordonnancement des ateliers de type job-shop totalement flexibles, Mémoire Magister en sciences, Université d'Oran, Algérie, 2011.
- [14] SOPHIE Jacquin, hybridation des métaheuristiques et de la programmation dynamique pour les problèmes d'optimisation mono et multi-objectif : application à la production d'énergie, Thèse Doctorat en informatique, Université de Lille 1, 2015.
- [15] ZEBDA Nadjela, Optimisation Globale de la chaîne logistique en présence des nouvelles technologies de l'internet des objets, Mémoire Master en informatique (SIA), Université de M'sila, 2017.
- [16] MARINO W., Les métaheuristiques : des outils performants pour les problèmes industriels, Thèse Doctorat en informatique, Université de Fribourg France, 2001.
- [17] ZAGHDOUD, Radhia, Hybridation d'algorithme génétique pour les problèmes des véhicules intelligents autonomes: applications aux infrastructures portuaires de moyennetaille, Thèse Doctorat en génie informatique, Ecole centrale de Lille, 2015.
- [18] Gunantara, N., Putra, N., & Nyoman, The Characteristics of Metaheuristic Method in Selection of Path Pairs on Multicriteria Ad Hoc Networks, Journal of Computer Networks and Communications, 2019.
- [19] BENKADOUR H., Aribi R., Méta-heuristiques parallèles pour la résolution des problèmes difficiles, Mémoire Master en informatique, Université Kasdi Merbah Ouargla, 2013.
- [20] HAJ-RACHID, Mais et al., Différents opérateurs évolutionnaires de permutation: sélections, croisements et mutations, Rapport de recherche, Laboratoire D'informatique, Université de Franche-Comte, 2010.
- [21] DOUIRI Mohamed, Méthodes de Résolution Exactes Heuristiques et Métaheuristiques, Mémoire Master en informatique, Université d'Oran, 2008.
- [22] CAN, Umit et ALATAS, Bilal, Physics based metaheuristic algorithms for global optimization, American Journal of Information Science and Computer Engineering, 2015.
- [23] GHERBI Samiha, Algorithmes génétique appliqué à un problème du voyageur de commerce, Mémoire Master en informatique, Université de m'sila, Algérie, 2013.
- [24] BOISSON Jean, Modélisation et résolution par métaheuristiques coopératives: de l'atome à la séquence protéique, Thèse de Doctorat en informatique, Université Lille 1, 2008.
- [25] LAITITIA J., Métaheuristiques pour l'extraction connaissances application à la génomique, Thèse Doctorat en informatique, Université des sciences technologique de LILLE U.F.R., 2003.

- [26] C.H. PAPADIMITRIOU, K. STEIGLITZ, Combinatorial optimization: algorithms and complexity. Prentice Hall, 1982.
- [27] LAYEB Abdeslam, Utilisation des Approches d'Optimisation Combinatoire pour La Vérification des Applications Temps Réel, Thèse Doctorat en informatique, Université Mentouri de Constantine, 2010.
- [28] C.C. RIBEIRO, N. MACULAN, Applications of combinatorial optimization, Annals of Operations Research, 1994.
- [29] CETIN, Suna, and Cevriye Gencer, A heuristic algorithm for vehicle routing problems with simultaneous pick-up and delivery and hard time windows, Open Journal of Social Sciences, 2015.
- [30] CHASSAING Maxime, Problèmes de transport à la demande avec prise en compte de la qualité de service, Thèse Doctorat en informatique, Université Blaise Pascal-Clermont-Ferrand II, 2015.
- [31] MICHALLET Julien, Problèmes de tournées de véhicules périodiques avec contraintes de sécurité ou de qualité de service, Thèse Doctorat en optimisation, Troyes, 2013.
- [32] AKIL, Meriem, Problème de tournées de véhicules avec contraintes et fenêtre de temps. Mémoire Magister en informatique, UMMTO, 2013.
- [33] CADET David Joseph, Optimisation des flux : application aux problèmes de distribution en nutrition animale, Thèse Doctorat en optimisation des systèmes, Université de Technologie de Troyes, 2013.
- [34] HAMANI Amir, Delopment of à tool for optimization vehicle routing problem, Mémoire Master en informatique décisionnelle et optimisation, Université de m'sila, Algérie, 2018.
- [35] MARSİ Loubna, Etude de cas d'un problème de tournées des véhicules à la société SNTL, Mémoire Licence en sciences et techniques, Université Sidi Mohamed Ben Abdellah-Faculte des Sciences et Techniques, 2015.
- [36] ZHAO Xin, Une méthode génétique pour la résolution du problème dynamique de routage de véhicules avec temps de parcours variables, Thèse Doctorat en informatique. Université d'Artois, 2008.
- [37] MONTOYA JA, Electric Vehicle Routing Problems: models and solution approaches, Thèse Doctorat en informatique, Université d'Angers, 2016.
- [38] HOUSROUM Haiyan, Une approche génétique pour la résolution du problème VRPTW dynamique, Thèse Doctorat en informatique, Université d'Artois, 2005.

ملخص

يحتل التحسين التوافقي مكانا مهما للغاية في بحوث العمليات، الرياضيات وعلوم الكمبيوتر. إن مشكلة توجيه السيارة مع احترام الوقت المحدد (PTVFT) هي مشكلة تحسين توافق مع الإشكالية التالية: "ما هي المجموعة المثلى من المسارات لأسطول من المركبات من أجل التسليم إلى مجموعة معينة من العملاء مع احترام مواعيد التسليم و العودة إلى المستودع عند الانتهاء ؟ هدفنا في هذا العمل هو تقليل المسافة التي تقطعها المركبات و الوقت المستهلك بتطبيق الخوارزميات الجينية باستعمال لغة البرمجة "بايثون".

الكلمات المفتاحية : مشكلة توجيه السيارة مع احترام الوقت PTVFT , التحسين , الخوارزميات الجينية, Python.

Abstract

Combinatorial optimization occupies a very important place in operations research, discrete mathematics and computer science. The Vehicle Routing problem with time window (VRPTW) is a combinatorial optimization that answers: «What is the optimal set of routes for a fleet of vehicles to traverse in order to deliver to a given set of customers with respect delivery times and returning to the warehouse once completed»? Our goal in this work is to minimize the cost of distance traveled and the time spend, by applying genetic algorithms as heuristics and by using the "Python" programming language for coding.

Keywords: Vehicle routing problem with time window VRPTW, optimization, genetic algorithms, Python.

Résumé

L'optimisation combinatoire occupe une place très importante dans la recherche opérationnelle, en mathématiques discrètes et en informatique. Le problème de tournée de véhicule avec fenêtre de temps (PTVFT) est un problème avec la problématique : "Quel est l'ensemble optimal de routes à traverser afin de livrer à un ensemble donné de clients en respectant les délais de livraison et en retournant à l'entrepôt une fois terminé" ? L'objectif de ce travail est de minimiser le coût de distance parcourue ainsi le temps, en appliquant les algorithmes génétiques comme heuristique et en utilisant le langage de programmation "Python" pour le codage.

Mots-clés : Problème de tournée de véhicule avec fenêtre temps PTVFT, optimisation, algorithmes génétiques, Python.