



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE DE BATNA 2 – MUSTAPHA BEN BOULAID
FACULTE DE TECHNOLOGIE
DEPARTEMENT GENIE INDUSTRIEL



THESE

PRÉSENTÉE AU LABORATOIRE D'AUTOMATIQUE ET PRODUCTIQUE
EN VUE DE L'OBTENTION DU DIPLÔME DE DOCTORAT LMD

Spécialité : Génie industriel

Option : Informatique industrielle et productique

Par

Lahcene GUEZOULI

Licence en Informatique, option informatique

Master en informatique, option ingénierie de réseau et communication

THEME

*Développement d'une approche hybride multi-critères basée clustering
pour la résolution d'un problème de distribution de produits*

Thèse dirigée par Mr. **Samir ABDELHAMID**

MCA. Université de -BATNA 2-

Soutenue publiquement le : 08/12/2018

Devant le jury composé de :

Président : *Leila Hayet Mouss*

PROF. Université de – BATNA 2 –

Rapporteur : *Samir Abdelhamid*

MCA. Université de – BATNA 2 –

Examineur : *Abdelkamel Tari*

PROF. Université de – BEJAIA –

Examineur : *Mohamed Benmohammed*

PROF. Université de – CONSTANTINE 2 –

Examineur : *Djamel Mouss*

PROF. Université de – BATNA 2 –

Remerciement

Je tiens à exprimer ici toute ma reconnaissance au Dr ***Samir Abdelhamid*** pour l'honneur qu'il m'a fait, en assurant la direction et le suivi scientifique et technique de ce travail, pour leur précieux conseils et remarques constructives, sa patience et pour tout le temps qu'il m'a consacré. J'apprécie aussi sa grande sympathie qui a rendus le travail à ses côtés très agréable.

Je remercie vivement Pr ***Leila Hayet Mouss***, pour l'honneur qu'elle me fait en acceptant de Présider le jury de ma soutenance.

Je remercie également les professeurs : ***Djamel Mouss, Abdelkamel Tari*** et ***Mohamed Benmohammed*** pour avoir accepté d'examiner ce travail.

Finalement je tiens aussi à remercier, Dr ***Larbi Guezouli***, Dr ***Lyamine Guezouli***, Pr ***Abdelkamel TARI***, tous mes collègues du laboratoire LAP et tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste travail.

Veuillez bien trouver ici le témoignage de ma profonde estime.

DEDICACES

Je dédie ce modeste travail



A mon cher père et ma chère mère ;



A ma chère épouse N.E. Bendjemâa;



A mes chères sœurs : Louisa et Linda ;



*A mes chers frères : Larbi, Lotfi,
Lyamine et Lyes ;*



A tous mes amis et proches ;

Qui m'ont toujours accordé leurs amours

Et leurs sacrifices.

Lahcene Guezouli.

ملخص

إن سوء إدارة الطرق التي يسلكها السائق في شبكة الطرق، مثل شبكة الطرق الجزائرية، يمكن أن يسبب عدة مشاكل، مثل: انتهاء صلاحية المنتجات الموزعة، استهلاك الوقود غير الضروري، فقدان الوقت والتلوث ... وتولد هذه المشاكل احتياجات جديدة للشركات التي تبحث عن حلول لتحسين إدارة مسارات توزيع منتجاتها، واكتساب ثقة عملائها. تحقيقاً لهذه الغاية، وبهدف مساعدة السائق في مهمة اتخاذ القرار الصحيح، تم تصميم نظام دعم القرار (SIAD) "الذي يمثل جوهر حلنا" الذي يسمح باختيار أفضل الطرق التي يتعين اتخاذها من قبل المركبات أثناء مهمة التوزيع. هذا الأخير يستخدم نظام لتحديد المواقع العالمي (GPS) كمورد رئيسي، من أجل استخراج خريطة للمسارات الممكنة وكذلك تحديد موقع وإحداثيات المركبات المستعملة.

في هذه الأطروحة، يهدف نظام دعم القرار الخاص بنا إلى تحسين المشكلة الكلاسيكية لتوجيه السيارات من خلال النظر في وجود أنواع مختلفة من المركبات (ذات ساعات وتكاليف متميزة) وعدة مستودعات، والتي نسميها مشكلة توجيه السيارات الغير المتجانسة مع إلزامية احترام نافذة الوقت و توفر عدة مستودعات (MDHVRPTW) و من خلال احترام مجموعة من المعايير بما في ذلك: جدول طلبات الزبائن، الساعات المختلفة للمركبات و لحل هذه المشكلة نقترح مخطط جديد يستند إلى تطبيق واحد من الاستدلالات المستوحاة من الطبيعة "الحيوية" و المعروفة بـ "الخوارزميات الجينية" واثنين من طرق التجميع المخصصة (KNN و Kmeans) التي تعمل على دمج خوارزمية التجميع (Clustering) في إطار تحسين نتائج (VRPTW).

أجريت عدة تجارب حسابية استناداً إلى حالات الاختبار المعياري « the benchmarks test » ، وتمت مقارنة النتائج التي تم الحصول عليها إلى أفضل النتائج التي تم الحصول عليها في مسائل مماثلة من حيث الحلول المتحصل عليها ووقت المعالجة، لاختبار فعالية المخطط المقترح.

الكلمات المفتاحية: النظام العالمي لتحديد المواقع، نظام دعم القرار، مسألة توجيه المركبات المتعددة المستودعات، التجميع، التوجيه، الجدولة، الخوارزمية الجينية.

Résumé

La mauvaise gestion du trajet pris par le conducteur dans un réseau routier, tel que celui de l'Algérie, peut causer plusieurs problèmes, tels que : l'expiration des produits distribués, la consommation de carburant inutile, la perte de temps, la pollution ... Ces problèmes génèrent de nouveaux besoins pour les entreprises qui cherchent des solutions pour mieux gérer les trajets de distribution de leurs produits, et gagner la confiance de leurs clients. Pour cela, et dans le but d'aider le conducteur dans sa tâche de prise de la bonne décision, nous avons opté pour la conception et la réalisation d'un système d'information d'aide à la décision (SIAD), qui représente le noyau de notre solution, permettant de sélectionner les meilleurs chemins à prendre par les véhicules lors de la tâche de distribution. Ce dernier utilise un GPS comme ressource principale, afin d'extraire la carte des chemins possibles ainsi que de positionner les véhicules utilisés.

Dans cette thèse, le système d'information d'aide à la décision vise à optimiser le problème classique de tournée de véhicules en considérant l'existence de différents dépôts et d'une flotte composée de véhicules hétérogènes (avec des capacités et des coûts distincts), The Multi Depot Heterogeneous Vehicle Routing Problem with Time Windows (MDHVRPTW), en respectant un ensemble de critères, y compris : la planification des demandes des clients, la capacité hétérogène des véhicules ... Nous résolvons ce problème en proposant un nouveau schéma basé sur l'application d'une hybridation entre une des métaheuristiques bio-inspirées nommée « les algorithmes génétiques » et deux méthodes dédiées clustering (l'heuristique du Plus Proche Voisin et la méthode d'apprentissage Kmeans) qui permettent d'intégrer la notion de regroupement « clustering » dans un cadre d'optimisation VRPTW.

Les expériences computationnelles ont été effectuées en se basant sur les instances de tests de références « the benchmarks tests », et les résultats obtenus ont été comparés aux meilleurs résultats déjà obtenus suite à l'optimisation d'autres problèmes similaires en termes de solutions générées et de temps de traitement, afin de tester l'efficacité de l'approche proposée.

Mots-clés: SIAD, VRP, HVR, GPS, Regroupement, Routage, Planification, Algorithmes génétiques.

Abstract

The poor management of the routes taken by the driver in a road network, such as that of Algeria, can cause several problems, such as: the expiration of distributed products, unnecessary fuel consumption, loss of time, pollution ... These problems generate new needs for companies looking for solutions to better manage the distribution routes of their products, and gain the trust of their customers. For such needs, and in order to assist the driver in the task of making the right decision, a decision support system (DSS) has been designed and implemented "which represents the core of our solution" to select the best paths to be taken by the vehicles during the distribution task. The later uses a global positioning system (GPS) as a main resource, in order to extract the map of the possible paths as well as locates the used vehicles.

In this thesis, our decision support system aims to optimize the classical capacitated vehicle routing problem by considering the existence of different vehicle types (with distinct capacities and costs) and multiple available depots, that we call the multi-depot heterogeneous vehicle routing problem with time window (MDHVRPTW) by respecting a set of criteria including: planning client requests, the heterogeneous capacity of vehicles..., and we solve this problem by proposing a new scheme based on the application of a hybridization between one of the bio-inspired metaheuristics named "genetic algorithms" and two neighborhoods heuristics dedicated clustering (the Nearest Neighbor heuristics and the Kmeans learning method) that allow to integrate a clustering algorithm in a VRPTW optimization framework.

The computational experiments were performed based on test benchmarks, and the obtained results were compared to the best results already obtained by optimizing other similar problems in terms of generated solutions and processing time, to test the effectiveness of the proposed approach.

Keywords: MD-VRP, HVRP, GPS, DSS, Clustering; Routing, Scheduling, Genetic Algorithms.

TABLE DES MATIÈRES

Introduction générale	12
Chapitre 1 : Les Problèmes de transports	18
1. Introduction	19
2. Le problème de transport.....	19
2.1. Le problème du voyageur de commerce	21
2.2. Le problème du postier chinois	24
2.3. Problème de tournées de véhicules	24
3. Conclusion	36
Chapitre 2 : Résolution des problèmes « VRP ».....	37
1. Introduction	38
2. Problème d'optimisation	38
3. Les méthodes exactes pour le VRP	39
3.1. Procédure de séparation et d'évaluation	39
3.2. Procédure de coupes et de séparation.....	40
3.3. La programmation dynamique	40
4. Méthodes heuristiques pour le VRP	41
4.1. Heuristiques de résolution.....	41
4.2. Les méta-heuristiques.....	50
5. Conclusion	60
Chapitre 3 : Modélisation du processus de distribution de produits.....	62
1. Introduction	63
2. Le réseau routier en Algérie « extraction des informations par le module GPS »	63
3. Le parc automobile en Algérie	64
3.1. Gestion du parc d'automobiles.....	65
4. L'espace géographique	65
5. La marchandise.....	66

6.	La consommation du carburant en Algérie.....	66
7.	Formulation mathématique du MD-VRPTW « Comment résoudre »	67
7.1.	Introduction	67
7.2.	Modélisation.....	67
7.3.	La Formulation du problème.....	69
7.4.	Formulation	72
8.	Conclusion.....	74
Chapitre 4 : Résolution mono-objectif du MD-VRPTW par algorithmes génétiques « ProGenClust »		75
1.	Introduction	76
2.	Codage du chromosome	79
3.	Décodage du passage des véhicules	80
4.	Croisement.....	80
5.	Mutation.....	81
5.1.	Mutation « Intra-Dépôt ».....	81
5.2.	Mutation Inter-Dépôts	82
6.	Génération de la population initiale.....	82
6.1.	Le regroupement (Clustering)	83
6.2.	Le routage.....	84
6.3.	L'ordonnancement	84
7.	Procédure de calcul du « fitness »	86
8.	L'algorithme génétique proposé pour la minimisation du coût total de transport	87
9.	Conclusion.....	88
Chapitre 5 : Optimisation multi-objectif pour la résolution du « MD-VRPTW ».....		89
1.	Introduction	90
2.	Résolution d'un problème de décision multi-objectif	90
3.	Approches de résolution d'un problème de décision multi-objectif.....	92
4.	Classification du point de vue du décideur.....	93
5.	Classification du point de vue du concepteur.....	93
5.1.	Approche scalaire.....	93

5.2.	Approches Pareto	95
5.3.	Approches non scalaires et non Pareto.....	96
6.	VRP et multi-objectif.....	97
7.	L'approche multicritères proposée pour l'optimisation de notre problème ...	98
8.	Conclusion	99
Chapitre 6 : Simulations et résultats		100
1.	Introduction	101
2.	Instances de tests utilisés pour l'évaluation expérimentale.....	101
3.	Résultats expérimentaux.....	103
4.	Conclusion	115
VII.	Conclusion générale & perspectives.....	116
Bibliographie.....		117

LISTE DES TABLAUX

Table IV.1 Codage par liste de permutation	79
Table IV.2. Ordre du passage des véhicules	80
Table VI.1. Les spécifications des 20 benchmarks utilisés	102
Table VI.2. Les paramètres expérimentaux	103
Table VI.3. Les routes optimales résultantes de la simulation basée sur notre approche ProGenClust	107
Table VI.4. Moyenne des résultats pour FSMVRP avec coût fixe pour 12 problèmes de tests	112
Table VI.5. Comparaison de notre algorithme génétique « ProGenClust» avec l'algorithme génétique connu « GenClust » [118]	113

LISTE DES FIGURES

Fig. I.1 Schéma global de notre système	14
Fig. I.1	21
Fig. I.2. Exemple de TSP	21
Fig. I.3 Une solution d'un problème de tournées de véhicules	28
Fig. II.1. Méthode avec économies	41
Fig. II.2 Méthode d'insertion de Mole et al.	42
Fig. II.3 Voisinages agissant sur plusieurs tournées	49
Fig. II.4 Minimums locaux et minimum global	49
Fig. II.5 Récapitulatif des méthodes de résolution	59
Fig. III.1 Augmentation du parc automobile en Algérie « http://www.ons.dz/-Parc-Automobile-.html »	65
Fig. III.2 Consommation du gas-oil / Secteur d'activités -2016- (http://www.memalgeria.org , http://www.mtp.gov.dz).	67
Fig. III.3 Exemple de MD-VRPTW avec 2 dépôts et 10 clients	68
Fig. III.4 Les étapes de prise de décision dans MD-VRPTW	68
Fig. IV.1 Principe de fonctionnement d'un algorithme génétique	77
Fig. IV.2 Schéma général de ProGenClust	79
Fig. IV.3 Exemple de Mutation par inversion	81
Fig. IV.4 Exemple de regroupement « clustering »	83
Fig. IV.5 Exemple d'un processus d'optimisation d'un problème MD-VRPTW	85
Fig. V.1 Exemple de dominance.	91
Fig. VI.1 Interface de GUIMOO.	104
Fig. VI.2 Création d'un fichier pour le problème	105
Fig. VI.3 Création un fichier des données	105
Fig. VI.4 Front de Pareto	106
Fig. VI.5 Exemple de routes optimisées utilisant ProGenClust pour l'instance p04111	
Fig. VI.6 Le nuage de points représentant les écarts entre les résultats de ProGenClust et GenClust [118]	114

LISTE DES ALGORITHMES

Algo IV.1 Méthodologie des algorithmes génétiques.....	78
Algo IV.2 Algorithme d'ordonnancement.....	84
Algo IV.3 Algorithme de correction de capacité.....	85
Algo IV.4 L'algorithme génétique proposé pour résoudre un mono-objectif MD-VRPTW	87
Algo V.1 Structure de l'algorithme génétique multicritères proposé pour l'optimisation de notre problème FSMVRPTW	98

INTRODUCTION GÉNÉRALE

De nos jours, la distribution de produits se fait sur des chemins compliqués et de plus en plus encombrés. La croissance importante au niveau des réseaux routiers en Algérie, la création des zones industrielles ainsi que l'importante augmentation du parc d'automobile, ont engendrés un encombrement dans la circulation des véhicules, surtout au niveau des centres villes, et une augmentation dans la consommation du carburant et parfois la pénurie de cette matière essentielle dans l'industrie et le transport.

Dans l'industrie, la résolution du problème de transport a toujours été la clé des problèmes de logistique, cette dernière est devenue une composante essentielle du bon fonctionnement des entreprises. L'aspect très compétitif rend crucial la recherche de solutions logistiques efficaces qui tendent à vouloir maximiser les bénéfices ou de manière équivalente à minimiser les pertes, c'est-à-dire réduisant le coût de transport, le nombre de véhicules utilisés, les délais de livraisons, l'émission de CO₂...

De manière naturelle, on cherche soit à minimiser la durée ou la distance, ou encore, maximiser la quantité de la marchandise transportée par les véhicules. Ainsi, pour répondre à ce besoin constant d'optimisation, des efforts ont été fournis pour le développement d'outils formels aidant à atteindre l'optimalité relativement aux décisions qu'ils doivent être prisent. L'objectif de ces outils d'optimisation est de fournir une solution de qualité à un problème donné en un temps raisonnable.

Dans le cas des problèmes de distribution de produits, plusieurs critères doivent être prisent en considération lors de la phase d'optimisation tel que :

- Le type du produit à distribuer qui joue un rôle très important dans le choix du chemin à prendre. Par exemple, la distribution des produits alimentaires est caractérisée par des livraisons fréquentes de petits volumes à de nombreux clients et dans un intervalle de temps étroit ;
- Le type de véhicules utilisés qui dépend de la quantité à distribuer ;
- La durée moyenne exigée d'une tournée qui représente le nombre d'heures moyen de travail du chauffeur ;
- Le nombre de dépôts disponibles qui représentent les points de départs ;
- Les différentes contraintes et exigences des clients tels que les heures d'ouvertures et de fermetures, la localisation géographique...

La distribution des produits sur un réseau routier, tel que celui de l'Algérie, est un contexte riche en complexité, et il est possible de le modéliser sous forme d'un problème de tournées de véhicules (VRP). La mauvaise gestion du trajet pris par le conducteur peut causer plusieurs problèmes, parmi lesquels on peut citer : l'expiration des produits distribués, perte de clients, consommation du carburant inutile, perte de temps, pollution ... Cette mauvaise gestion est due à la mauvaise prise de décision par le conducteur.

Pour aider le conducteur de prendre le bon trajet de distribution des produits, nous proposons un système d'information d'aide à la décision. Ce dernier, facilite la tâche du conducteur pour sélectionner facilement le meilleur chemin à suivre.

Nous nous servons des cartes géographiques d'un système de positionnement global pour tracer les chemins possibles et de localiser les véhicules durant leurs tournées.

La Fig. I.1 donne une vue globale sur le système proposé.

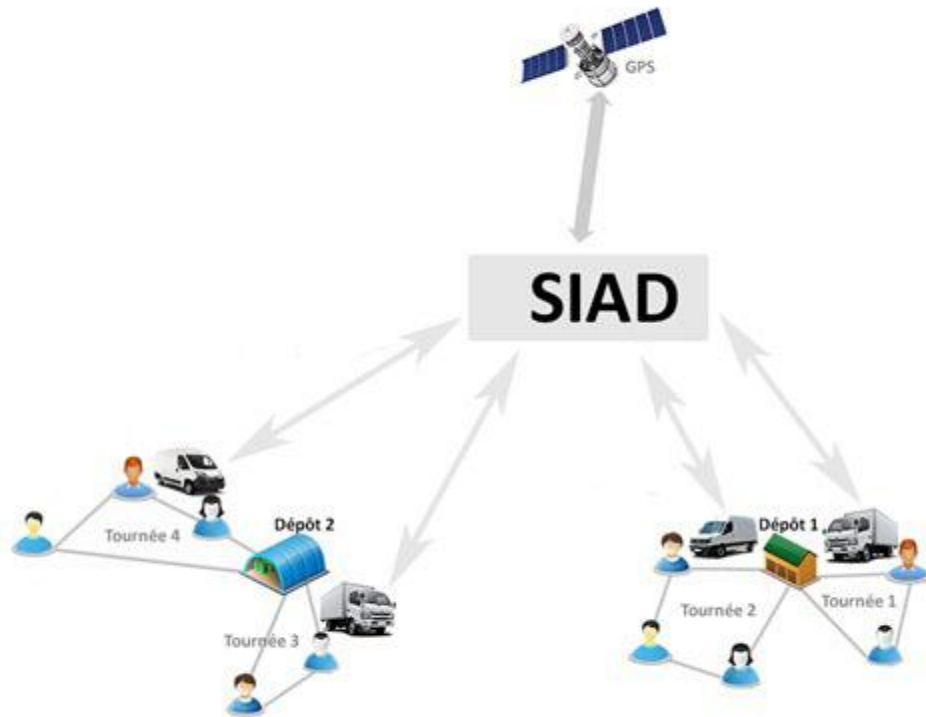


Fig. I.1 Schéma global de notre système

Dans ce présent travail, notre SIAD a pour but d'apporter de l'aide au conducteur pour sélectionner le meilleur chemin possible à prendre lors de son trajet de distribution. Donc, il est essentiel de faire appel à des méthodes d'optimisation dédiées à la résolution des problèmes de tournée de véhicule.

Plusieurs méthodes d'optimisation ont déjà été utilisées pour résoudre des problèmes de tournées de véhicule telles que les méthodes exactes qui permettent de fournir des solutions optimales pour de petits problèmes de tournée. Cependant, elles deviennent inapplicables à cause de l'augmentation exponentielle des temps de calcul en fonction de la taille des problèmes. Pour ce type de problèmes, nous avons fait recours à des méthodes heuristiques et métaheuristiques, qui sont des méthodes spécifiques au

problème traité permettant d'obtenir une solution de bonne qualité en un temps raisonnable. Même si ces heuristiques ne fournissent aucune garantie théoriquement prouvée sur la qualité du résultat, elles sont exploitées avec succès sur des problèmes pratiques majeurs.

Le développement d'une heuristique est une tâche difficile nécessitant une expertise importante. L'apparition des métaheuristiques a réduit cette difficulté en proposant des schémas de résolution heuristique pouvant être adaptés à différents problèmes d'optimisation.

L'un des domaines où l'on trouve que l'utilisation des métaheuristiques a donné des résultats satisfaisants est bien le domaine de tournée de véhicule, qui représente un domaine complexe et qui a attiré l'attention du monde académique tout en étant pertinent, car il est applicable dans un contexte réel. De plus, les solutions obtenues s'avèrent souvent encourageantes. Certaines entreprises réussissent à réaliser des économies, pouvant atteindre jusqu'à 20% d'après Toth et Vigo en 2001 [1].

Notre problème de base consiste en l'optimisation multicritère de la distribution de produits dans une entreprise économique. En déterminant les tournées et les itinéraires que doivent emprunter les conducteurs des véhicules qui démarrent d'un des différents dépôts disponibles visitant tous les clients et qui retournent à la fin au même dépôt. Cet objectif doit être atteint en minimisant le coût total de transport « en termes de kilométrage », le nombre de véhicules utilisés lors de la distribution et la somme totale des retards, tout en respectant les contraintes sur la capacité des véhicules disponibles, leurs fenêtres de temps de service, ... on parle alors de la variante « MD-VRPTW ».

Notre variante « MD Fleet Size and Mix VRP » diffère aussi de la VRP classique, sur le point qu'elle traite avec une flotte hétérogène de véhicules ayant différentes capacités « FSMVRP ». Par conséquent, la FSMVRP implique la conception d'un ensemble de tournées de véhicules, chacune d'elles commence et se termine à un dépôt donné, pour une flotte hétérogène de véhicules qui serve un ensemble de clients avec des demandes spécifiques. Chaque client est visité une seule fois, et la demande totale d'une route ne doit pas dépasser la capacité maximale de chargement du véhicule qui lui est assigné. Le coût de routage d'un véhicule est la somme de ses coûts associés proportionnellement à la distance parcourue.

Dans ce travail, nous proposons une modélisation du problème de distribution « MD-FSMVRPTW », sous forme d'un modèle mathématique multi-objectif, en se basant sur la modélisation des problèmes de tournées de véhicules mono-objectif. Nous proposons une approche de résolution pour ce problème basée sur une hybridation entre une métaheuristique très connue et très utilisée dans le domaine de tournée de véhicules « les algorithmes génétiques » et deux algorithmes dédiés clustering « l'heuristique de

voisinage du plus proche voisin et l'algorithme d'apprentissage Kmeans » afin de garantir la notion de regroupement « clustering ».

Dans le cadre des méthodes multi-critères, la présence de différents objectifs conflictuels complique de plus en plus la résolution du problème, car il existe un équilibre tel que l'on ne peut pas améliorer un critère sans endommager au moins un autre. Pour cela, diverses solutions ont été proposées dans la littérature tel que le cas où on transforme le problème multi-objectif en un problème mono-objectif, ou bien en gardant un seul objectif prioritaire, ou bien comme dans notre cas, en gardant le problème avec plusieurs objectifs en se basant sur l'approche Pareto.

Cette thèse est divisée en deux parties. La première partie est consacrée à un état de l'art détaillé sur les problèmes de tournées de véhicules et les techniques de résolution de ses variantes. La deuxième partie s'intéresse au système d'aide à la décision dont le but est d'apporter de l'aide au conducteur pour sélectionner le meilleur chemin possible à prendre lors de son trajet de distribution. Nous présentons aussi les approches et les modèles mathématiques déjà proposés ainsi que l'optimisation et la présentation des résultats obtenus.

Le premier chapitre est réservé à un état de l'art des problèmes de transport d'une manière générale, problème de voyageur de commerce et le problème de tournées des véhicules et ses variantes. Dans le deuxième chapitre, nous présentons les méthodes de résolutions les plus connues et les plus utilisées dans la résolution des différentes variantes du problème de tournées de véhicules.

Le chapitre 3 est consacré à la proposition d'un modèle mathématique de distribution de produits qui a pour bût de détailler les différents objectifs et contraintes, ainsi que l'approche proposée pour la résolution de notre problème de distribution de produits.

Nous avons remarqué que les problèmes d'optimisation combinatoire provenant des cas réels sont, par nature, multi-objectif. Pour cela, nous avons présenté dans le chapitre 4, quelques modèles mathématiques multi-objectif ainsi que les différentes méthodes d'optimisation utilisées pour leurs résolutions. Nous présentons par la suite, le modèle multi-objectif ainsi que la méthode d'optimisation utilisée pour sa résolution.

Le chapitre 5 est dédié à la présentation des différentes étapes d'implémentation des instances de tests. Nous avons présenté en premier lieu ces instances, ensuite, nous avons testé notre approche en utilisant ces instances de tests. Les simulations effectuées et les résultats acquis ont été exposés en dernier lieu en utilisant l'outil GUIMOO.

Cette thèse s'achève par une conclusion générale et quelques perspectives de recherche.

PARTIE I
ETAT DE L'ART

CHAPITRE 1 : LES PROBLÈMES DE TRANSPORTS

1. Introduction

Après plus de cinquante années de recherche, le problème d'élaboration de tournées de véhicules introduit par Dantzig et al en 1959 [2], demeure un des nombreux défis les plus liés à la vie courante. Et pour cela, nous allons examiner dans se présent chapitre la littérature pertinente du problème de tournées de véhicules. Le principe général du problème de base consiste, en optimisant un ou plusieurs critères, à construire un ensemble de tournées, pour un nombre fini de véhicules, commençant et finissant à un dépôt. Dans ces tournées, un client doit être desservi une seule fois par un seul véhicule et la capacité de transport d'un véhicule pour une tournée ne doit pas être dépassée.

Dans la littérature nous trouvons que la problématique liée à l'ordonnancement des transports a été largement étudiée au cours des dernières décennies. Dans cette catégorie de problèmes, on trouve différentes classes de problèmes qui ont beaucoup attiré l'attention des chercheurs, nous allons présenter les plus étudiés dans ce qui suit :

2. Le problème de transport

Le problème de transport (transportation problem) fut formulé pour la première fois par Hitchcock en 1941 [3]. Ce problème consiste à minimiser le coût de transport total d'un plan d'expédition. Le problème est un cas particulier d'un problème de graphe (un graphe orienté biparti avec un coût minimal).

Le problème de transport peut être décrit de la façon suivante. Une quantité de produit uniforme est disponible en certains points appelés origines (par exemple dépôts, ...). D'un autre côté, une quantité de produit est demandée en d'autres points appelés destinations (par exemple clients, points de vente, ...). Il s'agit d'envoyer les produits des origines aux destinations. Nous considérons qu'il existe toujours un chemin pour aller de chaque origine à chaque destination. Le coût de transport d'une unité de produit des origines vers les destinations est connu. L'objectif est de déterminer la distribution de produit qui permette de satisfaire la demande de chaque destination et de minimiser le coût total de transport [4].

Nous supposons un ensemble de « m » origines et « n » destinations. Notons a_i la quantité d'unités de produit disponible à l'origine i , et b_j la quantité d'unités de produit demandé par la destination j . Le coût de transport d'une unité de produit depuis l'origine i vers la destination j est noté c_{ij} . Les variables de décision x_{ij} représentent le nombre d'unités de produit envoyées de l'origine i vers la destination j .

Le problème peut être formulé comme un programme linéaire :

$$\begin{aligned} \text{Min } & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ & \sum_{j=1}^n x_{ij} \leq a_i \quad \forall i = 1 \dots m \\ & \sum_{i=1}^m x_{ij} \geq b_j \quad \forall j = 1 \dots n \end{aligned} \quad \text{Eq. I.1}$$

Où :

c_{ij} : le coût de transport du client i à j ;

x_{ij} : Taille de la marchandise transporté du client i à j ;

a_i : Capacité maximale à transportée ;

b_j : Capacité minimale à envoyée ;

Il existe une solution réalisable pour le problème (Eq. I.1) si la condition :

$$\sum_{j=1}^n b_j \leq \sum_{i=1}^m a_i \text{ est satisfaite.}$$

Les problèmes de transport, appelés aussi problèmes de routage, modélisent des problèmes réels liés au transport de marchandises ou de personnes. Afin d'introduire le problème de routage de véhicules, nous parlerons de deux autres problèmes de transport : le problème du voyageur du commerce et le problème du postier chinois.

2.1. Le problème du voyageur de commerce

Le problème du voyageur de commerce (PVC), mieux connu sous le nom de Traveling Salesman Problem (TSP), a été soulevé pour la toute première fois par Euler (en 1736). Depuis, il a été si largement étudié dans la littérature puisqu'il est la pierre angulaire de tout problème de tournées, il représente aussi un des problèmes les plus étudiés en recherche opérationnelle, est un des problèmes d'optimisation combinatoire les plus connus. Il pose le problème suivant ; Avec un seul véhicule partir d'un dépôt, trouver un chemin de longueur totale minimale qui passe exactement une fois par chaque client et revienne au dépôt, un exemple est donnée par la Fig. I.2.

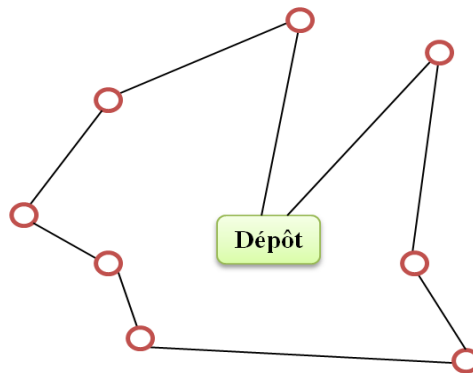


Fig. I.2. Exemple de TSP

Plus formellement, un PVC est modélisé sous forme d'un graphe où les sommets représentent les villes à visiter, et les arêtes les liaisons entre ces villes. La pondération ou le poids associé à chaque arête représente le coût de la liaison entre les deux villes et correspond généralement à la distance qui les sépare. L'objectif est de trouver un cycle hamiltonien.

2.1.1. Formulation mathématique d'un PVC

En tant que problème d'optimisation, le problème du voyageur de commerce étant NP-difficile, le temps pour trouver la solution optimale augmente de façon exponentielle en fonction de la taille du problème. Il se définit mathématiquement comme ceci :

Soit $G = (V, E)$ ou G est un graphe, et $V = \{v_0, v_1, \dots, v_n\}$ ou représente l'ensemble des nœuds (villes). Par ailleurs, $E = \{(v_i, v_j) | i < j; v_i, v_j \in V\}$ représente les arrêtes reliant ces nœuds si les distances sont symétriques et $E = \{(v_i, v_j) | i \neq j; v_i, v_j \in V\}$ représente les arcs lorsqu'elles sont asymétriques. Dans le cas où le problème est symétrique c'est-à-dire où le graphe est non orienté, on parlera d'un cycle et dans le cas contraire on parlera d'un circuit.

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

- c_{ij} : Distance entre la ville i et la ville j ,
- n : Nombre de villes,
- x_{ij} : Variable binaire qui prend la valeur 1 si la ville i est visité immédiatement avant la ville j . Sinon, cette variable prend la valeur 0.

Le problème consiste à minimiser la longueur du cycle Hamiltonien. La fonction objectif Z est donnée par (Eq. I.2) :

$$\min Z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad \text{Eq. I.2}$$

Les contraintes sont les suivantes :

$$\sum_{j=1}^n x_{ij} = 1 \quad i=1, \dots, n \quad \text{Eq. I.3}$$

$$\sum_{i=1}^n x_{ij} = 1 \quad j=1, \dots, n \quad \text{Eq. I.4}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad S \subset V, 2 \leq |S| \leq n - 2 \quad \text{Eq. I.5}$$

$$x_{ij} \in \{0,1\} \quad i=1, \dots, n, j=1, \dots, n. \quad \text{Eq. I.6}$$

L'objectif (Eq. I.2) est de minimiser le coût de la tournée. La contrainte (Eq. I.3) assure qu'on ne sort qu'une seule fois de chacun des points tandis que la contrainte (Eq. I.4) vérifie que l'on entre seulement une fois à chaque point. Ces contraintes permettent donc de visiter tous les points une seule fois. Par contre, si l'on n'ajoute pas la contrainte (Eq. I.5) des sous-tours se formeront, cette dernière élimine donc tous les sous-tours possibles dans la tournée, i.e. la tournée devra obligatoirement débuter et se terminer au dépôt ; S étant le nombre de sommets inclus dans un sous-tour potentiel composé des clients de l'ensemble S . Dans cette contrainte $|S|$ dénote la cardinalité de l'ensemble S . Cette contrainte doit être décrite pour tous les sous-ensembles S de V . Enfin, le respect des variables binaires est assuré par la contrainte (Eq. I.6). Le problème est de type NP-complet, c'est-à-dire qu'il n'existe pas d'algorithme de résolution exacte en un temps "raisonnable". Intuitivement, cela revient à dire qu'il faut tester toutes les solutions réalisables. Or la plupart de ces problèmes sont tellement combinatoires que la simple énumération des solutions est impossible dès que la taille des données augmente. Par exemple, prenons un TSP avec 50 villes, on aura 50! tours possibles.

2.1.2. Le PVC symétrique et asymétrique

Une première particularité de ce problème se situe au niveau des caractéristiques du graphe. Effectivement, les arcs du graphe peuvent avoir un poids inégal selon le sens dans lequel on le parcourt, dans ce cas on dira que la matrice de ce problème est asymétrique.

Dans le cas contraire, où la distance est la même peu importe le sens, alors la matrice du problème sera symétrique. D'après Oberlin [5] Un problème est dit symétrique si le coût du déplacement du client 'i' à 'j' est égale au coût de déplacement du client 'j' à 'i', pour tous les nœuds.

2.2. Le problème du postier chinois

Le nom du problème vient du fait qu'il a été étudié par le mathématicien chinois Meigu Guan (en 1962), et qu'il modélise la tournée d'un facteur devant effectuer le plus efficacement possible sa tournée en passant au moins une fois par chaque rue de son secteur.

Le Problème du Postier Chinois (PPC), fait partie aussi de la classe des problèmes de transport, un réseau de villes est représenté sous forme d'un graphe où les sommets et les arcs correspondent respectivement aux villes et aux liaisons. Mais contrairement au voyageur du commerce, le postier chinois doit effectuer un tour passant par toutes les arêtes au moins une fois, tout en minimisant la distance totale parcourue. Ce problème peut modéliser les tournées de ramassage d'ordures ménagères ou de distribution de courrier [6].

La différence principale entre le PVC et le PPC réside dans l'élément du graphe sur lequel opère l'optimisation : le PVC impose que chaque sommet soit visité et le PPC impose que chaque arête (ou arc) soit visitée.

Plus généralement, les problèmes de routage peuvent être classés en deux catégories : les problèmes qui opèrent sur les villes (Vertex Routing Problems) et les problèmes qui opèrent sur les liaisons (Arc Routing Problems).

2.3. Problème de tournées de véhicules

2.3.1. Introduction

Le Problème de Tournées de Véhicules plus connu sous son nom anglais Vehicle Routing Problem est un problème qui a fait l'objet de nombreuses études. Il a été introduit en 1959 par Dantzig [2]. Il permet de modéliser et résoudre des problèmes de transport opérationnel. Il est l'intersection de deux problèmes difficiles également bien connus :

- **Le problème du voyageur de commerce** : consiste à organiser un circuit pour visiter une et une seule fois un nombre prédéfini de clients de manière optimale, c'est-à-dire généralement en minimisant les coûts [5], représentant généralement la distance de parcours du circuit.
- **Le problème du bin-packing** : où il s'agit de trouver le rangement le plus économique possible d'un ensemble d'articles dans des boîtes [7].

Le but de la résolution d'un problème de tournée de véhicule consiste à concevoir un ensemble de tournées, partant et revenant d'un même dépôt et en passant par un ensemble de points (souvent nommés clients) prédéfinis. Chaque point de passage doit être visité mais chacun d'entre eux ne le sera que par une seule tournée. Le critère d'optimisation

concerne généralement la distance totale parcourue par les véhicules, mais on peut également rencontrer le nombre de véhicules utilisés, le temps total de parcours, l'équilibrage de charge entre les véhicules etc... Le problème de VRP se modélise généralement à l'aide d'un graphe dans lequel les arcs représentent les axes routiers et les nœuds les points de passage des véhicules. Ce problème est connu pour être NP-difficile [8] c'est à dire qu'il n'existe pas d'algorithme en temps polynomial pour le résoudre.

On associe généralement à chaque point de passage une quantité (à collecter ou à livrer) et/ou un temps de service. De plus, les véhicules sont souvent limités en termes de capacité. La capacité d'un véhicule représente la quantité de produit maximum pouvant être contenu dans le véhicule. Ce problème se nomme problème de tournées de véhicules avec contrainte de capacité (Capacited Vehicle Routing Problem ou CVRP) et consiste à livrer (ou collecter) une charge sur les points de passage à l'aide d'une flotte de véhicules dont la capacité est limitée. Ce type de problème est souvent confondu avec le problème de base qui est celui du VRP. En effet, dans les cas pratiques, les tournées correspondent à une livraison (ou une collecte) avec une certaine demande sur chaque point de passage et doivent donc prendre en compte la contrainte de capacité à respecter par chaque véhicule, on rencontre très rarement des problèmes sans contrainte de capacité.

Ainsi les contraintes usuelles de base du VRP ou CVRP sont les suivantes :

- Tous les clients doivent être visités,
- Un client ne sera servi qu'une unique fois,
- Un véhicule qui arrive chez un client doit en repartir,
- La capacité des véhicules doit être respectée.

Sur ce problème de base viennent se greffer d'autres contraintes permettant de prendre en compte des extensions à ce premier problème et donc modéliser différents problèmes d'optimisation. Nous verrons dans la suite de ce chapitre quelques-unes de ces extensions.

2.3.2. Formulation du problème

Une formulation mathématique du problème de CVRP a été proposée par Fisher et al. [9]. Elle se base sur un graphe : $G = (Z, A)$ où :

- $Z = (z_0, z_1, z_2, \dots, z_{n-1})$ est l'ensemble des n sommets (nœuds) du graphe représentant l'ensemble des clients, avec z_0 représentant le dépôt,
- $A = \{(z_i, z_j) | z_i, z_j \in Z; i \neq j\}$ est l'ensemble des arcs orientés représentant le trajet entre deux sommets.

Soient

- $K = \{1, \dots, m\}$: l'ensemble des véhicules disponibles ;
- Q : la capacité maximale des véhicules (tout en supposant que les capacités des véhicules sont égales) ;
- q_i : la capacité à distribuer sur un nœud z_i ;
- d_{ij} : la distance de parcours de l'arc (z_i, z_j) .

Les variables de décision utilisées dans cette formulation sont :

- $y_i^k = 1$ si le nœud z_i est visité par le véhicule k et 0 sinon.
- $x_{ij}^k = 1$ si l'arc (z_i, z_j) est dans la tournée du véhicule k et 0 sinon.

Les contraintes du CVRP s'expriment alors de la façon suivante :

$$\text{Min}(\sum_{i \in Z} \sum_{j \in Z} \sum_{k \in K} x_{ij}^k \cdot d_{ij}) \quad \text{Eq. I.7}$$

$$\sum_{i=1}^n q_i y_i^k \leq Q \quad (k=1, \dots, m) \quad \text{Eq. I.8}$$

$$\sum_{k \in K} y_i^k = 1 \quad (i = 1, \dots, n-1) \quad \text{Eq. I.9}$$

$$\sum_{k=1}^m y_i^k = m \quad (i = 0) \quad \text{Eq. I.10}$$

$$\sum_{i \in Z} x_{ij}^k = y_j^k \quad (j = 0, \dots, n-1); (k = 1, \dots, m) \quad \text{Eq. I.11}$$

$$\sum_{j \in Z} x_{ij}^k = y_i^k \quad (i = 0, \dots, n-1); (k = 1, \dots, m) \quad \text{Eq. I.12}$$

$$\sum_{i,j \in X} x_{ij}^k \geq |X| - 1 \quad (\forall X \subset Z, 2 \leq |X| \leq n - 2); (k = 1, \dots, m) \quad \text{Eq. I.13}$$

$$x_{ij}^k \in \{0,1\} \quad (i,j = 0, \dots, n-1 ; i \neq j); (k=1, \dots, m) \quad \text{Eq. I.14}$$

$$y_i^k \in \{0,1\} \quad (i = 0, \dots, n-1); (k=1, \dots, m) \quad \text{Eq. I.15}$$

Ce modèle mathématique exprime le fait que l'on cherche à déterminer un ensemble de tournées tout en minimisant la distance totale parcourue (Eq. I.7). La contrainte (Eq. I.8) exprime le fait que la capacité des véhicules est limitée. Les contraintes (Eq. I.9 et Eq. I.10) établissent que chacun des clients n'est desservi qu'une seule fois (ie. par un seul véhicule) sauf pour le sommet dépôt par lequel tous les véhicules doivent passer. Les deux contraintes suivantes (Eq. I.11 et Eq. I.12) établissent la cohérence du parcours : effectuer un trajet d'un nœud à un autre à l'aide d'un véhicule donné revient à dire que ce véhicule est utilisé pour collecter les nœuds en question. La contrainte (Eq. I.13) permet d'éliminer des sous-tours c'est-à-dire d'éliminer des solutions ne passant pas par l'ensemble des clients à visiter. La binarité des variables de décision est donnée par les contraintes (Eq. I.14 et Eq. I.15). Une solution d'un VRP est représentée par la figure (Fig. I.3).

D'autres contraintes peuvent venir s'ajouter à ce premier modèle. En effet, la littérature dénombre de nombreuses extensions à ce problème de base comme l'ajout de fenêtre de temps ou l'utilisation de véhicules de capacités différentes. Chacune de ces extensions représente un problème particulier provenant souvent d'un problème pratique. Dans certains cas, il faut cumuler plusieurs extensions pour avoir la modélisation complète du problème considéré et ainsi étudier des problèmes plus complexes puisque prenant en compte un plus grand nombre de contraintes. Par exemple, l'ajout de fenêtres de temps n'est pas rare dans une application, ainsi que l'utilisation de véhicules avec des capacités différentes.

Un autre type d'extension apparaît lorsque les informations associées au problème de tournées évoluent dans le temps.

Dans la suite de ce chapitre, nous allons voir plus en détails certaines extensions du problème de tournées de véhicules, correspondant à des ajouts de contraintes au modèle de base.

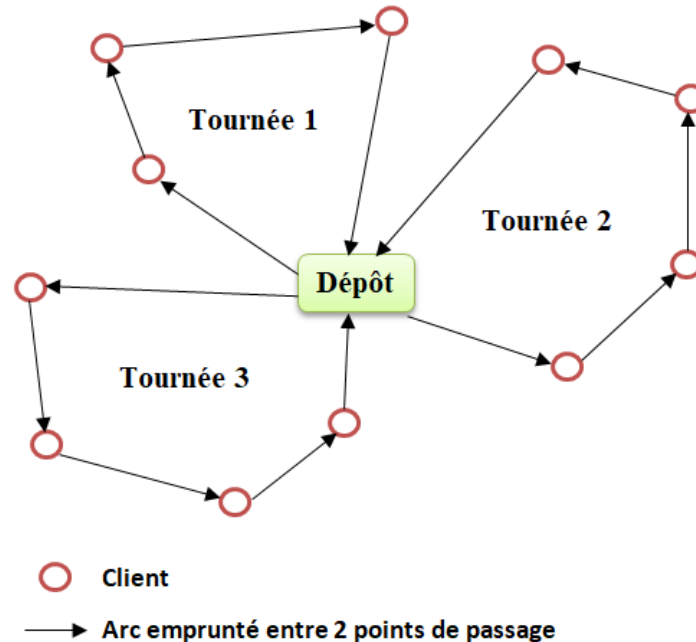


Fig. 1.3 Une solution d'un problème de tournées de véhicules

2.3.3. Les extensions du VRP

Dans la littérature existe différentes extensions au problème de VRP, parmi elles celles qui correspondent à des ajouts de contraintes au modèle de base :

2.3.3.1. VRP avec fenêtres de temps

Plus connu sous Vehicle Routing Problem with Time Windows (VRPTW) [10] est l'une des extensions les plus étudiées. Le problème de base est celui du VRP auquel on ajoute une contrainte sur l'horaire de passage à certains nœuds. Ces contraintes horaires traduisent des limites sur la date d'arrivée au plus tôt et sur la date de départ au plus tard en certains nœuds du réseau. Cette extension représente de nombreux cas concrets de gestion de tournées de véhicules. Dans la littérature, ce problème est traité de deux manières : soit en considérant que les fenêtres temporelles sont strictes et doivent donc être absolument respectées, soit en considérant que les fenêtres temporelles sont souples et peuvent donc ne pas être respectées. Dans ce dernier cas, des « pénalités de retard » sont alors généralement associées à la fonction coût traduisant le non-respect de ces fenêtres.

La formulation d'un VRPTW a été introduite par Solomon [11] et fait intervenir les notations suivantes (en plus de celles utilisées dans le modèle de base de CVRP) :

- $[a_i, b_i]$: la fenêtre de temps associée au nœud z_i avec a_i représentant la date d'arrivée au plus tôt dans le nœud z_i et b_i représentant la date de fin de service au plus tard pour le nœud z_i .
- t_i : le temps de service du nœud z_i ;
- t_{ij} : le temps de transport du nœud z_i au nœud z_j ;
- u_i^k : la variable de décision temporelle représentant l'heure d'arrivée du véhicule k dans le nœud z_i ;
- M : une grande valeur.

Les contraintes temporelles à ajouter au modèle du CVRP sont les suivantes :

$$a_i y_i^k \leq u_i^k \leq b_i y_i^k \quad (i=0, \dots, n-1); (k=1, \dots, m) \quad \text{Eq. I.16}$$

$$u_i^k + t_i + t_{ij} - M(1-x_{ij}^k) \leq u_j^k \quad (i=0, \dots, n-1); (k=1, \dots, m) \quad \text{Eq. I.17}$$

La contrainte (Eq. I.16) permet de vérifier que les nœuds sont servis dans leur fenêtre de temps respective pour un véhicule donné. Lorsqu'un véhicule ne distribue pas un nœud, ces contraintes forcent la variable temporelle à être nulle. La contrainte (Eq. I.17) traduit la succession entre la collecte de deux sommets z_i et z_j : si z_j est collecté après z_i par un même véhicule k alors le début de la collecte de z_j ne peut se faire tant que la collecte de z_i ne s'est pas terminée et tant que le trajet entre ces deux nœuds n'a pas été effectué. Dans ces contraintes, la valeur M représente un grand nombre et permet d'assurer le respect de la contrainte lorsque le sommet z_j ne succède pas au sommet z_i dans la tournée du véhicule k .

La prise en compte d'une limite sur la durée de chaque tournée (ie. Un temps de travail), notée T peut se faire en introduisant les contraintes ci-dessous (Eq. I.18) :

$$\sum_{i=1}^n t_i y_i^k + \sum_{i=1}^n \sum_{j \neq 1, j=1}^n x_{ij}^k t_{ij} \leq T \quad (k=1, \dots, m) \quad \text{Eq. I.18}$$

Dans notre cas, on prend en considération cette extension du problème du VRP avec des fenêtres temporelles souples, qui peuvent donc ne pas être respectées, « avec pénalités de retard ».

2.3.3.2. VRP périodique

Le Periodic Vehicle Routing Problem (PVRP) généralise le VRP en considérant que les points de passage peuvent être desservis plusieurs fois sur un horizon de planification. Pour chaque point de passage, on dispose d'une fréquence de distribution et d'un ensemble de combinaisons possibles de distribution.

Pour résoudre un PVRP [12], il faut déterminer simultanément une combinaison de jour de collecte pour chaque sommet et l'ensemble des tournées pour chacun des jours de l'horizon de planification. Ces deux étapes peuvent être faites simultanément ou l'une après l'autre. Une formulation du problème a été proposée par Russell et al. [13]. Ce type de problème se rencontre fréquemment pour les collectes des déchets.

La modélisation du PVRP utilise les paramètres complémentaires suivants :

- f_i : la fréquence de distribution du sommet z_i ,
- R_i : la combinaison des jours possibles pour le passage sur le sommet z_i .

Plusieurs classifications des problèmes de PVRP ont été proposées par Mourgaya et al. [14]. Dans cet article, les auteurs classent les problèmes de PVRP selon les objectifs, comme la distance de trajet, le temps de trajet, le nombre de véhicules, l'équilibrage des charges, la qualité de service ou encore la régionalisation des trajets (regroupement des points de passage en cluster). Ils proposent de classer également selon les différentes contraintes ajoutées, c'est-à-dire en ce qui concerne (i) la planification des visites (visite unique ou multiple), (ii) le type de demande (identique ou cumulative en fonction du temps ou d'un taux) ou encore (iii) la disponibilité des véhicules. Ils mettent en évidence le fait que ce problème de tournées de véhicules est tactique (choisir les jours de collecte) et opérationnel (construire les tournées). Francis et al. [15] introduisent dans leur étude un niveau de décision stratégique consistant à prendre en compte la fréquence de collecte comme une variable de décision.

Un exemple de PVRPTW (Periodic Vehicle Routing Problem with Time Windows) est donné dans l'article de Cordeau et al. [16]. Dans cet article, les auteurs proposent de nouvelles instances pour traiter des problèmes de tournées de véhicules et notamment celui de PVRPTW en définissant pour chaque client une fréquence de passage et les jours possibles de visite.

Pour résoudre ce type de problème on doit :

- Soit le séparées en deux niveaux (affectation des jours de distribution et création des circuits)
- Soit elles sont intégrées (l'affectation des jours et la création des circuits sont établies simultanément).

Ces deux solutions ne sont pas utiles pour notre cas, puisque nous travaillons sur la livraison à la demande.

2.3.3.3. Problèmes de tournées de véhicules avec livraison et retour chargé

Ce problème appelé Vehicle Routing Problem with Backauls (VRPB) est une extension du VRP incluant deux types de nœuds : des nœuds de livraison et des nœuds de collecte. Dans un VRPB les points de livraison vont recevoir une quantité de produit provenant du dépôt et les nœuds de collecte vont prélever une quantité de produit ramenée au dépôt.

Le point critique pour ce problème est d'organiser en premier les livraisons pour « vider » le contenu du véhicule pour pouvoir collecter les nouveaux produits. La somme des produits à livrer ainsi que la somme des produits à collecter ne doit pas excéder séparément la capacité du véhicule utilisé. La résolution de ce problème consiste à commencer les tournées par les clients à livrer et à les terminer par ceux à collecter. Cette contrainte s'explique par le fait qu'il est très difficile de réarranger les charges vu que les véhicules partent du dépôt chargé au maximum et peuvent vite être en surcharge, c'est ainsi qu'en pratique, les clients à livrer ont une plus forte priorité que ceux à collecter [1].

Dans notre travail on ne va pas rencontrer ce genre de problèmes puisque on s'intéresse à la livraison des produits et non pas à la collecte.

2.3.3.4. Problème général de collecte et de livraison

Le problème général de collecte et de livraison (Pickup and Delivery Problem ou PDP) est à rapprocher du problème de VRPB. La différence réside dans le fait que certains produits vont être collectés et redistribués dans la même tournée sans passer par le dépôt. Pour cela, la capacité du véhicule doit être connue, ainsi que les quantités de produit à redistribuer et les points d'origine et de livraison de chaque produit. Le VRP classique peut être donc considéré comme un PDP dans lequel l'ensemble des origines se situe au dépôt ou bien l'ensemble des livraisons, selon le problème traité.

Ce problème peut présenter certaines contraintes additionnelles, comme par exemple la contrainte LIFO, c'est-à-dire que le produit collecté en dernier doit être redistribué en premier.

Pareil pour cette extension qui est basée sur la notion de collecte et livraison, ça ne nous intéresse pas puisque dans notre cas le produit se trouve au dépôt avant la livraison.

2.3.3.5. Problème de trajet à la demande

Le DARP (Dial a Ride Problem) est une extension du PDP (Pick up and Delivery Problem) où les marchandises sont des personnes à prendre et à déposer [17]. Dans ce type de problème les usagers fixent leur point de ramassage et leur point de dépôt. Les usagers peuvent également spécifier un créneau horaire pour le ramassage et/ou le dépôt. En général ils n'en spécifient qu'un des deux. Ce type de problèmes est de plus en plus étudié en raison de la croissance du transport à la demande mis en place dans certaines agglomérations. Il est souvent étendu à un problème en ligne qui permet de gérer les demandes des usagers en dynamique.

En effet, cette solution est dédiée au problème de transport à la demande où les dépôts de la marchandise (personnes) sont dynamiques, au contraire de notre cas où les dépôts sont fixes et connus à l'avance.

2.3.3.6. Problème de tournées de véhicules avec livraison partagée

Ce problème, autrement appelé Split Delivery Vehicle Routing Problem (SDVRP) permet de satisfaire une demande d'un client par plusieurs tournées. Il a été introduit par Dror [18]. Plusieurs véhicules correspondant à des tournées différentes peuvent passer par un même nœud ; ainsi la demande d'un client peut être fractionnée et un client peut donc être visité par plusieurs véhicules.

Les solutions pour cette extension du VRP peuvent nous être utiles surtout dans le cas où chaque véhicule transporte un produit différent, ou aussi quand on a des produits qui

ne peuvent pas être transportés ensemble « ex. problème de variation de la température du stockage ».

2.3.3.7. Problème de Routage sur un Horizon

Depuis une vingtaine d'années, ce problème (Inventory Routing Problem ou IRP) suscite aussi beaucoup d'intérêt. Il s'agit d'une approche plus globale que celle du PVRP. Le problème consiste à distribuer un produit unique à partir d'un dépôt central vers plusieurs clients sur un horizon de planification de plusieurs périodes. Chaque client consomme le produit suivant un taux de consommation donné et une capacité d'entreposer localement le produit dans un réservoir. La distribution se fait au moyen d'un parc de véhicules identiques. Le problème consiste à déterminer, pour chaque période de l'horizon de planification, les clients à visiter, la quantité à livrer et le trajet à suivre pour chaque camion de façon à minimiser les coûts de distribution.

Dans ce cas tout est planifié préalablement (Clients à visiter, quantité à livrer et le trajet à suivre). Ceci est typiquement adapté au problème de transport de carburant.

2.3.3.8. Problème de Tournées de Véhicules avec Flotte Hétérogène

Le problème de tournées de véhicules avec flotte hétérogène (Heterogeneous fleet Vehicle Routing Problem ou HVRP) est une des extensions les plus étudiées du CVRP, car les flottes sont susceptibles d'être hétérogènes dans la plupart des situations pratiques. Selon Penna et al. [19], même lorsque la flotte acquise est homogène, elle peut devenir hétérogène au moment où des véhicules ayant des caractéristiques différentes sont incorporés. De plus, les coûts d'assurance, de maintenance et d'exploitation peuvent avoir des valeurs différentes en fonction du niveau de dépréciation ou du temps d'utilisation de la flotte.

Nous considérons les cas où la flotte est limitée (Heterogeneous Vehicle Routing Problem "HVRP") ainsi que les cas où la flotte est illimitée (Fleet Size and Mix "FSM"). Plus précisément, nous abordons les variantes suivantes :

- HVRPFV: flotte limitée, avec coûts fixes et variables;
- HVRPV: flotte limitée, avec coûts variables mais sans coûts fixes;
- FMSFV: flotte illimitée, avec coûts fixes et variables;
- FSMF: flotte illimitée, avec coûts fixes mais sans frais variables;
- FMSV: flotte illimitée, avec des coûts variables mais sans coûts fixes.

Depuis son introduction par Golden et al. [20], peu d'auteurs ont proposé des méthodes exactes pour les variantes de FSM. Yaman [21] a suggéré des inégalités valides et a présenté des limites inférieures pour le FSMF. Choi & Tcha [22] ont obtenu des limites inférieures pour toutes les variantes de FSM au moyen d'un algorithme de génération de colonne basé sur un ensemble de formulations. Baldacci et al. [23] ont proposé certaines inégalités ainsi qu'une formulation MIP à deux produits pour la même variante. Le HFVRP est considéré comme beaucoup plus difficile que les problèmes correspondants avec une flotte homogène. À ce moment-là, les instances proposées par Golden et al. [20] avec seulement 20 clients n'ont pas été résolues à l'optimalité. Pessoa et al. [24] ont proposé un algorithme Branch-Cut-and-Price (BCP) sur une formulation étendue capable de résoudre des cas avec plus de 75 clients. Plus récemment, Baldacci & Mingozzi [25] proposent un algorithme basé sur SP qui utilise des procédures de limitation basées sur la relaxation linéaire et la relaxation lagrangienne. Cet algorithme a obtenu des résultats encore meilleurs et pourrait résoudre quelques cas avec 100 clients. Néanmoins, ces algorithmes exacts peuvent prendre beaucoup de temps et ne conviennent pas aux instances plus importantes. D'autre part, il existe une littérature riche sur les méthodes heuristiques pour le HFVRP.

De nombreuses approches basées sur les heuristiques ont été proposées pour les FSM au cours des dernières années. Ochi et al. [26] ont proposé une procédure d'évolution hybride qui combine Scatter Search with Genetic Algorithm (GA) pour résoudre le FSMF. Une implémentation parallèle du même algorithme a été présentée par Ochi et al. [26]. Gendreau et al. [27] a développé un algorithme heuristique qui combine Tabu Search (TS), la mémoire adaptative et une approche GENIUS.

Renaud & Boctor [28] ont proposé une heuristique basée sur le balayage pour le FSMF qui emploie des procédures de VRP traditionnelles de construction et d'amélioration. Lee et al. [29] a proposé un algorithme hybride qui combine TS et SP. Brandão [30] propose un TS déterministe avec différentes procédures pour générer des solutions initiales. Deux algorithmes Memetic ont été développés par Prins [31] pour résoudre toutes les variantes de FSM et le HVRPV. Imran et al. [32] a développé un algorithme VNS (Variable Neighborhood Search) qui utilise des algorithmes classiques pour générer des solutions initiales. Toutes les variantes de FSM ont été considérées par les auteurs. Enfin, Penna et al. [19] ont proposé une heuristique basée sur ILS pour résoudre les mêmes variantes de FSM et HVRP considérées dans le présent travail.

Ce présent travail est basé sur l'extension FSMF vue qu'elle est utile pour notre cas puisque la flotte est composée d'un ensemble de véhicules qui ne sont pas forcément identiques. Le nombre de véhicules dans la flotte est considéré comme illimité et les coûts sont fixes et non-variables.

2.3.3.9. Le Problème de Tournées avec capacité sur les Arcs

Ce type de problème est à rapprocher du VRP puisqu'il adresse également des problèmes de tournées de véhicules. La différence se situe dans le type de tournées effectuées. Pour un VRP la livraison (ou la collecte) s'effectue sur les nœuds, alors que pour le CARP (Capacited Arc Routing Problem) la livraison (ou la collecte) se fait sur les arcs du réseau routier.

L'étude du CARP est plus récente que celle du VRP, les premiers travaux datant des années 1980.

Ce problème consiste, à l'aide d'un ensemble de véhicules partant d'un dépôt unique, à parcourir un ensemble d'arêtes d'un réseau représentant le réseau routier avant de revenir à ce même dépôt. Certaines arêtes du réseau doivent être prises en charge par l'ensemble des véhicules affectés aux tournées. Les demandes des clients sont alors associées aux arêtes. Du point de vue de la modélisation, cela signifie que les clients sont répartis le long des rues où il y a un ensemble de services à réaliser.

Les critères d'optimisation sont les mêmes que ceux utilisés dans le VRP. Comme le VRP, le problème de CARP est connu pour être NP-difficile.

Cette modélisation permet de représenter le réseau routier et les trajets effectifs des différents véhicules de manière plus précise que celle basée sur le VRP. Dans les années 2000, différentes extensions du CARP ont été étudiées. La communauté de chercheurs s'est intéressée à pouvoir prendre en compte des contraintes réelles rencontrées notamment dans le cas de réseaux routiers : voies à double sens ou non, sens interdit, interdiction de tourner etc. Ces travaux ont conduit au ECARP (Extended CARP) ou MCARP (Mixed CARP) [33]. Par ailleurs, une extension prenant en compte l'aspect périodique des tournées sur arcs a également été proposée par W. Ramdane-Cherif [34] dans sa thèse et dans Lacomme et al. [35]. Cette extension, appliquée notamment pour de la collecte de déchets, consiste comme pour le cas du VRP périodique, à prendre en compte un horizon de collecte et des fréquences de collecte associées aux différentes rues pour déterminer une affectation des rues aux jours de collecte tout en minimisant le coût total de l'ensemble des tournées sur l'ensemble des périodes étudiées. Une extension récente s'intéresse à l'intégration de contraintes temporelles limitant les horaires de passage sur certaines rues [36].

Pour ces différentes variantes du CARP, les validations expérimentales traitent des problèmes ayant quelques centaines de sommets et d'arêtes.

Des cas pratiques de CARP se rencontrent pour la collecte de déchets, l'entretien de rues ou de réseaux ferrés, la distribution de courriers, l'inspection de lignes électriques, etc.

Finalement, on note ici que dans notre cas, les contraintes étudiées par cette solution sont prises en charge par le module GPS.

3. Conclusion

Dans ce chapitre, nous avons donné une vue générale sur les différents problèmes du transport où nous nous sommes focalisés sur le problème de base de ce travail, celui des tournées de véhicules (VRP). Nous avons aussi présenté la formulation mathématique de base de ce problème, ainsi que certaines extensions proposées dans la littérature, et qui touchent de près ou de loin à notre problème.

Dans le chapitre suivant, nous allons présenter les différentes méthodes proposées dans la littérature et qui permettent de résoudre le problème de tournées de véhicules « VRP ».

CHAPITRE 2 : RÉSOLUTION DES PROBLÈMES « VRP »

1. Introduction

Parmi les problèmes les plus étudiés dans la littérature, le problème de tournées de véhicules « VRP ». Dans Toth et al. [1], les auteurs le présentent comme étant facile à expliquer mais difficile à résoudre. Le VRP fait partie de la classe des problèmes dit « NP-difficile », parce qu'il est « facile » d'évaluer une solution à un tel problème, mais l'exploration de l'espace de recherche pour la résolution du problème croît de manière exponentielle avec la taille des instances. Il n'existe pas, à ce jour, d'algorithme polynomial pour la résolution de ces problèmes. L'utilisation d'algorithmes permettant de trouver une solution approchée en un temps raisonnable est donc courante.

Dans cette partie, nous allons présenter les problèmes d'optimisation en général, puis les types de méthodes de résolution en se focalisant plus particulièrement sur les méthodes dédiées à la résolution de problèmes de VRP. Pour ce type de problème, on distingue généralement deux types de méthodes de résolution :

- Les méthodes exactes : permettant d'obtenir une solution optimale au problème posé en un temps d'exécution plus au moins important.
- Les méthodes approchées : permettant d'obtenir rapidement une solution au problème posé sans pouvoir affirmer qu'elle est optimale.

2. Problème d'optimisation

Un problème d'optimisation au sens général comprend un ensemble de solutions possibles X , dont la qualité peut être décrite par une fonction objectif f . On cherche alors à trouver la solution x^* possédant la meilleure qualité $f(x^*)$ (en minimisant ou en maximisant $f(x)$ selon l'objectif défini). Un problème d'optimisation peut présenter des contraintes sur x , être dynamique si $f(x)$ change avec le temps ou encore multi-objectif si plusieurs fonctions objectifs doivent être optimisées. Il existe des méthodes déterministes dites « exactes » permettant de résoudre certains problèmes en un temps fini de manière optimale, et des méthodes « heuristiques » permettant une optimisation dite « approchée » [37].

Certains problèmes d'optimisation demeurent cependant hors de portée des méthodes exactes. C'est le cas notamment lorsque le temps de calcul pour une méthode exacte croît de manière exponentielle en fonction de la taille des données. Ainsi, à partir d'une certaine taille des données (dépendant du type de problème et de la méthode de résolution) le temps de calcul nécessaire devient prohibitif et ne permet plus de résoudre de manière exacte le problème posé. Dans de tels cas, le problème d'optimisation est dit « difficile », car aucune méthode exacte n'est capable de le résoudre en un temps « raisonnable ». Pour trouver malgré tout une solution, on devra alors faire appel aux méthodes « heuristiques ». Cet ensemble de méthodes ne garantit en rien de trouver la

meilleure solution, mais a pour avantage de trouver une solution réalisable dans un temps correct. Il existe plusieurs types de méthodes heuristiques et donc plusieurs qualifications possibles de ces méthodes, décrites ci-dessous.

Le problème de VRP a fait l'objet de nombreuses recherches en ce qui concerne les méthodes dédiées pour sa résolution de manière exacte et de manière approchée. Pour le résoudre de manière exacte, les instances traitées ne doivent pas dépasser une centaine de nœuds à traiter [38, 39], par contre en ce qui concernent les méthodes approchées, les instances considérées peuvent être de grande taille (des dizaines de milliers de nœuds).

3. Les méthodes exactes pour le VRP

Ces méthodes permettent d'obtenir la solution exacte à un problème donné, c'est-à-dire qu'elles fournissent la solution optimale d'un problème. Selon Laporte et al. [40], les méthodes exactes peuvent être divisées en 3 grandes catégories :

3.1.Procédure de séparation et d'évaluation

L'algorithme de séparation et d'évaluation, plus connu sous le nom de « Branch and bound » se base sur l'énumération et l'évaluation progressive de différentes solutions possibles. Le principe de cette méthode consiste à, d'une part, construire l'arbre de recherche et d'autre à couper les branches qui ne conduiront pas au résultat optimal [1]. Pour cela, une borne est calculée. Cette borne représente la valeur maximale atteignable si la branche est poursuivie. Une attention particulière doit être portée pour le calcul de la borne, en effet plus cette borne est pertinente, moins les branches sont poursuivies et donc moins il y a de calcul. Au pire des cas, la méthode va énumérer l'ensemble des solutions possibles au problème de départ.

En général, seuls les problèmes de VRP de petite taille peuvent être résolus avec cette méthode (moins de 100 nœuds).

Nous pouvons citer comme exemple l'utilisation de la méthode « K-Trees » [41] qui permet de résoudre des problèmes de 25 à 71 nœuds, et celle utilisée par Naddef et al. [42] qui permet de résoudre des problèmes allant jusqu'à 100 nœuds.

3.2.Procédure de coupes et de séparation

Cette technique est connue également sous le terme anglais Branch and Cut. L'idée générale des méthodes de coupes est de résoudre un programme en nombres entiers comme une séquence de programmes linéaires.

Un algorithme de coupes et branchements (Branch-and-Cut algorithm) est une technique de séparation et évaluation dans laquelle on applique un algorithme de coupes pour calculer la borne de chaque sous-problème. Cette méthode introduite par Padberg et al.[43] pour le problème du voyageur de commerce s'est avérée très efficace, elle est maintenant largement utilisée pour résoudre d'une manière exacte des problèmes d'optimisation combinatoire.

En ce qui concerne le VRP, Naddef et al. [42] et Privé et al. [44] résolvent des problèmes allant jusqu'à 135 nœuds [1], d'un autre côté Ropke, S. [45] et Vidal, T. [46], qui ont pu avoir un résultat allant jusqu'à 150 nœuds. La difficulté pour ce problème est la gestion du nombre de contraintes de sous-tours que donnent les relaxations successives.

3.3.La programmation dynamique

Cette technique est basée sur le principe dit de Bellman « dans une séquence optimale, chaque sous séquence doit aussi être optimale ». La solution optimale d'un problème de taille n s'exprime en fonction de la solution optimale de problèmes de taille inférieure à n . L'idée est d'éviter de calculer plusieurs fois la solution pour un même sous-problème.

Cette technique est limitée puisqu'elle permet de ne résoudre qu'un très petit nombre de problèmes combinatoires. Pour le VRP cela représente des instances de 10 à 20 clients [47].

4. Méthodes heuristiques pour le VRP

Les méthodes heuristiques nous permettent de trouver des solutions « approchées » pour des problèmes d'optimisation. Ces solutions ne sont pas forcément optimales mais peuvent être obtenues dans un temps relativement correct. En général, ces méthodes permettent de réaliser un bon compromis entre le temps de calcul et la qualité de la solution [37].

Nous allons, dans ce qui suit, présenter quelques heuristiques déjà utilisées pour le VRP.

4.1. Heuristiques de résolution

4.1.1. Heuristiques de construction ou heuristiques gloutonnes

Le principe de base de cette catégorie est de construire un tour en partant d'une solution initiale et en étendant itérativement des solutions partielles. Ces heuristiques permettent de générer rapidement une première solution à un problème de tournées de véhicules. Il existe différents types d'heuristiques de ce genre :

4.1.1.1. Les heuristiques avec économies (savings) :

Cette méthode a été introduite par Clarke et al. [48] pour le CVRP et est l'une des plus connues. L'algorithme débute en créant un aller-retour pour chaque nœud i à visiter. A chaque itération une économie s_{ij} est calculée entre deux sommets de la manière suivante : $s_{ij} = d_{i0} + d_{j0} - d_{ij}$ où d_{ij} représente le coût entre le sommet i et le sommet j comme illustré sur la figure (Fig. II.1). Cette économie est calculée entre deux trajets possibles $(0, \dots, i, 0)$ et $(0, j, \dots, 0)$ pouvant être regroupés en un seul $(0, \dots, i, j, \dots, 0)$.

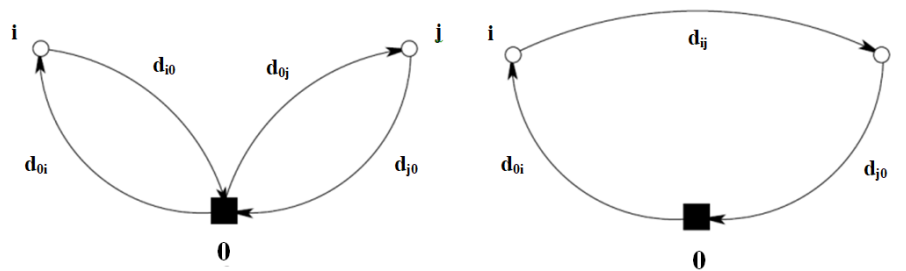


Fig. II.1. Méthode avec économies

On choisit de fusionner les deux trajets dont l'économie est la plus grande. L'application de cet algorithme peut se faire de deux manières : soit de manière séquentielle en augmentant une seule route à la fois, soit de manière parallèle en augmentant toutes les routes si possible (dans l'ordre décroissant des savings). Toth et al. [1] comparent les deux utilisations de cet algorithme et montrent que l'application de la

méthode parallèle donne de meilleurs résultats que la méthode séquentielle en termes de distance totale parcourue.

Cette méthode a fait l'objet de nombreuses adaptations. Certains chercheurs ont proposé des modifications du calcul des économies afin de la généraliser. Gaskell et Yellow [49] ont proposé d'ajouter un paramètre λ au calcul de l'économie tel que, en faisant varier λ , on donne plus ou moins d'importance à la distance entre les deux clients et différentes solutions vont pouvoir être obtenues. D'autres adaptations permettent de calculer les économies de façon à réorganiser la route affectée par le changement, sans préserver l'ordre des deux routes qui sont regroupées. Une autre adaptation a été proposée par Wark et al. [50] et consiste à chaque itération à résoudre un problème de sélection de clients à regrouper. Ce dernier algorithme a montré de très bonnes performances et permet d'atteindre les meilleurs résultats connus pour certaines instances comme répertorié dans Toth et al. [1].

4.1.1.2. Les heuristiques d'insertion :

A l'inverse des méthodes utilisant les économies, qui construisent une route pour chaque client, les méthodes d'insertion intègrent au fur et à mesure les clients qui ne font pas encore partie d'un itinéraire.

Les méthodes d'insertion introduites par Mole et al. [51], permettent de sélectionner un nœud puis de l'insérer à la tournée en cours de réalisation de la meilleure façon possible. Pour cela elle utilise deux critères :

_ Un premier critère $\alpha(i, k, j) = d_{ik} + d_{kj} - \lambda d_{ij}$ permet de savoir où s'insère chaque nœud (non affecté à une route) de manière minimale dans la route en construction, « les paramètres (i, j) varie pour déterminer où s'insère chaque nœud ».

_ Un second critère $\beta(i, k, j) = \mu d_{0k} - \alpha(i, k, j)$ permet de choisir le meilleur nœud à insérer dans la route en construction.

La figure (Fig. II.2) décrit le processus d'insertion.

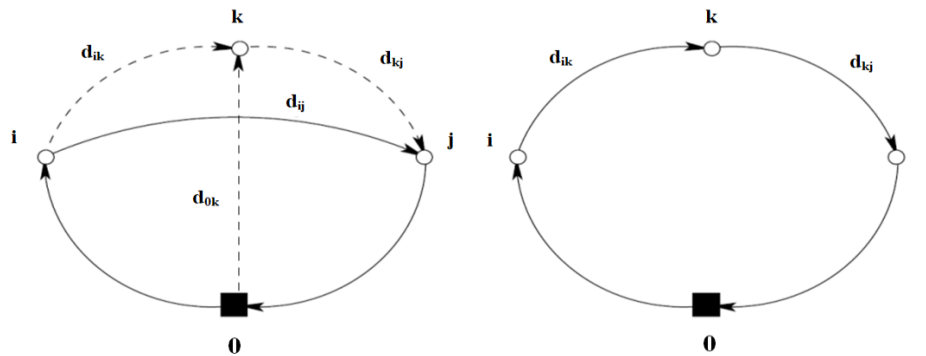


Fig. II.2 Méthode d'insertion de Mole et al.

Le choix des valeurs pour les paramètres λ et μ changent les politiques d'insertion. Par exemple si $\lambda = 1$ et $\mu = 0$ alors l'algorithme va choisir d'insérer le nœud qui donne la plus petite distance d'ajout. Tandis que si $\lambda = 0$ et $\mu = 0$ alors la méthode va choisir le nœud qui va donner la plus petite distance entre deux nœuds voisins. Cette méthode a été appliquée de manière séquentielle, c'est-à-dire qu'elle construit une route après une autre.

Toth et al. [1] propose une méthode d'insertion un peu plus sophistiquée qui se déroule également en deux temps et qui utilise également les deux paramètres λ et μ , mais elle peut être utilisée en séquentielle ou en parallèle. La méthode parallèle suppose d'avoir une première phase qui évalue le nombre de tournées à construire au total. La méthode de Paolo donne de meilleurs résultats que celle de Mole.

Solomon [11] propose différentes méthodes d'insertion pour résoudre des problèmes de VRPTW. Une de ces méthodes utilise également deux critères d'insertion à chaque itération c1 et c2.

Nous allons voir plus en détails cette méthode. Le choix du nouveau nœud à intégrer à une tournée se déroule en deux temps : un premier temps qui consiste à chercher la meilleure insertion dans les différentes tournées pour chaque nœud non encore inséré, puis de choisir celui qui s'insèrera le mieux selon des critères $c1(i, u, j)$ et $c2(i, k, j)$ définis en fonction de la distance et du temps de la manière suivante :

$$c_{11}(i, u, j) = d_{iu} + d_{uj} - \mu d_{ij}, \mu \geq 0 \quad \text{Eq. II.1}$$

$$c_{12}(i, u, j) = b_{ju} - b_j \quad \text{Eq. II.2}$$

$$c_1(i, u, j) = \alpha_1 c_{11}(i, u, j) + \alpha_2 c_{12}(i, u, j), \alpha_1 + \alpha_2 = 1, \alpha_1 \text{ et } \alpha_2 \geq 0 \quad \text{Eq. II.3}$$

$$c_2(i, u, j) = \lambda d_{0u} - c_1(i, u, j), \lambda \geq 0 \quad \text{Eq. II.4}$$

L'équation (Eq. II.1) représente la distance en plus liée à l'insertion d'un nœud u entre les nœuds i et j . Tandis que le critère (Eq. II.2) nous donne l'écart sur le temps de début de service du nœud j si un nœud u est inséré en amont dans le circuit. Ces deux éléments (Eq. II.1 et Eq. II.3) forment le critère 1, qui établit pour chaque nœud restant à insérer le meilleur compromis entre la nouvelle distance et le nouveau temps de service. Ensuite le deuxième critère (Eq. II.4) nous permet de déterminer, toujours pour chaque nœud, l'éloignement du nouveau nœud à insérer par rapport au nœud dépôt. Une fois calculé le critère 2 pour chaque nœud, il suffit de choisir le nœud qui va s'insérer au mieux dans le circuit, c'est-à-dire celui qui donne le maximum du critère 2. Les paramètres de la méthode (α_1 , α_2 , λ et μ) permettent de définir différentes stratégies d'insertion de nœud. Par exemple si $\mu = \alpha_1 = \lambda = 1$ et $\alpha_2 = 0$ cela correspond à l'économie qui est faite pour

servir le client u sur la même route que i et j à la place de le collecter sur une autre tournée.

Cette méthode peut être utilisée de deux manières possibles :

- De manière séquentielle : on construit les tournées les unes après les autres. La difficulté se pose pour choisir le point de départ de la tournée en cours de construction.
- De manière parallèle : on construit les tournées en même temps. La difficulté se pose pour déterminer le nombre de tournées et les points de départ de chaque tournée.

Potvin et al. [52] utilisent cet algorithme pour définir le nombre de véhicules puis utilisent une mesure de regret pour insérer les différents clients.

D'un autre côté, on trouve une littérature riche d'heuristique de construction fortement utilisée dans le domaine de clustering. Nous citons l'algorithme du plus proche voisin (NN heuristic), l'algorithme du k plus proche voisins (KNN heuristic) et l'algorithme de Kmeans. Ces méthodes sont utilisées dans le travail de K. Fukunaga [53] où il a appliqué une recherche locale minimisant les distances entre les objets d'un même cluster, et le travail de Fränti et Kivijärvi [54] dont ils font une recherche locale aléatoire pour le clustering.

4.1.1.3. L'heuristique du plus proche voisin :

L'heuristique du plus proche voisin, Nearest Neighbor Heuristic (NNH) [55] est une heuristique très rapide. Sa complexité est de l'ordre de $O(n^2)$, mais son ratio d'approximation dans le pire des cas est en fonction du nombre de villes ou de nœuds : $NNH=OPT \leq 0,5(\lceil \log_2 n \rceil + 1)$.

Les étapes de cette heuristique sont :

- Choisir aléatoirement un nœud source $s \in S$;
- Le pointeur du nœud courant $c = s$;
- Tant que $\exists x \in S$ non visité Faire
 - Sélectionner v , le voisin le plus proche de c , tel que : $\forall v, v' \in S, p(c, v) \leq p(c, v')$;
 - Aller de c vers v ;
 - $c = v$;
- Aller de c vers s ;

Le défaut de l'heuristique du plus proche voisin c'est qu'elle a un comportement glouton en ne poursuivant que des minima locaux. Une règle informelle utile pour savoir

si un tour obtenu par cette heuristique est bon, dit qu'il suffit d'observer les derniers sauts réalisés par celle-ci, s'ils ont relativement la même taille que ceux de départ, alors le tour est bon, s'ils sont beaucoup plus grands, il existe probablement de meilleurs tours qu'il faut rechercher.

4.1.1.4. L'heuristique du k-plus proche voisin :

L'heuristique K-plus proche voisin(KPP) est efficace pour résoudre des problèmes hautement complexes tels que VRP. Ce sont principalement des problèmes d'optimisation. La technique du voisin le plus proche utilise généralement la distance euclidienne ou la comparabilité cosinus entre les tuples de formation[56, 57]. Elle est efficace pour:

- Fournir une solution rapide et précise.
- Réduire le taux d'erreur causé par la précision des hypothèses.
- En phase de distribution d'un problème sur des ordinateurs parallèles (ce qui est difficile en programmation linéaire), les résultats distribués peuvent être facilement comparés.

Les étapes de cette heuristique sont :

- $V[k]$ tableau contenant les plus proches voisins par ordre décroissant
- Choisir aléatoirement un nœud source $s \in S$;
- Tant que $\exists x \in S$ non visité et $i \leq k$ Faire
 - Sélectionner v , le voisin le plus proche de s et qui n'est pas encore visité, tel que :

$$\forall v, v' \in S, p(s, v) \leq p(s, v') ;$$
 - $V[i]=v$ (marqué comme visité)

Les algorithmes KPP sont non-déterministes, c'est-à-dire qu'ils sont stochastiques dans les décisions, ce qui les rend plus robustes. C'est une technologie clé dans le domaine de la recherche aléatoire pour résoudre des problèmes complexes et peu clairs et qui requièrent beaucoup d'espace pour une solution optimale. C'est une technique importante dans la recherche du choix parfait d'un ensemble de solutions disponibles pour un design particulier. KPP est une excellente technique pour résoudre de gros problèmes avec une complexité de calcul élevée, en particulier la science des incompatibles, où les problèmes doivent améliorer les solutions. KPP peut être appliquée dans les applications suivantes [58, 59]:

- Sélection de la tournée : l'objectif de l'optimisation de la tournée pour le problème VRP et le problème de routage des véhicules pour réduire les coûts et visiter tous les lieux avec des contraintes.

- Les Réseaux de neurones, dans lequel l'opération de sélection de modèle ou de classification basée sur une entrée précédente pour sélectionner un modèle optimal.
- De nombreux types de problèmes d'optimisation : tels que la distribution de paquets, optimisation du trafic, optimisation du puzzle ...

4.1.2. Heuristiques à deux-phases

Pour ces heuristiques, la construction d'une première solution est divisée en deux étapes consistant à regrouper des nœuds au sein de clusters et à calculer un trajet entre les nœuds d'un même cluster. Selon l'ordre d'application de ces étapes, il existe deux types d'heuristiques :

- **Cluster-first, route-second:** cette heuristique commence par partitionner les nœuds du réseau entre les véhicules et construit par la suite pour chaque véhicule la tournée correspondante.

Dans cette catégorie d'heuristiques nous pouvons retrouver, entre autres, deux algorithmes très connus, le « sweep algorithm » développé par Gillett et al. [60] et le « petal algorithm » dont le meilleur exemple est celui développé par Renaud et al. [61]. Le « sweep algorithm » consiste à faire tourner une ligne imaginaire provenant du dépôt, autour de ce dépôt pour former des clusters dans lesquels viennent s'intégrer les nœuds à collecter. Cette ligne balaie les nœuds en faisant un tour complet et chaque nœud balayé par cette ligne sera intégré au cluster en construction pour former par la suite une tournée. Le cluster cesse de grossir lorsque la valeur limite de capacité, de temps ou de distance est atteinte. Un autre cluster est alors entamé. Le « petal algorithm » est une extension de la méthode « sweep » qui consiste à générer des itinéraires appelés « pétales » au sein du cluster en construction et par la suite choisir le meilleur.

- **Route-first, cluster-second :** à l'inverse des heuristiques cluster-first, route-second, cette heuristique construit un « tour géant » (en relaxant la contrainte de capacité) passant par tous les nœuds du réseau puis découpe ensuite ce tour en tournées selon le nombre de véhicules nécessaire [62].

4.1.3. La méthode k-means :

K-means est une méthode d'apprentissage non supervisé développé par MacQueen en 1967 [63], elle est l'une des méthodes les plus utilisées en clustering [64]. La logique formelle de K-means est de diviser les données en K groupes (clusters). Le but de l'analyse de regroupement est basé sur le principe de maximisation de la similarité des groupes à l'intérieur d'eux-mêmes et de minimiser la similarité entre groupes. La similarité des clusters est mesurée par la distance moyenne entre les points centraux du cluster et le point de l'ensemble de données.

L'algorithme a besoin d'un nombre fixe de clusters. Par conséquent, l'algorithme est appelé K-means. K représente le nombre de clusters qui est connu, et c'est un nombre entier positif constant [65].

L'algorithme K-means donne des résultats acceptables avec tous les types de données malgré le fait que certains algorithmes de clustering obtiennent de meilleurs résultats dans certaines séries [66]. Un inconvénient de cet algorithme est de déterminer les valeurs de K . Par conséquent, le nombre K optimal doit être trouvé par un essai et une erreur pour parvenir à un regroupement réussi [63].

Les étapes de fonctionnement d'une méthode de clustering k-means sont :

- Choisir aléatoirement k points différents dans l'espace $k \in S$;
- Tant que $\exists x \in S$ non visité Faire
 - Calculer la distance euclidienne entre chaque point de données de l'ensemble de données et les points moyens ;
 - Affecter un point de données à son point moyen le plus proche ;
 - Recalculer les moyens ;
- Une fois terminé, nous aurons k clusters.

4.1.4. Les méthodes de descente (Hill Climbing)

Les méthodes de descente sont assez connues et doivent leur succès à leur rapidité et leur simplicité, ils partent d'une solution initiale et tente d'améliorer cette première solution en explorant son voisinage. Cette exploration recherche une solution de meilleure qualité parmi les voisins de la solution initiale. Cette nouvelle solution devient alors la solution initiale et une nouvelle exploration de voisinage se déroule. La descente s'arrête quand tous les voisins candidats sont moins bons que la solution courante ; c'est-à-dire lorsqu'un optimum local est atteint. Dans le cadre du VRP, On distingue différents types de descente en fonction de la stratégie de génération de la solution de départ et du parcours du voisinage : une descente avec un voisinage modifiant l'ordre des nœuds au sein d'une seule tournée ou une descente avec un voisinage défini en déplaçant des nœuds entre des tournées différentes.

```

Procédure :  $\varphi$  fonction de coût
Variable locale : S solution courante
Choix d'une solution initiale  $S_0$ ;
Solution courante  $S = S_0$ ;
(a.) Génération des candidats par voisinage ;
Choix du meilleur candidat C ;
Si  $\varphi(C) < \varphi(S)$  Alors
     $S = C$ 
    Aller en (a.)
FinSi
Retourner S

```

Algo1. Méthode de descente générique

4.1.5. Voisinages agissant sur une seule tournée

Ces voisinages sont ceux utilisés pour un problème de voyageur de commerce classique. Ils visent à améliorer la qualité de chaque tournée indépendamment des autres. Ces voisinages se basent, pour la plupart, sur celui développé par Lin [49] la très célèbre méthode λ - opt, qui consiste à enlever λ nœud(s) dans une tournée et à reconnecter ces λ nœuds de toutes les manières possibles dans la tournée étudiée. En se basant sur cette méthode Or [67] développe le voisinage Or - opt qui consiste à déplacer de 1 à 3 nœuds consécutifs dans une même tournée. Nous pouvons également citer la méthode GENIUS développée par Gendreau et al. [68] qui améliore également une tournée à la fois.

4.1.6. Voisinages agissant sur plusieurs tournées

Ces voisinages visent à améliorer simultanément la qualité globale de l'ensemble des tournées. Pour cela, Thompson et al. [68] décrivent un cadre général « b-cyclic, k-transfer » dans lequel selon une permutation circulaire de b tournées, k clients seront intégrés à la tournée suivante selon la permutation considérée. Breedam [69], donne une classification des différents voisinages multi-tournées en 4 catégories : « String-cross », « String-exchange », « String-relocation » et « String-mix » qui concernent toutes les échanges de nœuds entre deux tournées.

Le voisinage « String-cross » consiste à échanger deux suites de nœuds entre deux tournées en échangeant deux arêtes de chaque tournée. Le voisinage « String-exchange » échange au plus k sommets entre deux tournées. Le voisinage « String-relocation » transfère au plus k sommets d'une tournée à une autre (sans échange). Et enfin le voisinage « String-mix » consiste à choisir la meilleure amélioration entre celle donnée par la méthode « String-exchange » et celle donnée par la méthode « String-relocation ». Ces voisinages sont schématisés dans la figure (Fig. II.3).

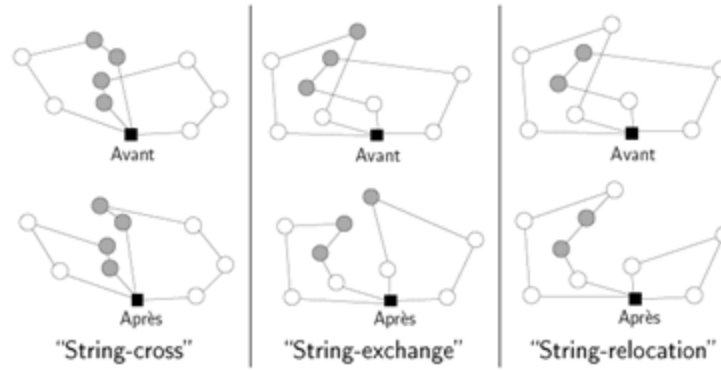


Fig. II.3 Voisinages agissant sur plusieurs tournées

Kindervater et al. [70] définissent des opérations similaires qui améliorent les résultats pour le VRPTW. Dans [71], les auteurs proposent d'utiliser ces voisinages mais en se basant sur différentes méthodes de sélection des nœuds à changer de place dans les tournées.

Les méthodes de descente se caractérisent par l'arrêt de la recherche de solutions lorsque le voisinage défini ne permet plus de trouver une solution de meilleure qualité. La recherche risque donc de rester « bloquée » dans un optimum local. Pour pouvoir dépasser cet optimum local de nombreuses métaheuristiques ont été développées. Dans la figure (Fig. II.4), nous avons illustré les minima locaux et le minimum global d'une fonction $F(s)$ à minimiser.

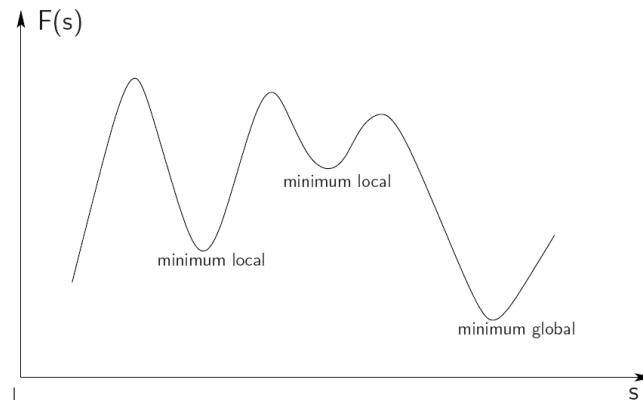


Fig. II.4 Minima locaux et minimum global

4.2. Les méta-heuristiques

Le terme méta(-) sert à construire des néologismes signifiant ce qui dépasse ou bien ce qui englobe.

Certains chercheurs ont tenté de donner une définition à ce terme [72-74], mais aucune n'est reconnue universellement.

Les méta-heuristiques tirent en particulier leur intérêt dans leur capacité à éviter les optima locaux, soit en acceptant une dégradation de la fonction objectif au cours de leur progression, soit en utilisant une population de solutions.

Par le biais des définitions proposées dans la littérature, nous pouvons en tirer certaines propriétés fondamentales communes à toutes ces méthodes :

- Les méta-heuristiques sont des stratégies qui permettent de guider la recherche d'une solution optimale ;
- Le but visé par les méta-heuristiques est d'explorer l'espace de recherche efficacement afin de déterminer des solutions de bonne qualité ;
- Les techniques qui constituent des algorithmes de type méta-heuristiques vont de la simple procédure de recherche locale à des processus d'apprentissage complexes ;
- Les méta-heuristiques sont en général non-déterministes et ne donnent aucune garantie d'optimalité ;
- Les méta-heuristiques peuvent contenir des mécanismes qui permettent d'éviter d'être bloqué dans des régions de l'espace de recherche ;
- Les concepts de base des méta-heuristiques peuvent être décrits de manière abstraite, sans faire appel à un problème spécifique ;
- Les méta-heuristiques peuvent faire référence à des heuristiques qui tiennent compte de la spécificité du problème traité, mais ces heuristiques sont contrôlées par une stratégie de niveau supérieur ;
- Les méta-heuristiques peuvent faire usage de l'expérience accumulée durant la recherche de l'optimum, pour mieux guider la suite du processus de recherche.

Souvent inspirées d'analogies avec la réalité (physique, biologie, éthologie, ...), elles sont généralement conçues au départ pour des problèmes discrets, mais peuvent faire l'objet d'adaptations à des problèmes continus. Les méta-heuristiques, du fait de leur capacité à être utilisées sur un grand nombre de problèmes différents, se prêtent facilement à des extensions. Par exemple :

- L'optimisation multi-objectif (dites aussi multi-critères) [75], où il faut optimiser plusieurs objectifs qui peuvent être contradictoires. La recherche vise alors non pas à trouver un optimum global, mais un ensemble d'optima « au sens de Pareto » formant la « surface de compromis » du problème ;
- L'optimisation dynamique où la fonction objective varie dans le temps. Il faut alors approcher au mieux l'optimum à chaque pas de temps ;
- L'hybridation, qui vise à tirer parti des avantages respectifs de méta-heuristiques différentes en les combinant [76], ...

Du fait du foisonnement de la recherche, un grand nombre de méta-heuristiques existent, nous présentons certaines de ces méta-heuristiques dans la suite de ce chapitre, celles qui sont le plus couramment utilisées pour le VRP. Nous les avons classés par type de méta-heuristiques :

4.2.1. Méta-heuristiques à solution unique

4.2.1.1. Le recuit simulé

La méthode du recuit simulé a été mise au point par Cerny [77]. Elle est inspirée d'un processus utilisé en métallurgie. En faisant fondre un alliage, lors de sa solidification des cristaux se forment à partir des impuretés présentes dans le métal en fusion : ce sont les optima locaux dans un problème d'optimisation combinatoire. La forme et la taille de ces cristaux conditionneront les caractéristiques mécaniques de la pièce solide. Pour modifier les caractéristiques mécaniques de la pièce et diminuer ces « défauts », on effectue souvent une opération que l'on nomme recuit. C'est à dire que l'on chauffe à nouveau la pièce métallique proche du point de fusion, puis on refroidit à nouveau plus lentement. Les anciens cristaux métalliques, sont en partie remplacés par d'autres, des cristaux formés par les impuretés disparaissent, ce qui constitue donc un nouvel arrangement de la matière, et on obtient une structure mieux ordonnée, assimilable à une tendance de se rapprocher de l'optimum global.

Par analogie avec le processus physique, la fonction à minimiser deviendra l'énergie E du système.

On introduit également un paramètre fictif, la température T du système. Cette procédure a pour caractéristique principale d'intégrer des éléments probabilistes pour être guidé, ceci implique une difficulté de paramétrage de ces éléments.

L'algorithme de recuit simulé part d'une première solution (soit calculée, soit obtenue de manière aléatoire), et choisit aléatoirement dans son voisinage une autre solution. Soit celle-ci améliore le critère que l'on cherche à optimiser, on dit alors qu'on a fait baisser l'énergie du système, soit celle-ci le dégrade. Si on accepte une solution améliorant le critère, on tend ainsi à chercher l'optimum dans le voisinage de la solution de départ. L'acceptation d'une solution de moins bonne qualité permet de ne pas s'enfermer dans un optimum local. La difficulté pour ce type d'algorithme est de déterminer les valeurs pour les paramètres en entrée et notamment la température T . Cette température décroît au cours de l'algorithme (et tendre vers 0) et correspond à la probabilité d'accepter une moins bonne solution (plus T sera élevée plus la probabilité d'accepter une moins bonne solution sera forte). Déterminer T et sa loi de décroissance peut se révéler compliqué puisque si T refroidit trop rapidement l'algorithme va se retrouver « bloqué » dans un optimum local et si elle est trop lente, l'algorithme va être trop gourmand en temps de calcul. Contrairement à d'autres méta-heuristiques, il a été montré que le recuit simulé converge asymptotiquement vers l'optimum global [6].

En ce qui concerne les problèmes de VRP et ses dérivés, les premiers recuits simulés ont été développés par Alfa et al. [78], mais ces développements n'ont pas donné de très

bons résultats [1]. D'autres chercheurs, par la suite, ont utilisé cette technique avec succès pour l'appliquer au VRP comme par exemple dans les travaux de (O. Chebbi) [79]. Plus récemment les mécanismes utilisés pour les recuits simulés sont utilisés de manière combinée avec une autre méta-heuristique, comme par exemple dans [80] dans lequel les auteurs utilisent deux méta-heuristiques différentes pour garder les avantages de chacune.

4.2.1.2. La recherche tabou

La recherche tabou a été initiée au début des années 80. Elle a été introduite par Glover [81].

Cette méthode est une méta-heuristique itérative fondée sur l'exploration d'un voisinage. C'est l'une des rares méta-heuristiques qui soit déterministe.

Partant d'une solution initiale, le processus consiste, à chaque itération, à choisir la meilleure solution dans le voisinage de la solution courante, même si elle n'entraîne pas une amélioration de la fonction objectif (afin de ne pas « tomber » dans un optimum local). Pour éviter alors de boucler sur un même ensemble de solutions, la méthode tabou s'appuie sur une mémoire des solutions explorées. Cette mémoire est appelée liste de solutions « interdites » ou tabou (d'où le nom de la méthode). Cette liste tabou peut contenir par exemple certains attributs des dernières solutions visitées et interdit pendant un certain nombre d'itérations le retour vers l'ensemble des solutions contenues dans cette liste. Lorsqu'une nouvelle solution est choisie, la liste tabou est mise à jour en remplaçant les attributs de la plus ancienne par ceux de la nouvelle. La méthode tabou stoppe dès qu'un critère d'arrêt est vérifié. Ce critère peut être le temps d'exécution de la méthode, le nombre d'itérations, un seuil minimal à atteindre pour l'objectif de la solution ,... La meilleure solution faisable de toutes celles qui ont été explorées est bien sûr celle qui sera conservée (ce n'est pas forcément la dernière solution explorée).

En ce qui concerne le VRP, et ses extensions (comme par exemple le VRPTW), il existe de nombreuses applications utilisant une recherche tabou. Dans [82], les auteurs exposent certaines méthodes déjà développées pour le problème de VRPTW. Nous pouvons, par exemple, citer [83] qui utilise la recherche tabou couplée à une « mémoire adaptative » qui a pour but de modifier les différentes solutions de départ à l'aide de combinaisons de tournées extraites de bonne solution déjà trouvées et peuvent mêmes dans certains cas nous donner les meilleures solutions.

La recherche tabou développée par Gendreau et al. [84] permet de trouver des résultats de très bonne qualité voir même égales aux meilleures solutions connues. Cordeau et al. [16] proposent une méthode qui permet de visiter des solutions ne respectant pas les contraintes du problème tout en associant des pénalités à ces violations de contraintes dans la fonction de coût.

Plus récemment, nous pouvons citer Toth [85] et Tarantilis [86] qui proposent des méthodes tabou qui sont très efficaces et qui produisent des résultats de très bonne qualité. Toth et al. [85] prétraitent le problème pour améliorer les résultats. Tandis que Tarantilis et al. [86] utilisent la notion de mémoire adaptative.

En ce qui concerne le DVRP, plusieurs études ont été menées, notamment par Rego et al. [87] et Ichoua et al. [88].

4.2.1.3. Variable Neighborhood Search (VNS)

Cette méta-heuristique introduite récemment par Mladenović et al. [89] est basée sur l'exploration successive de plusieurs voisinages. Pour mettre en œuvre cette méthode, il faut disposer d'un ensemble ordonné de voisinages qui sont explorés successivement.

Cette méthode part d'une solution initiale « s » et elle se déroule ainsi :

- (a) une solution s' est choisie aléatoirement dans un voisinage donné de s ;
- (b) le voisinage de s' est ensuite exploré pour en extraire la meilleure solution s'' ;
- (c) Si s'' n'améliore pas la meilleure solution connue, on passe au voisinage suivant ;

(d) Si en revanche s'' améliore la meilleure solution connue, on la mémorise puis on remplace s par s'' et on retourne à l'étape (a).

A l'inverse des autres méta-heuristiques, cette méthode utilise peu ou pas de paramètres, mais nécessite un ensemble de voisinages avec un ordre entre eux. Elle a déjà prouvé son efficacité pour certain problème de VRP, comme le VRPTW , et même le CARP [90].

4.2.1.4. Large Neighborhood Search (LNS)

Cette méta-heuristique a été présentée par Shaw[91]. Elle se base sur les mêmes principes que ceux de VNS. La différence s'établit lors de l'utilisation des voisinages. Ceux-ci ne sont pas appliqués sur l'ensemble du problème mais sur une partie des variables du problème et peuvent donc être combinés entre eux sur un même problème. Ceci permet de visiter le voisinage d'une solution de manière plus large.

Une étude récente proposée par Pisinger et al. [92] utilise une extension du LNS : Adaptive Large Neighborhood Search (ALNS) qui utilise une méthode de recherche spécifique pour trouver le voisinage, et consiste à détruire une partie de la solution pour ensuite la reconstruire à l'aide de méthodes heuristiques spécifiques. Cette méthode a été testée sur plusieurs types de problèmes types de VRP, et notamment ceux avec fenêtres de temps et donne pour certains problèmes de meilleurs résultats que ceux déjà connus.

4.2.2. Méta-heuristiques à population de solutions

4.2.2.1. Les algorithmes génétiques

Les algorithmes génétiques sont des algorithmes d'optimisation qui s'appuient sur des techniques inspirées de la génétique et de l'évolution naturelle : sélection, croisement, mutation, ...

C'est au début des années 60 que John Holland a commencé à s'intéresser à ce qui allait devenir les algorithmes génétiques. Ses travaux trouvent un premier aboutissement en 1975 [93].

L'ouvrage de Goldberg [94] a également largement contribué à les vulgariser. Les quatre éléments fondamentaux des algorithmes génétiques sont :

- L'évaluation du niveau d'adaptation d'un individu (évaluation de la fonction objective à optimiser),
- La sélection : représentant le choix des individus en fonction de leur niveau d'adaptation,
- Le croisement : correspond au mélange des bagages génétiques entre individus,
- La mutation : traduisant une modification abrupte du bagage génétique d'un individu.

Les algorithmes génétiques servent à simuler le processus d'évolution d'une population. A partir, d'une population de « n » individus (n solutions d'un problème donné), des opérateurs de sélection, croisement et mutation sont appliqués à l'ensemble de ces individus pour en définir des nouveaux [37]. La sélection a pour but de favoriser les meilleurs éléments de la population, le croisement et la mutation assurent une large exploration de l'espace de recherche de départ en diversifiant la population d'individus. De nouveaux individus vont être évalués et vont venir remplacer certains plus anciens ou moins bons, ... Ainsi la population des « n » individus évolue au cours du temps et contient des individus de mieux en mieux adaptés au problème. Elle se dirige donc vers l'optimum.

Les critères d'arrêt de la méthode sont un nombre fixé de générations, une limite de convergence de la population, ou une population qui n'évolue plus suffisamment, etc. Les algorithmes génétiques diffèrent des algorithmes classiques d'optimisation et de recherche essentiellement en quatre points fondamentaux :

- Ils utilisent un codage des éléments de l'espace de recherche et non pas les éléments eux-mêmes,
- Ils recherchent une solution à partir d'une population de solutions et non pas à partir d'une seule solution,
- Ils n'imposent aucune régularité sur la fonction étudiée (continuité, dérivabilité, convexité. . .),
- Ils ne sont pas déterministes, et utilisent des règles de transition probabilistes (comme le recuit simulé).

Les algorithmes génétiques ont fait l'objet de nombreuses études et notamment pour le VRP et ses extensions. Des algorithmes génétiques pour le VRP classique ont été proposés récemment contrairement au VRPTW [95, 96]. C'est en 2013 que (Vaira, G.) [97] proposent un algorithme génétique compétitif pour le problème de CVRP basé sur l'évolution parallèle de deux populations de solutions afin de minimiser la distance totale de parcours. Ils s'inspirent des techniques connues en tournées de véhicules pour faire évoluer les populations. En intensifiant la recherche dans des parties de l'espace de solution qui semblent intéressantes de par l'utilisation d'algorithmes génétiques, ils diversifient la recherche, et en utilisant les techniques connues, ils intensifient la recherche dans certaines parties. Prins [98] propose également un algorithme évolutionnaire qui permet de résoudre efficacement le problème de VRP et des algorithmes pour résoudre le CARP.

Cependant, les recherches déjà effectuées indiquent que la résolution basée méta-heuristiques des problèmes VRP de multiples dépôts, en particulier les algorithmes génétiques, est limitée par rapport à sa contrepartie avec VRP de dépôt unique et ses variantes. Haghani et al. [99] et Salhi et al. [100] ont présenté une formulation pour résoudre le problème dynamique du routage du véhicule avec des temps dépendant du temps de déplacement en utilisant les algorithmes génétiques. Skok et al. [101] ont étudié l'utilisation des algorithmes génétiques pour le MDVRP. Ils se sont concentrés sur l'utilisation des algorithmes basés sur les algorithmes génétiques qui permettent l'acheminement de tous les véhicules. Ghani et al. [102] ont proposé un algorithme génétique traitant des problèmes de routage du véhicule sans horaires et axé sur le coût total du voyage.

En revanche, plusieurs Algorithmes génétiques hybrides "HGA" ont été proposés pour les problèmes de MDVRP. Jeon et al. [103] ont utilisé trois processus heuristiques différents pour améliorer la génération d'une solution initiale et un taux de mutation de flotter pour échapper à la solution locale afin de trouver la meilleure solution. Pour résoudre le MDVRP efficacement, deux algorithmes génétiques hybrides ont été développés par Ho et al. [104]. Chen et Al. [105] ont développé un algorithme génétique hybride (GA) avec un recuit simulé pour résoudre le MDVRP. Dans Lau et al. [106], on a

utilisé une technique de recherche stochastique appelée algorithme génétique guidé par la logique floue (FLGA) pour résoudre le problème. Vidal et al. [46] ont proposé un algorithme génétique hybride pour le Multi-Depot VRP périodique qui combine la largeur d'exploration de la recherche évolutive basée sur la population, les capacités d'amélioration agressives des méta-heuristiques basées sur la notion de voisinage, et des programmes avancés de gestion de la diversité de la population.

Récemment, l'aspect dynamique du VRP a également fait l'objet de diverses études telles que [107] et [99], où des problèmes de temps réel ont été étudiés.

Les algorithmes génétiques sont avérés efficaces pour acquérir une solution optimale ou presque optimale dans le but de résoudre les problèmes d'optimisation. Cependant, ils comportent deux problèmes majeurs : (i) vitesse de recherche lente et (ii) convergence prématurée. Pour cela, nous développons dans ce présent travail une nouvelle solution pour le MD-VRPTW basée sur les algorithmes génétiques avec un meilleur ajustement de ses paramètres.

4.2.2.2. Les colonies de fourmis

C'est une méta-heuristique des plus récentes. Les colonies de fourmis proviennent des travaux de Colomi [108], elles ont été créées à partir de la constatation suivante : les fourmis résolvent naturellement des problèmes souvent complexes. Les biologistes ont étudié comment les fourmis arrivent collectivement à résoudre des problèmes trop complexes pour un seul individu, notamment les problèmes de choix lors de la recherche de sources de nourriture.

Leur technique se base sur la pose de marqueurs chimiques, les phéromones, déposées sur les trajets parcourus et qui pourront être suivis par leurs congénères. Les fourmis utilisent les pistes de phéromones pour marquer leur trajet, par exemple entre le nid et une source de nourriture.

Une colonie est ainsi capable de choisir (sous certaines conditions) le plus court chemin vers une source à exploiter, et ceci sans que les individus aient une vision globale du trajet. En effet, les fourmis qui reviennent le plus rapidement au nid (celles qui auront emprunté le chemin le plus court) auront déposé sur leur trajet deux fois plus de phéromones. Les fourmis suivantes ayant tendance à emprunter les chemins les plus marqués, une solution va progressivement se dessiner.

Les premiers algorithmes de colonies de fourmis développés [109] concernaient le problème du Traveling Salesman Problem. Cette méthode a été, par la suite, testée et adaptée à nombreux types de problèmes. Mais les auteurs ayant remarqué que les colonies de fourmis n'explorent pas assez l'espace de recherche, ils les ont souvent couplées avec des méthodes de descente.

Malgré que cette approche soit assez récente, elle a déjà été avec succès pour certaines applications, et notamment pour des problèmes de VRP ou certaines de ces extensions [110]. Avec une heuristique 2-opt ajoutée pour améliorer les tournées individuellement obtenues par les colonies, cette approche donne des résultats légèrement inférieurs à ceux donnés par la recherche tabou.

Plus récemment, on peut noter l'algorithme développé par Tan et al. [111]. L'aspect DVRP a été étudié par Montemanni et al. [112].

Pour conclure, nous avons conçu un schéma récapitulatif (Fig. II.5) des différentes méthodes déjà utilisées dans la littérature pour résoudre le problème de tournées de véhicules (VRP).

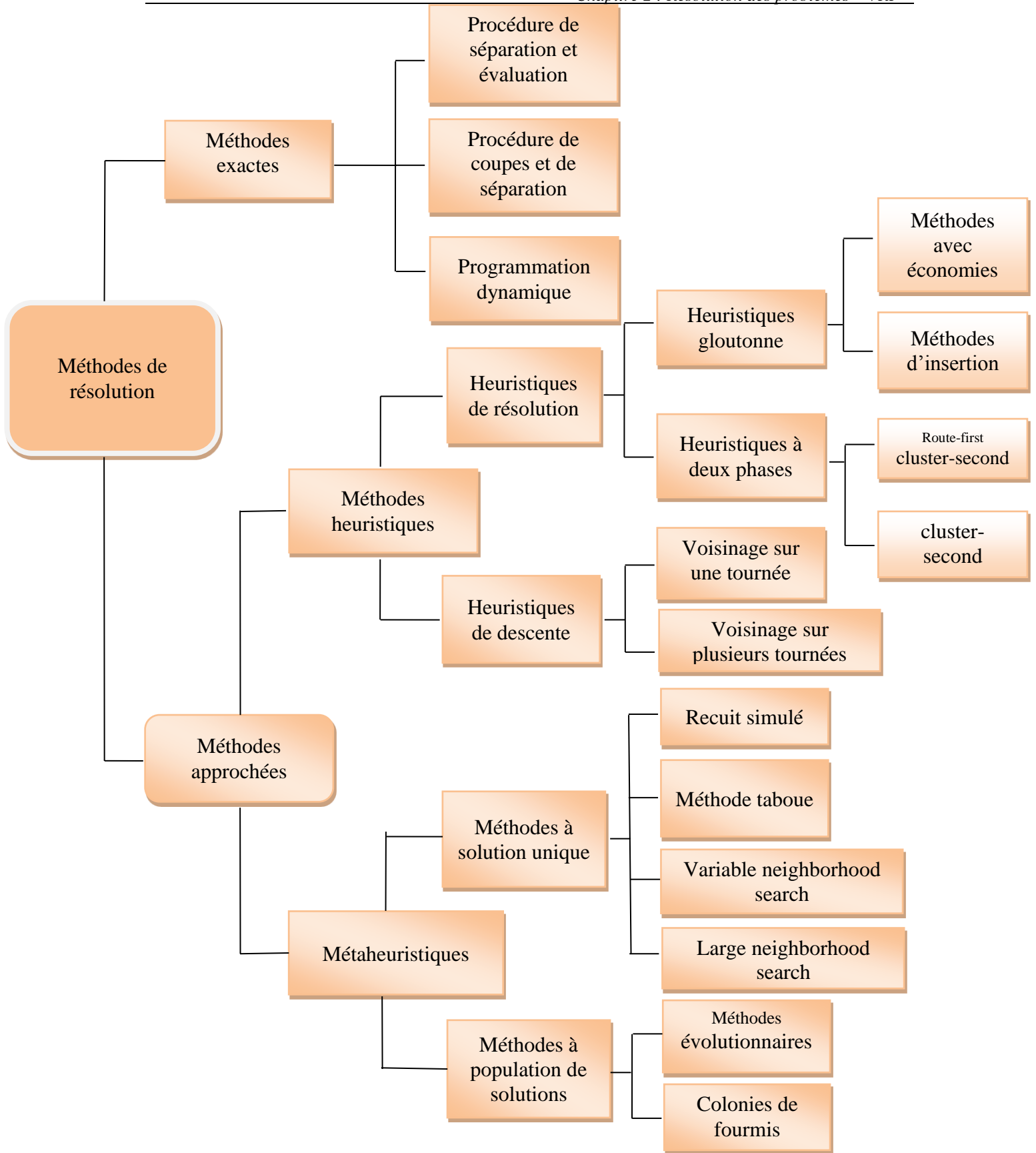


Fig. II.5 Récapitulatif des méthodes de résolution

5. Conclusion

Le problème de tournées de véhicules (VRP) fait partie des problèmes d'optimisation NP-difficile et utilise donc des méthodes de résolution appropriées. Ces méthodes sont divisées en deux grandes catégories, les méthodes exactes qui permettent d'obtenir la solution optimale et les méthodes approchées qui ne garantissent en rien de trouver la solution optimale mais permettent de résoudre en un temps raisonnable tous les types de problèmes. Les méthodes approchées sont réparties entre les méthodes de descente qui ont l'avantage d'être faciles à mettre en œuvre mais qui ont l'inconvénient de s'arrêter sur des optimaux-locaux, et les méta-heuristiques qui sont plus difficiles à mettre en œuvre mais qui permettent de dépasser en partie cette barrière des optimaux-locaux. Les méthodes exactes ne sont pas adaptées aux problèmes de grande taille en raison de temps de calcul qu'elles nécessitent. Les méthodes approchées par contre sont adaptées à tous les types de problèmes.

Parmi les méthodes approchées, les méthodes heuristiques utilisent des mécanismes relativement simples. Les méta-heuristiques donnent en général des solutions de meilleure qualité mais en contrepartie sont plus difficiles à implémenter notamment en raison de leur paramétrage qui peut être compliqué et prend du temps.

En guise de conclusion, différentes méthodes (exactes ou approchées) ont déjà été utilisées dans le but de proposer une meilleure solution pour le VRP et ses variantes. Nous pouvons constater qu'en pratique, les algorithmes génétiques sont révélés très efficaces, ce qui explique pourquoi nous avons opté pour leur utilisation dans une hybridation avec d'autres méthodes de voisinages pour une résolution efficace du problème MD-VRPTW.

PARTIE II
CONTRIBUTIONS ET
VALIDATION

CHAPITRE 3 :
MODÉLISATION DU
PROCESSUS DE
DISTRIBUTION DE
PRODUITS

1. Introduction

La satisfaction des clients se fait par la livraison quotidienne des produits demandés et dans leurs délais, mais la distribution géographique de ces clients rend la tâche de plus en plus difficile à planifier. Cette dernière peut être réalisée à l'aide d'une flotte de véhicules dotés de capacités limitées. Comme dans notre cas où on se trouve face à un cas réel, un problème de tournées de véhicules avec capacité (Capacited Vehicle Routing Problem CVRP).

Avant d'entamer la modélisation de la distribution au niveau des entreprises algérienne, il est préférable de commencer, par une description du réseau routier en Algérie, qui constitue une partie importante du processus de transport de marchandises. Il constitue ainsi un support fondamental au développement économique du pays et des entreprises. La demande de transport s'exprime en règle générale par les déplacements des marchandises. L'offre de transport est représentée d'une part par le parc de véhicules, et d'autre part par le réseau routier. Le trafic est ainsi l'expression de la confrontation entre l'offre et la demande de transport.

Une partie de ce chapitre sera consacrée à la proposition d'une modélisation mathématique multi-objectif, de la distribution des produits. Mais bien avant, nous allons commencer par une définition de la gestion du parc d'automobile des entreprises en Algérie. Pour cela, nous allons essayer de le découper en plusieurs facteurs tel que : le réseau routier, l'espace géographique, le véhicule, la marchandise, ...

2. Le réseau routier en Algérie « extraction des informations par le module GPS »

Le rôle de la route est capital face à l'étendue du pays 2.4 millions de Km², l'infrastructure routière avec une longueur totale de 112 milles kilomètres et plus de 3 milles ouvrages d'art. Le réseau routier en Algérie est le plus important en Maghreb et en Afrique, d'une longueur de 108 302 km, répartie sur 76 028 km de routes nationales et 32 274 km de routes secondaires. Le réseau routier est en plein développement grâce au programme de modernisation des transports routier et ferroviaire (22 milliards de dollars d'investissement public) qui prévoit la réalisation de l'autoroute est-ouest de (1 216 km), l'autoroute des hauts plateaux de (1 330 km), et la réalisation de 19 000 km de route, ainsi la finition de la route transsaharienne (nord-sud)[37, 113].

Quelques chiffres clés de l'autoroute Est-Ouest :

- Longueur: 1 216 km,
- Début des travaux 2005, achèvement des travaux 2012,
- Coût des travaux : 1130 milliards de dinars (11,3 milliards d'euros),
- Composée de : 90 viaducs, 6 tunnels, 524 ponts,
- Péage : Autoroute à péage conçue selon les standards internationaux,
Projet de l'autoroute transsaharienne nord-sud
- Reliera l'Algérie au sept pays africains
- Achèvement en 2019
Projet d'autoroute des hauts plateaux.
- Longueur : 1 330 km
- Début des travaux 2009, achèvement en 2020
- Coût du projet : 11 milliards de dollars
Route transsaharienne (Nord-Sud) :
- L'Algérie est traversée du Nord au Sud par la route transsaharienne, qui est maintenant goudronnée sur sa quasi-totalité et est même une autoroute sur le début de sa partie Nord. Cette route est poussée en avant par le gouvernement pour accroître le commerce entre les six pays traversés par la route transsaharienne (Algérie, Mali, Niger, Nigeria, Tchad et Tunisie).
Rocades et voies express
- Raccordement des autoroutes (est-ouest, hautes-plateaux, transsaharienne nord-sud, Routes du littoral...)
- Début des travaux 2009, achèvement en 2019.

3. Le parc automobile en Algérie

Le nombre de véhicules enregistrés est passé de 86 milles en 1962 à plus de 5,6 millions de véhicules en circulation en 2016, ce qui représente une augmentation annuelle moyenne continue de 10 % de tous les types d'automobiles à moteur. La population, dans la même période, est passée de 9 millions à 40,4 millions d'habitants. Durant les années 2000 (entre 2000 et 2016) le parc a connu une augmentation sans précédent avec plus de 4,2 millions de véhicule toutes catégories confondues.

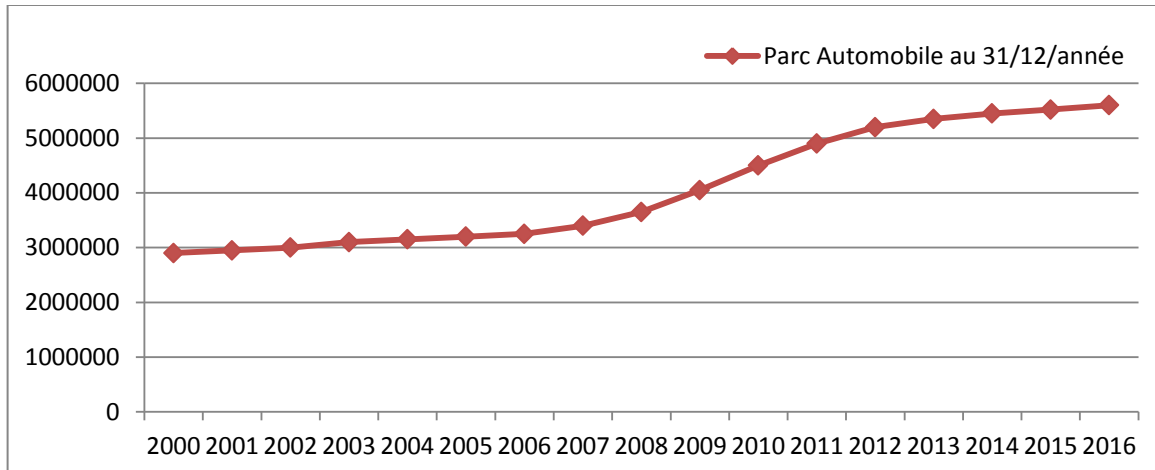


Fig. III.1 Augmentation du parc automobile en Algérie
 « <http://www.ons.dz/-Parc-Automobile-.html> »

3.1. Gestion du parc d'automobiles

Les responsabilités du gestionnaire du parc automobile sont essentiellement administratives, la réception des marchandises à transporter, la planification du transport, l'émission des ordres de missions, la gestion des heures de travail (pointage des chauffeurs), l'émissions des notes de frais pour le chauffeur, l'émissions des documents de transport, le contrôle, le suivi et l'actualisation des documents des véhicules et des documents des chauffeurs, le suivi du contrôle technique des véhicules, la gestion des entrées/sorties des véhicules, le contrôle des processus de gestion informatique de l'information. La gestion des stocks de consommables et de pièces de rechange, la gestion des ressources humaines (horaires de travail/repos), la gestion de la maintenance, la planification des livraisons, la gestion des transports (traçabilité, documents de transport) [37].

La gestion du parc automobile est pratiquée dans les entreprises de location automobile, les entreprises industrielles et commerciales. Pour ces deux dernières, elles peuvent selon les cas disposer d'un parc intérieur ou d'un parc extérieur.

4. L'espace géographique

L'espace géographique influence sur le choix des moyens, mais aussi l'organisation de l'opération de transport. Pour chaque opération de transport, il est essentiel de définir le lieu de départ, le lieu d'arrivée et les principaux points d'escales qui ensembles constituent l'itinéraire. L'organisateur de l'opération de transport peut ainsi recenser les lieux de prélèvements et de livraisons successifs des marchandises tout au long de son itinéraire et déterminer le chemin optimal. Dans notre cas en Algérie, sur une superficie de 2 381 741 Km² et un réseau de transport très important et des distances très importantes

entre les clients ; on trouve aussi des régions qui ont des itinéraires difficiles à parcourir avec des vitesses uniformes, c'est-à-dire parcourir deux routes de même distance, avec une même durée de parcours et une même quantité de consommation de carburant.

L'espace géographique enfin permet de prendre aussi en considération l'état des routes, le relief, le climat, la population qui influencent chacun à sa manière le choix du véhicule et le choix des horaires dans le planning.

5. La marchandise

La marchandise influence le choix des moyens de transport. Il est important d'avoir au préalable une information juste sur le type d'emballage ou le conditionnement des Unités Logistiques (UL). Informations qui plus tard servent à renseigner les documents de transport, permettent d'anticiper sur la nature des moyens matériels à mobiliser au lieu du chargement et du déchargement des marchandises, respect du dimensionnement du véhicule, optimisation du chargement, respect des contraintes liées aux moyens de manutention à mobiliser, respect des conditions de transport (emballage, protection, ...). On peut définir la marchandise tout ce qui fait l'objet d'un trafic autre que le trafic "passagers".

6. La consommation du carburant en Algérie

Le trafic routier présente une dépendance vis-à-vis des produits pétroliers (essences, gas-oil). Le gas-oil est généralement utilisé comme carburant pour les moteurs des véhicules du transport de la marchandise et dans l'industriel. La tendance au niveau mondial et surtout des pays développés, est de limiter la consommation de certains produits dans le cadre de la protection de l'environnement.

Le gas-oil est un produit pétrolier énergétique très polluant. L'utilisation du gas-oil engendre des émissions de : Oxyde d'azote (NOX), Oxyde de carbone (CO2).

Le transport par route assure le transport des marchandises et le transport des personnes. On constate une forte augmentation de la consommation du gas-oil ces dernières années et même des pénuries. Suite à l'augmentation des importations des véhicules, augmentation de déplacement des personnes du fait d'une offre toujours plus importante en services de transports voyageurs ; accroissement des opérateurs activant dans le domaine du transport routier de la marchandise.

Le secteur de transport représente la part la plus importante (49%), suivi par le secteur résidentiel et tertiaire (23%), l'agriculture (17%) et l'industrie et BTP 11%. Voir la figure suivante :

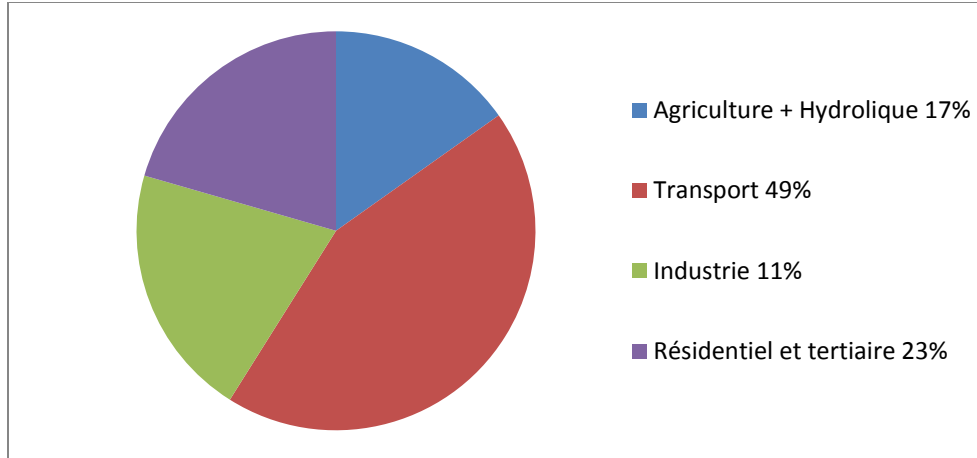


Fig. III.2 Consommation du gas-oil / Secteur d'activités -2016- (<http://www.memalgeria.org>, <http://www.mtp.gov.dz>).

7. Formulation mathématique du MD-VRPTW « Comment résoudre »

7.1.Introduction

Le problème de tournées de véhicules est toujours modélisé, sous forme d'un programme mathématique sous contraintes. Dans cette section, nous exposons une formulation mathématique de la problématique qui permet une meilleure compréhension du problème et de ses contraintes. La taille de ce programme linéaire mixte augmente très rapidement lorsque l'on considère plusieurs produits et plusieurs périodes. De plus, certaines variables de décision sont bivalentes ce qui rend l'obtention de solutions optimales ou même simplement réalisables difficile.

7.2.Modélisation

À partir de l'étude comparative des différentes extensions du VRP et des objectifs que nous voulons atteindre, nous avons pu extraire une formulation mathématique pour notre problème qui permet une meilleure compréhension du problème et de ses contraintes.

Dans notre proposition, nous supposons que le nombre des clients, l'emplacement et les demandes individuelles des clients, ainsi que le nombre et l'emplacement de tous les dépôts potentiels sont connus. Le type et la taille des véhicules sont également fournis. Chaque dépôt est suffisamment grand pour stocker tous les produits commandés par les clients. Chaque véhicule démarre et termine son trajet au même dépôt. L'emplacement et la demande de chaque client sont également connus à l'avance. Chaque client est visité par un véhicule exactement une fois. Les clients sont servis dans leur fenêtre de temps respective et chaque véhicule doit suivre une chronologie de distribution pendant son voyage. Fig. III.3 montre un exemple de MD-VRPTW avec 2 dépôts et 10 clients. Étant donné qu'il existe des dépôts supplémentaires pour stocker les produits, les décideurs

doivent déterminer les dépôts par lesquels les clients sont servis [104]. Les étapes de prise de décision sont classées en regroupement, routage, planification et optimisation, comme le montre la figure (Fig. III.4).

En phase de regroupement, les clients sont regroupés en fonction de la distance entre clients et dépôts. Dans l'exemple, les clients 1,5,9,4,8 sont affectés au dépôt A, tandis que les clients 7,10,3,6,2 sont affectés au dépôt B.

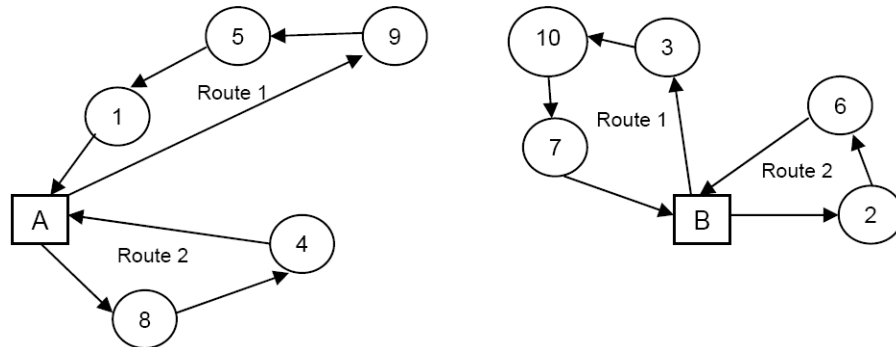


Fig. III.3 Exemple de MD-VRPTW avec 2 dépôts et 10 clients

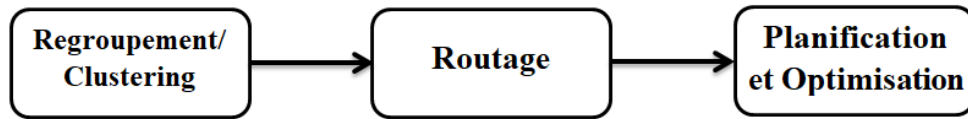


Fig. III.4 Les étapes de prise de décision dans MD-VRPTW

Les contraintes qui doivent être respectées dans notre proposition sont les suivantes :

- Un véhicule qui arrive chez un client doit en repartir.
- La capacité des véhicules doit être respectée.
- Les véhicules ne sont pas forcément identiques (capacité hétérogènes).
- Le nombre de véhicules dans la flotte est illimité.
- Le nombre de dépôts ≥ 1 .
- Les clients sont servis dans leurs fenêtres de temps respectifs.
- Chaque véhicule doit suivre une chronologie de distribution au cours de son trajet.
- Une tournée est limitée par un temps T_k , « Un conducteur, ne peut conduire de manière continue, plus d'un certain nombre d'heures T_k , produits fragiles, ».

7.3. La Formulation du problème

Compte tenu d'un système logistique, on suppose que le nombre de clients, la localisation, les besoins individuels des clients et le nombre et l'emplacement de tous les dépôts potentiels sont connus. Le type et la taille des véhicules sont également fournis.

Maintenant, nous utilisons le modèle MD-VRPTW adopté par Renaud et al. (1996) [114]. Soit $G = (Z, A)$ un graphe direct où $Z = \{z_1, \dots, z_n\}$ représentant l'ensemble des clients, $P = \{z_{n+1}, \dots, z_{n+p}\}$ l'ensemble des dépôts, et $K = \{1, \dots, m\}$: l'ensemble des véhicules disponibles.

L'ensemble d'arcs « A » désigne toutes les connexions possibles entre les nœuds (y compris les nœuds dénotant les dépôts). Nous définissons une matrice de coûts $C = (c_{ij}^k)$ sur « A » correspondant aux temps de déplacement (c_{ij}^k représente le coût de l'arc (i, j) pris par le véhicule k). Les notations utilisées et le modèle mathématique sont les suivants :

7.3.1. Les paramètres :

Pour définir un problème de tournées de véhicules, nous devons tous d'abord bien définir les différentes contraintes du problème, les variables nécessaires, le réseau, la demande, la flotte et surtout les objectifs.

Soient :

- Q_k : La capacité maximale du véhicule K .
- q_i : La capacité à distribuer sur un nœud z_i .
- d_{ij} : Distance euclidienne entre les deux clients i et j .
- d_i : La demande du client i .
- $[a_i, b_i]$: La fenêtre de temps associée au nœud z_i avec : a_i représente la date d'arrivée au plus tôt dans le nœud z_i et b_i représente la date de fin de service au plus tard pour le nœud z_i .
- T_k : Le temps maximal de la tournée du véhicule k .
- t_i : Le temps de service du nœud z_i .
- t_{ij} : Le temps de transport du nœud z_i au nœud z_j .
- u_i^k : La variable de décision temporelle représentant l'heure d'arrivée du véhicule k dans le nœud z_i .
- N : Nombre de véhicules.
- M : une grande valeur.

7.3.2. Les variables de décision

Pour modéliser mathématiquement un problème réel, il est essentiel de définir les variables de décision. Dans notre cas nous avons utilisé quelques variables binaires :

- $y_i^k = \begin{cases} 1 & \text{si le noeud } z_i \text{ est visité par le véhicule } k \\ 0 & \text{sinon} \end{cases}$
- $x_{ij}^k = \begin{cases} 1 & \text{si } i \text{ précède immédiatement } j \text{ dans la tournée du véhicule } k \\ 0 & \text{sinon} \end{cases}$
- $Z_{ij} = \begin{cases} 1 & \text{si le client } i \text{ est attribué au dépôt } j \\ 0 & \text{sinon} \end{cases}$
- S_{lk} = Variable auxiliaire pour les contraintes d'élimination de sous-tour dans la tournée du véhicule k .

7.3.3. Le réseau

Notre réseau est représenté par le réseau routier en Algérie, sur lequel se situe, les différents dépôts, et qui est constitué d'un ensemble de sommets notés Z , représentant les clients à visiter et à servir, ainsi que les arcs qui permettent de représenter les liaisons (routes) existantes entre les différents clients noté A .

7.3.4. La demande

Le problème traité est un problème de distribution, on parle de la demande de chaque client. Chaque demande est associée à un nœud. Elle est constituée d'un ou plusieurs produits. On s'intéresse à la capacité totale de la demande.

Dans notre cas, chaque client qui a déjà effectué une demande, doit être servi par exactement un véhicule (une tournée) (Eq. III.1).

$$\sum_{k \in K} y_i^k = 1 \quad (i \in Z) \quad \text{Eq. III.1}$$

7.3.5. La flotte

Les tournées correspondent souvent aux chemins que doit suivre une flotte de véhicules pour visiter les nœuds et les arcs du réseau. La première caractéristique de la flotte est le nombre de véhicules m . Toutefois, même s'il n'y a qu'un seul véhicule, cela ne signifie pas que la solution du problème est une tournée unique. Dans le problème de distribution par exemple, il est possible pour un seul véhicule de faire plusieurs tournées si l'ensemble de la demande est très important.

7.3.6. Dépôt associé

Un véhicule doit commencer et terminer sa tournée en un nœud précis, appelé dépôt. Dans ce présent travail plusieurs dépôts sont disponibles. En effet, chaque véhicule doit visiter au-moins un dépôt (Eq. III.2) :

$$\sum_{k \in K} \sum_{i \in (Z \cup P)} x_{ij}^k \geq 1 \quad (j \in Z) \quad \text{Eq. III.2}$$

7.3.7. Capacité

Les véhicules ont une capacité limitée maximale Q_k , qui dépend du type du véhicule k , exprimée en termes de poids qui ne doit pas être dépassé. Cette contrainte est donnée par (Eq. III.3), elle assure le respect de la capacité des différents véhicules (hétérogènes). Pour chaque tournée, la somme des demandes ne doit pas dépasser la capacité du véhicule affecté à cette tournée.

$$\sum_{i \in Z} d_i \sum_{j \in (Z \cup P)} x_{ij}^k \leq Q_k \quad (k \in K) \quad \text{Eq. III.3}$$

7.3.8. Limitations

En plus de la capacité limitée des véhicules, différentes limitations peuvent être définies, telle que la distance maximale d'une tournée que le véhicule ne doit pas dépasser. Le temps maximal d'une tournée (Eq. III.4) ne doit pas dépasser un certain nombre d'heures (exp. un conducteur, ne peut pas conduire d'une manière continue plus d'un certain nombre d'heures par jour).

$$\sum_{i \in Z} \sum_{j \in Z} (t_{ij} x_{ij}^k + t_{ji} y_{ji}^k) \leq T_k \quad (k \in K) \quad \text{Eq. III.4}$$

7.3.9. Les objectifs

Le réseau de distribution est important, ce qui implique que les distances sont aussi importantes entre les dépôts et les clients. Pour cela, on propose de regrouper les clients les plus proches à un dépôt exclusif, et de faire une étude pour chaque dépôt exclusif séparément, comme un problème de tournée de véhicule. Dans ce cas, nous proposons plusieurs objectifs pour chaque dépôt.

Dans le but de simplifier la résolution de notre problème, nous fixons quelques objectifs intermédiaires. Ces objectifs sont basés sur le coût total de la tournée, le nombre de véhicules, la somme totale des retards et un ensemble de contraintes à respecter. Nous les formulons comme suit :

- La première fonction objectif est de minimiser le nombre de routes "véhicules utilisés" donnés par :

$$\text{Min } k \quad \text{Eq. III.5}$$

La deuxième fonction objectif est de minimiser le coût total de la tournée :

$$\text{Min } \left(\sum_{k=1}^m \sum_{i=1}^n \sum_{j=1}^k c_{ij}^k \cdot d_{ij} \right) \quad \text{Eq. III.6}$$

La troisième fonction objectif est de minimiser la somme totale des retards :

$$\text{Min } \left(\sum_{i=1}^n \text{Max}(0, [u_i^k + t_i] - b_i) \right) \quad \text{Eq. III.7}$$

7.4. Formulation

Après avoir défini les fonctions objectifs et l'ensemble des contraintes à respecter, nous récapitulons la formulation mathématique de notre problème comme suit :

$$\text{Min } k \quad \text{Eq. III.8}$$

$$\text{Min } \left(\sum_{k=1}^m \sum_{i=1}^n \sum_{j=1}^k c_{ij}^k \cdot d_{ij} \right) \quad \text{Eq. III.9}$$

$$\text{Min } \left(\sum_{i=1}^n \text{Max}(0, [u_i^k + t_i] - b_i) \right) \quad \text{Eq. III.10}$$

Sous les contraintes :

$$\sum_{k \in K} y_i^k = 1 \quad (i \in Z) \quad \text{Eq. III.11}$$

$$\sum_{k \in K} \sum_{i \in (Z \cup P)} x_{ij}^k \geq 1 \quad (j \in Z) \quad \text{Eq. III.12}$$

$$\sum_{i \in Z} d_i \sum_{j \in (Z \cup P)} x_{ij}^k \leq Q_k \quad (k \in K) \quad \text{Eq. III.13}$$

$$\sum_{i \in Z} \sum_{j \in Z} (t_{ij} x_{ij}^k + t_i y_i^k) \leq T_k \quad (k \in K) \quad \text{Eq. III.14}$$

$$\sum_{k \in K} y_i^k d_i = 1 (i \in Z) \quad \text{Eq. III.15}$$

$$S_{1k} - S_{jk} + N x_{ij}^k \leq N-1, \quad (i,j) \in J, k \in K \quad \text{Eq. III.16}$$

$$a_i y_i^k \leq u_i^k \leq b_i y_i^k \quad (i=0, \dots, n-1); \quad (k \in K) \quad \text{Eq. III.17}$$

$$u_i^k + t_i + t_{ij} - M(1-x_{ij}^k) \leq b_j \quad (i \in Z); (k \in K) \quad \text{Eq. III.18}$$

$$\sum_{i \in Z} t_i y_i^k + \sum_{i \in Z} \sum_{(j \in Z)-1} x_{ij}^k t_{ij} + (t_s + t_{is} + t_{sj} - t_{ij}) \leq T_k (s \in Z), (k \in K) \quad \text{Eq. III.19}$$

$$x_{ij}^k, y_i^k, D_i^k \in (0,1) \quad (i,j \in Z; i \neq j); (k \in K) \quad \text{Eq. III.20}$$

$$Q_k, q_i, d_{ij}, T_k, t_i, t_{ij}, M \geq 0 \quad \text{Eq. III.21}$$

Ce modèle mathématique exprime le fait que l'on cherche à déterminer un ensemble de tournées tout en minimisant : le coût total de la tournée (Eq. III.9), le nombre de véhicules (Eq. III.8), la somme totale des retards (Eq. III.10). La contrainte (Eq. III.13) exprime le fait que la capacité des véhicules est limitée. La contrainte (Eq. III.15) assure que les clients visités sont ceux qui ont effectués une demande. La contrainte (Eq. III.12) assure que chaque véhicule a visité au-moins un dépôt. (Eq. III.16) permet d'éliminer les trajets non aboutis c-à-d ne passant pas par l'ensemble des clients à visiter. La contrainte (Eq. III.17) assure que les nœuds doivent être servis dans leurs fenêtres de temps respectifs. La contrainte (Eq. III.18) permet d'établir une chronologie de distribution aux véhicules au cours de leurs trajets. La prise en compte d'une limite sur la durée de chaque tournée (Chaque produit doit être distribué avant un temps donné i.e. un temps de travail), notée T_k peut se faire en introduisant la contrainte (Eq. III.14). Cependant, la contrainte (Eq. III.19) vérifie la possibilité de l'insertion du nouveau nœud « s » si le temps T_k le permet. La binarité des variables de décision est donnée par la contrainte (Eq. III.20).

8. Conclusion

Ces dernières années, on remarque une importante croissance du réseau routier en Algérie, la création des zones industrielles, une importante augmentation dans le parc d'automobiles, ce qui engendre un encombrement dans la circulation des véhicules surtout au niveau des centres villes, une augmentation de la consommation du carburant et par fois la pénurie de cette matière essentielle dans l'industrie et le transport.

Le type de produit à distribuer est l'un des critères les plus essentiels à prendre en considération lors de la distribution. Par exemple, la distribution des produits alimentaire est caractérisée par la livraison fréquente de petits volumes à des clients nombreux et dans un intervalle de temps étroit, le type de véhicules à utiliser (selon la quantité à distribuer) et la moyenne de la durée d'une tournée (nombre d'heures de travail du chauffeur).

La distribution des produits dans un réseau routier tel que celui de l'Algérie est un contexte riche en complexités. Il est nécessaire de le traduire en problème de tournées de véhicules. Dans ce chapitre, nous avons proposé une modélisation mathématique multi-objectif pour notre problématique.

Après une étude approfondie des différentes extensions des problèmes VRP et leurs contraintes, et dans le but de proposer une meilleure résolution pour notre problème, qui permet à son tour de converger vers des résultats optimaux dans des délais raisonnables et sans tomber dans la barrière des optimaux-locaux, nous avons opté pour l'utilisation des méta-heuristiques « plus précisément les algorithmes génétiques ».

CHAPITRE 4 :
RÉSOLUTION MONO-
OBJECTIF DU MD-VRPTW
PAR ALGORITHMES
GÉNÉTIQUES

1. Introduction

Après une étude approfondie des méthodes de résolutions déjà existantes dans la littérature et les plus utilisées dans le domaine VRP (montrée schématiquement par la Fig. II.5), et dans le but d'atteindre les objectifs de notre projet de recherche, nous avons opté pour une hybridation entre les algorithmes génétiques « avec un meilleur ajustement de leurs paramètres » et un algorithme d'apprentissage « Kmeans », où nous l'avons intégré dans l'une des étapes de fonctionnement de base de l'algorithme génétique, exactement dans la phase de génération de la population initiale dans le but d'assurer la notion de regroupement « Clustering » pour une meilleure convergence vers l'optimum global. Notre proposition est nommée **ProGenClust**.

Les algorithmes génétiques ont été utilisés pour la première fois par Holland [93] et ses étudiants en 1962. Ce travail a ensuite été étendu due à la contribution de David Goldberg [94] en 1989. Les algorithmes génétiques sont basés sur un mécanisme de recherche parallèle, ce qui les rend plus efficaces que d'autres techniques classiques d'optimisation telles que la méthode de « Branch and bound », la recherche tabou et le recuit simulé.

Les algorithmes génétiques imitent le mécanisme de la sélection naturelle et la survie des plus forts comme en témoigne l'évolution naturelle, ils commencent par générer une population aléatoire d'individus. Pour passer de la génération « k » à la génération « k+1 », les trois opérateurs : croisement, mutation et évaluation sont répétés pour tous les éléments de la population « k ». L'opérateur de croisement est appliqué aux deux parents P1 et P2 avec une probabilité P_c et génère deux enfants E1 et E2. L'opérateur de mutation est appliqué avec une probabilité P_m (P_m est généralement inférieur à P_c) et génère un enfant muté E. L'évaluation est ensuite appliquée aux enfants E1, E2 et E avant de les insérer dans la nouvelle population. La figure (Fig. IV.1) détaille schématiquement les principales étapes de fonctionnement d'un algorithme génétique.

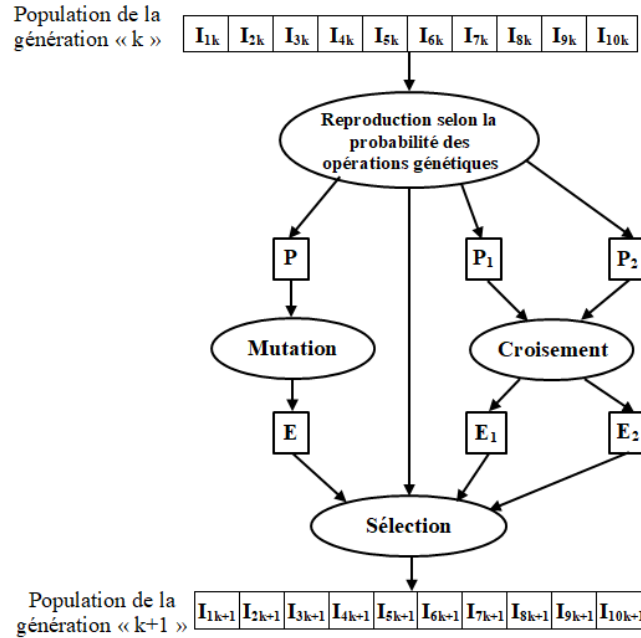


Fig. IV.1 Principe de fonctionnement d'un algorithme génétique

Le point fort des algorithmes génétiques réside dans le fait qu'ils peuvent éviter de se faire piéger dans un optimum local en accordant les opérateurs génétiques, le croisement et la mutation. En raison de son potentiel élevé d'optimisation globale, les algorithmes génétiques ont reçu une grande attention dans le domaine de résolution des problèmes de routage des véhicules.

Maintenant, nous présentons en détail la représentation des chromosomes, la fonction d'évaluation appelée « fitness » et d'autres composants des algorithmes génétiques déjà utilisés pour les problèmes de routage des véhicules. Après l'étape d'initialisation de l'algorithme génétique, une stratégie de regroupement « Clustering » est utilisée pour affecter tous les clients un par un à un dépôt donné.

Ensuite, un ensemble de solutions candidates (chromosomes) est généré et transformé de manière aléatoire par un planificateur de route en un ensemble de routes, puis les chromosomes sont soumis à un processus évolutionnaire jusqu'à ce qu'un nombre minimum possible de routes soit atteint ou l'un des critères d'arrêts est satisfait.

Les algorithmes génétiques intègrent une mutation adaptative inter-dépôt, qui permet si possible d'améliorer les affectations des clients initiaux aux dépôts correspondants. Une sélection par tournoi avec retenue [94] est utilisée pour effectuer une sélection des individus basés sur l'évaluation de la fitness pour une reproduction évolutive. Algo IV.1 décrit la méthodologie des algorithmes génétiques.

1. Générer une population initiale, POP;
2. **Évaluer** la fonction fitness $F(x)$ pour chaque chromosome x de la population, et calculer la fitness moyenne;
3. Créer une nouvelle population en répétant les étapes suivantes jusqu'à ce que la nouvelle population soit complète;
 - **Sélection** : Sélectionnez deux chromosomes parent de la population en utilisant la sélection par tournoi;
 - **Recombinaison** : Appliquer une opération de croisement avec une probabilité P_c aux parents pour former de nouveaux enfants.
 - **Mutation** : Appliquer une opération de mutation intra-dépôt ou inter-dépôt avec une probabilité P_m pour former de nouveaux enfants;
 - **Acceptation** : Intégrer de nouveaux enfants dans la population, remplaçant les parents;
 - **Elitisme** : Remplacer aléatoirement 1% de la population initiale par 1% de la meilleure population d'enfants;
4. Mettre à jour l'ancienne population avec la nouvelle population produite;
5. **Si** Le nombre actuel de générations est atteint **Alors** renvoyer la fitness moyenne et la fitness de la meilleure solution (chromosome) dans la population actuelle;
6. **Sinon** Aller à l'étape 2;

Algo IV.1 Méthodologie des algorithmes génétiques

2. Schéma général de ProGenClust

Notre proposition est basée sur une hybridation entre un algorithme génétique que nous proposons, pour résoudre un problème MD-VRPTW, et un algorithme d'apprentissage « Kmeans ». Nous schématisons les différentes étapes de l'algorithme ProGenClust dans la figure suivante :

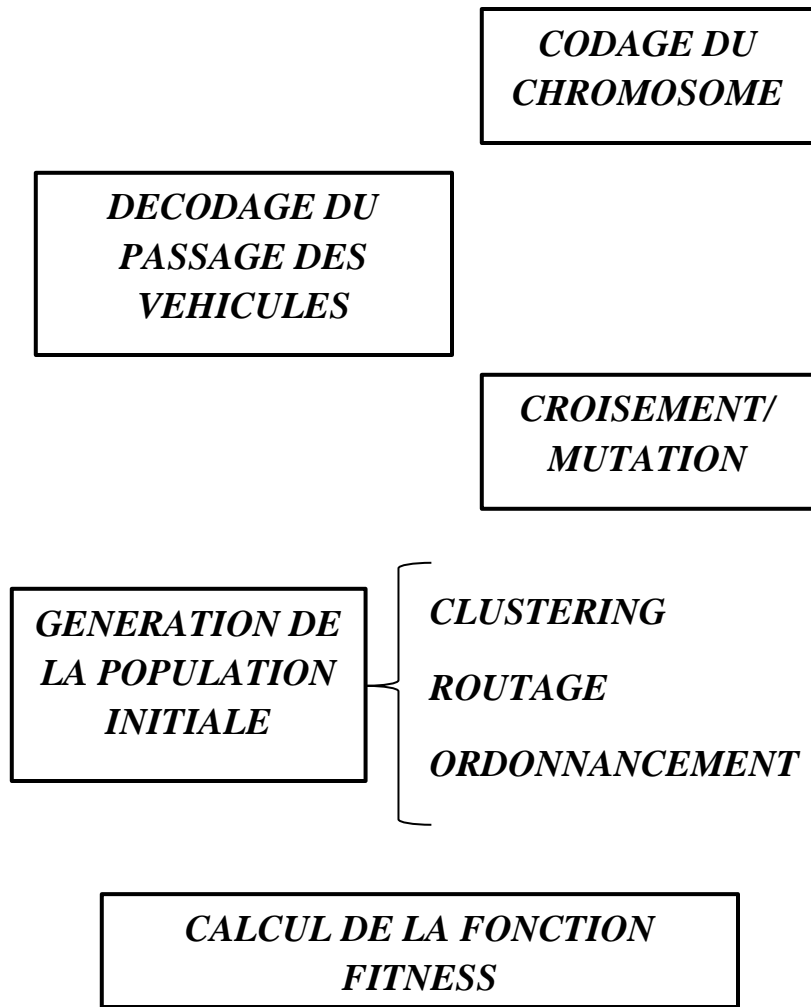


Fig. IV.2 Schéma général de ProGenClust

3. Codage du chromosome

L'encodage permet de représenter les solutions en tant que chromosomes. Un chromosome est une séquence de nœuds, indiquant l'ordre dans lequel les véhicules doivent visiter tous les nœuds. Ce type de codage (Table IV.1) s'appelle codage par liste de permutation.

Table IV.1 Codage par liste de permutation

Position	Début	1 ^{er}	2 ^{ieme}	3	4	5	6	7	8	9	10	Fin
Nœud (i)		5	2	6	4	3	10	8	7	9	1	

4. Décodage du passage des véhicules

Le décodage permet, à partir de chaque chromosome de la population, d'obtenir une solution initiale indiquant le passage de chaque véhicule sur les nœuds correspondants.

Table IV.2. Ordre du passage des véhicules

Position	Coût	Début	1 ^{er}	2 ^{ème}	3 ^{ème} /Fin	4 ^{ème}	Fin
V1	C1		2	4	1	5	
V2	C2		3	6	7	10	
V3	C3		8	9			

5. Croisement

Après la génération aléatoire de la population initiale, nous procédons à la phase de croisement qui assure la recombinaison des gènes parentaux pour former de nouvelles générations.

Dans ce travail, nous avons appliqué une nouvelle technique appelée the Best Cost Route Crossover (BCRC) développée par D. Fogel. [115], pour le problème de routage des véhicules avec fenêtres de temps (VRPTW), qui permet de garantir que les solutions générées par l'évolution génétique sont réalisables. Nous montrons en outre l'applicabilité de BCRC en l'étendant au MDVRPTW, mais avec quelques légères améliorations.

Les étapes de la technique CRCB sont :

- Choisir les parents pour sélection par tournoi ;
- Sélectionnez au hasard un dépôt pour subir la reproduction ;
- Sélectionnez une route de chaque parent de manière aléatoire ;
- Supprimez tous les clients appartenant à la route 1 du parent 2 ;
- Supprimez tous les clients appartenant à la route 2 du parent 1 ;
- Pour chaque client appartenant à la route 1
 - Calculer le coût d'insertion de la route 1 dans chaque emplacement du parent 2 et stockez les coûts dans une liste ordonnée ;
 - Pour chaque emplacement d'insertion, vérifié si l'insertion est possible ou non ;
 - Générer un nombre aléatoire « r » ($0 \leq r \leq 1$).
 - Choisir le premier emplacement d'insertion possible si $r_n \leq$ seuil ;
 - Sinon, choisir la première entrée dans la liste ordonnée, quelle que soit la faisabilité ;
- Répéter l'étape précédente pour chaque client appartenant à la route 2.

6. Mutation

La mutation est un opérateur génétique utilisé pour maintenir la diversité génétique d'une génération de population de chromosomes à la suivante. Il modifie un ou plusieurs gènes dans un chromosome à partir de son état initial. La mutation est utilisée pour trouver une solution pour le MD-VRPTW en se basant sur les algorithmes génétiques. Une sous-chaîne du parent est sélectionnée de manière aléatoire et permutée dans le but de former un nouvel enfant.

Nous avons utilisé différents types de mutations pour permettre de multiples façons de se libérer des minimums-locaux et d'améliorer les coûts de la tournée lorsque cela est possible. Dans la littérature, nous avons trouvé deux types de mutations :

6.1. Mutation « Intra-Dépôt »

Les mutations appliquées à des routes à dépôts unique sont appelées mutations intra-dépôts, elles sont efficaces pour intégrer la diversité dans les routes de chaque dépôt. Trois types de mutations Intra-Dépôts ont été étudiés :

6.1.1. Mutation d'inversion « Reversal mutation »

Nous proposons une adaptation d'une simple mutation largement utilisée, généralement appelée « Reversal mutation » [116]. Deux points de coupe sont sélectionnés dans le chromosome associé au dépôt déjà choisi et les valeurs des deux gènes entre ces deux points de coupure sont inversées. Un exemple de mutation par inversion est illustré dans la figure (Fig. IV.3) et qui consiste à choisir au hasard deux gènes dans un chromosome et échanger leurs valeurs.

1	4	6	2	8	0	7	5	3	9	10
Avant mutation (Parent)										
1	4	7	2	8	0	6	5	3	9	10
Après mutation (Enfant)										

Fig. IV.3 Exemple de Mutation par inversion

6.1.2. Re-routage des clients individuels « Single customer re-routing »

Dans ce cas, nous sélectionnons au hasard un client, et nous l'éliminons de l'itinéraire existant. Ce client est ensuite inséré dans le meilleur emplacement d'insertion possible dans tout le chromosome. Cela implique de calculer le coût total d'insertion à chaque localisation d'insertion, qui réinsère finalement le client dans le lieu le plus réalisable.

6.1.3. Permutation « Swapping »

Cet opérateur de mutation simple sélectionne aléatoirement deux routes et échange un client choisi de manière aléatoire d'un itinéraire à un autre.

6.2. Mutation Inter-Dépôts

La mutation Inter-Dépôts permet d'échanger des clients d'un dépôt à un autre, ce qui contribue à améliorer la qualité de la solution, puisque le regroupement statique initial (Clustering) qui permet l'affectation des clients aux dépôts disponibles, fait une hypothèse imposante sur le groupe où chaque client doit être intégré.

Chaque 10 générations, au lieu d'appliquer une des mutations intra-dépôt déjà décrites ci-dessus, on utilise un opérateur de mutation inter-dépôts (si c'est applicable bien sûr).

Dans la mutation inter-dépôts, les clients peuvent réduire les coûts totaux de l'itinéraire, la somme totale des retards ou le nombre de véhicules utilisés en les réaffectant à différents dépôts.

Notez ici que nous devons respecter la contrainte de capacité, afin d'assurer la non-surcharge de chaque véhicule.

7. Génération de la population initiale

Le mécanisme de génération de la population initiale permet la production d'une population d'individus comme base pour les futures générations. Nous jugeons important le choix de la population initiale car il peut rendre plus ou moins rapide la convergence vers l'optimum global.

Afin de converger rapidement vers l'optimum global, nous avons divisé la tâche de génération de la population initiale en deux phases :

La première phase appelée (Clustering) consiste à partager équitablement les nœuds sur les dépôts, de manière à affecter à chaque dépôt les nœuds les plus proches, comme indiqué dans la figure (Fig. IV.4).

Dans l'exemple illustré dans la figure (Fig. IV.5), il y a 10 clients désignés 1-10. Deux routes sont requises par les véhicules pour servir tous les clients pour l'instance (A 8 4 A 9 5 1 A). La première route part du dépôt A et se déplace vers les clients 8 et 4 et retourne au même dépôt. De même, la deuxième route commence à partir du dépôt A, sert les clients 9, 5, 1 et revient au même dépôt. La solution réalisable pour le processus d'optimisation est générée en trois étapes de base : le regroupement, le routage et l'ordonnancement.

7.1. Le regroupement (Clustering)

Dans la littérature, on trouve différentes heuristiques de voisinages dédiées clustering. Dans notre cas, nous avons appliqué le principe de l'algorithme Kmeans[63] avec une légère modification afin d'assurer le clustering. Nous avons procédé comme suit :

- Choisir k nœuds dans l'espace $k \in S$ (les k nœuds = les dépôts utilisés) ;
- Tant que $\exists x \in S$ non visité Faire
 - Calculer la distance euclidienne entre chaque nœud de l'ensemble S et les dépôts existants ;
 - Affecter un nœud au plus proche dépôt « selon la règle de distance détaillé ci-dessous »;
- Une fois terminé, nous aurons k clusters.

Initialement, chaque client est affecté au plus proche dépôt en termes de distance euclidienne [117]. Comme discuté ci-dessus, pour k dépôts dans le MD-VRPTW, le chromosome se compose de « k » clusters et les clients sont affectés à chacun de ces « k » clusters [118]. Dans l'exemple (Fig. IV.4), il existe deux dépôts A et B, chaque client C_i doit être affecté à un seul dépôt exactement. Ce processus de regroupement est effectué en fonction du calcul de la distance selon la règle suivante :

- Si $D(C_i, A) < D(C_i, B)$, le client C_i est affecté au dépôt A
- Si $D(C_i, A) > D(C_i, B)$, le client C_i est affecté au dépôt B
- Si $D(C_i, A) = D(C_i, B)$, le client C_i est affecté à un dépôt choisi arbitrairement entre A et B
- $D(C_i, P)$: représente la distance entre le client C_i et le dépôt P, tels que :

$$D(C_i, P) = \sqrt{(X_{Ci} - X_P)^2 + (Y_{Ci} - Y_P)^2} \quad \text{Eq. IV.1}$$

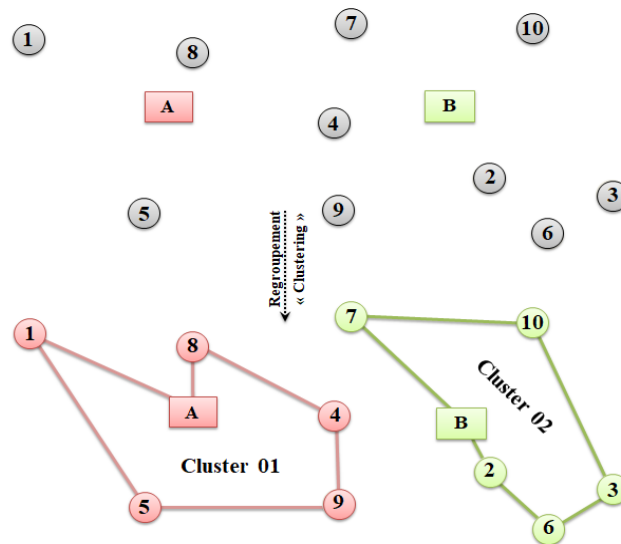


Fig. IV.4 Exemple de regroupement « clustering »

7.2. Le routage

Dans un problème MD-VRPTW, un chromosome doit spécifier le nombre de routes (c'est-à-dire les véhicules) et l'ordre de livraison dans chaque itinéraire. Les clients du même groupe sont affectés à plusieurs itinéraires en utilisant la méthode Clarke et Wright Saving (G. Clarke et al. (1964))[48]. Le processus de routage est basé sur la distance parcourue par les véhicules pour servir les clients. Une matrice d'économie S (C_i, C_j) est construite pour chaque deux clients i et j dans le même lien. Tel que,

$$S(C_i, C_j) = D(P, C_i) + D(P, C_j) - D(C_i, C_j) \quad \text{Eq. IV.2}$$

7.3. L'ordonnancement

A ce stade, un véhicule doit partir du dépôt et, à partir du premier client indiqué par le premier gène du chromosome, la séquence de livraison est choisie de telle sorte que le prochain client soit aussi proche du client précédent. Cette dernière est assurée en appliquant le principe de l'heuristique de voisinage 'le plus proche voisin' KPP[55], et renforcée par les deux algorithmes de correction de successeurs et de capacité « comme le montre l'(Algo IV.3) », en tenant compte du fait que les contraintes de longueur d'itinéraire et de capacité du véhicule ne soient pas violées avant d'ajouter un client à l'itinéraire actuel. Ce processus continu jusqu'à ce que chaque client soit affecté à une seule route. Le principe d'ordonnancement est illustré dans (Algo IV.2).

```

I : Individu ;
n : Nombre de gènes par individu ;
Début
  m := 1 ;
  Tanque m <= n Faire
    i := m ;
    Tanque I[i] <> 0 et I < n Faire
      min := i ;
      j := i + 1 ;
      Tanque I[j] <> 0 et j <= n Faire
        Si I[j] ≤ I[min] Alors
          min := j ;
        FinSi
      j := j + 1 ;
    FinTQ
    i := i + 1 ;
  FinTQ
  m := i + 1 ;
FinTQ
Fin.

```

Algo IV.2 Algorithme d'ordonnancement

```

I : individu ;
n : Nombre de gènes par individu ;
qt[] : Tableau des quantités à distribuer pour chaque client;
Début
   $Q_{\max}^T$  ; // Capacité maximale des véhicules de type T
  qt=0 ; // Capacité actuelle
  m :=1 ;
  Tantque m <= n Faire
    i :=m ;
    Tantque I[i] <>0 et i<=n Faire
      Si qt + qt[i] ≤  $Q_{\max}^T$  Alors
        qt = qt + qt[i] ;
      Sinon
        J :=i+1 ;
        Tantque I[j] <>0 et j<=n Faire
          Si qt + qt[j] ≤  $Q_{\max}^T$  Alors
            qt := qt + qt[j] ;
            N:=I[j]
            Pour l de j a i+1,-1 Faire
              I[l] = I[l-1] ;
            FinPour
            I[i] := N
          FinSi
          j :=j+1 ;
        FinTQ
      FinSi
      i :=i+1 ;
    FinTQ
    m :=i+1 ;
  FinTQ
Fin.
    
```

Algo IV.3 Algorithme de correction de capacité

A la fin de la phase d'ordonnancement, une solution réalisable de l'exemple du problème MD-VRPTW est construite comme le montre la figure (Fig. IV.5).

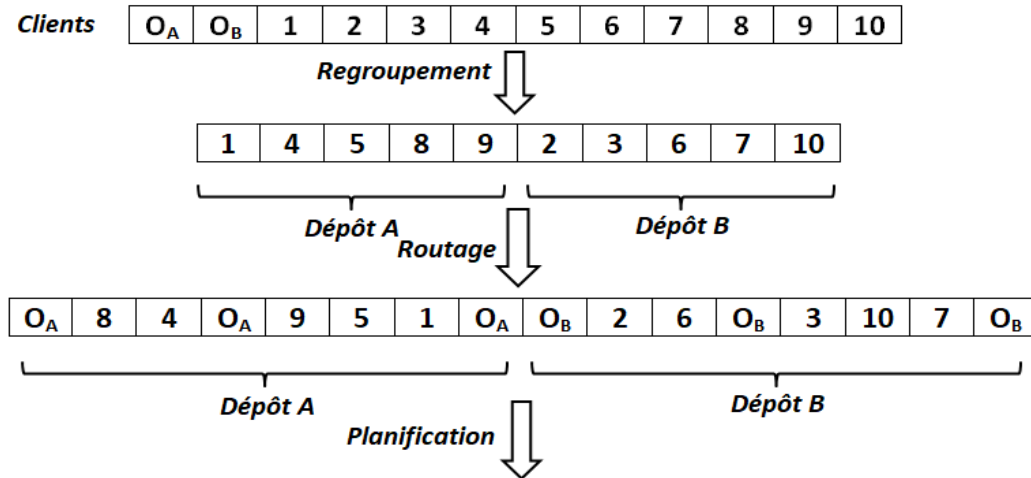


Fig. IV.5 Exemple d'un processus d'optimisation d'un problème MD-VRPTW

8. Procédure de calcul du « fitness »

Chaque véhicule doit servir les nœuds dans sa tournée dans l'ordre d'affectation. Une fois de retour au dépôt, nous calculons le coût du transport correspondant à la distance parcourue, et nous répétons cela jusqu'à ce que tous les nœuds soient servis.

Nous reproduisons cette tâche pour chaque individu de la population, en prenant en compte les différentes combinaisons possibles entre les véhicules et les nœuds qui correspondent aux clients afin d'obtenir ultérieurement les individus qui minimisent notre fonction objective.

Une fois que chaque chromosome a été transformé en une topologie de réseau faisable en se basant sur les techniques de groupement, routage et ordonnancement donné ci-dessus, la valeur de fitness de chaque chromosome est déterminée en utilisant une fonction de fitness pondérée « weighted-sum » [119].

Cette méthode nécessite d'ajouter les valeurs des fonctions de fitness en utilisant des coefficients pondérés pour chaque objectif individuel. La fonction fitness $F(x)$ d'un individu x est défini comme suit :

$$F(x) = \beta \cdot \sum_{k \in K} V_k \quad \text{Eq. IV.3}$$

$$V_k = \sum_{i \in Z} \sum_{j \in Z} d_{ij} \quad \text{Eq. IV.4}$$

β est un paramètre de poids associé à la distance totale parcourue par les véhicules. La valeur du paramètre de poids utilisé dans cette fonction a été fixée à $\beta = 0.001$, ce qui concentre efficacement l'évolution de la minimisation de la longueur de la route.

9. L'algorithme génétique proposé pour la minimisation du coût total de transport

On suppose que « N » est la taille de la population initiale, « F_i » la fonction objectif à minimiser (donnée par l'équation Eq. IV.5) correspondant à l'itération i. Nous déterminons ensuite :

$$f_{\text{recherché}} = \min_i f_i \quad \text{Eq. IV.5}$$

L'algorithme génétique développé pour résoudre notre problème et pour minimiser le coût total du transport est illustré dans Algo IV.4.

```

Début
Etape 1: Création de la population initiale de taille = n);
Etape 2: Affectation de tous les nœuds (clients) aux dépôts
correspondants « création des clusters »;
Tanque nombre de générations non atteint Faire
  Début
    Etape 3: Création de la population intermédiaire de taille = 2n avec
    de nouveaux individus de "croisement ou mutation"
    Etape 4: Application des procédures d'ordonnancement et de
    correction de capacité; «Appelez les algorithmes Algo IV.2,
Algo IV.3 »
    Etape 5 : Création de la deuxième population intermédiaire de
    taille = 2n * 2n représentant les différentes routes prises par
    chaque véhicule;
    Etape 6 : Déterminer la valeur de fitness pour chaque individu dans
    la nouvelle population (coût de transport);
    Etape 7 : Ordonner la population en fonction de la valeur de fitness
    minimale (coût de transport);
    Etape 8 : // Copier la meilleure solution;
    Si la solution actuelle est la meilleure Alors
      La meilleure solution = la solution actuelle;
    Sinon
      Incrémenter le nombre de générations;
  FinTQ
Fin.
  
```

Algo IV.4 L'algorithme génétique proposé pour résoudre un mono-objectif MD-VRPTW

10. Conclusion

Nous avons présenté la méthodologie utilisée basée principalement sur une hybridation entre les algorithmes génétiques et le principe de deux méthodes (l'heuristique du plus proche voisin et l'algorithme d'apprentissage Kmeans), qui ont été intégré directement dans les étapes d'optimisation des algorithmes génétiques, afin d'assurer le regroupement « clustering » dans la phase de génération de la population initiale pour une convergence rapide vers l'optimum global ainsi que de garantir un meilleur ordonnancement dont le but est la résolution mono-objective de notre problème « MD-VRPTW ». Après avoir présenté cette méthodologie, nous avons détaillé les différentes procédures de corrections utilisées dont le but est de renforcer la phase d'ordonnancement tel que : la procédure de correction de successeur et de capacité. A la fin, nous avons proposé un encodage qui représente les paramètres caractérisant chaque individu, et nous avons proposé notre propre algorithme qui permet de minimiser le coût total de transport.

Dans ce présent travail, minimiser le coût total de transport n'est pas notre seul objectif, nous devons également minimiser le nombre de véhicules utilisés ainsi que la somme totale des retards. Pour cela, nous présenterons, dans le chapitre suivant, la notion d'optimalité Pareto et son utilisation pour résoudre notre problème dans le but d'optimiser la totalité de nos critères.

CHAPITRE 5 :
OPTIMISATION
MULTI-OBJECTIF POUR
LA RÉSOLUTION DU
« MD-VRPTW »

1. Introduction

Pour la plupart du temps, on trouve que les problèmes d'optimisation combinatoire provenant de cas réels sont, par nature, multi-objectif. Plusieurs critères, qui peuvent être contradictoires, sont à considérer simultanément pour résoudre le problème.

L'optimisation multi-objectif tire ses racines de l'économie. Malgré l'intérêt de l'étude de ce type de problèmes, il va falloir attendre les années 80-90 pour voir émerger des méthodes de résolution de problèmes multi-objectif.

Dans un problème d'optimisation multi-objectif, on ne cherche pas à optimiser une fonction objectif unique mais un vecteur de fonctions :

$$(PMO) = \begin{cases} \min F(x) = (f_1(x), f_2(x), \dots, f_n(x)) \\ \text{sous contraintes} \\ g(x) \geq 0 \\ x \in X \end{cases} \quad \text{Eq. V.1}$$

Où $n \geq 2$ est le nombre de fonctions objectif et $F(x) = (f_1(x), f_2(x), \dots, f_n(x))$ est le vecteur des fonctions à optimiser. Le vecteur x et les ensembles X et Ω sont définis de la même manière que pour un problème d'optimisation mono-objectif.

Dans le cadre multi-objectif, le décideur raisonne plutôt en termes d'évaluation d'une solution pour chaque objectif et se place naturellement dans l'ensemble $F(X)$, appelé espace des objectifs, l'ensemble $Y = F(\Omega)$ représente les points réalisables de l'espace des objectifs et $y = (y_1, y_2, \dots, y_n)$ avec $y_i = f_i(x)$, avec $x \in \Omega$, est un point de l'espace des objectifs.

2. Résolution d'un problème de décision multi-objectif

Il n'existe qu'un seul optimum global dans le cas d'un problème d'optimisation mono-objectif avec éventuellement plusieurs solutions correspondantes, ce n'est plus le cas pour un problème d'optimisation multi-objectif. Il faut donc utiliser une autre notion d'optimum.

La plus communément admise est l'optimalité Pareto. On parle alors d'optimum Pareto qui se définit comme suit :

Une solution $x^* \in \Omega$ est dite Pareto optimale si pour tout $x \in \Omega$, soit la proposition suivante est vraie :

$$\forall i \in \{1, 2, \dots, n\}, f_i(x) = f_i(x^*),$$

Soit la proposition suivante est vérifiée :

$$\exists i \in \{1, 2, \dots, n\}, f_i(x) > f_i(x^*).$$

Cette définition peut s'interpréter de la manière suivante : une solution x^* est Pareto optimale s'il n'existe pas une solution réalisable x qui améliore la valeur d'un objectif sans détériorer au moins celle d'un autre objectif. Les solutions Pareto optimales sont aussi connues sous les noms de solutions admissibles, efficaces, non-dominées et non-inférieures [120].

D'autres notions sont associées à l'optimalité Pareto. Il est possible de définir une relation d'ordre partiel, appelée relation de dominance ou relation de dominance Pareto :

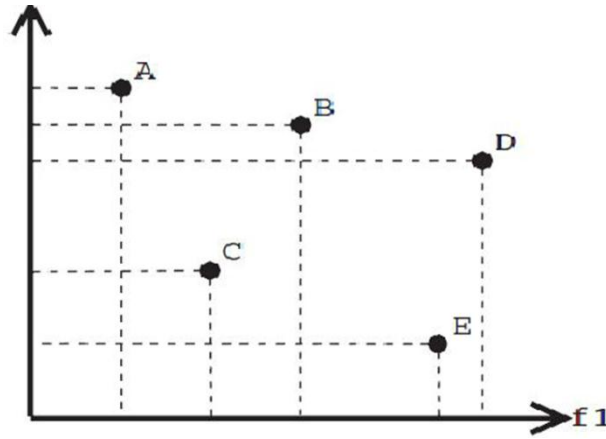


Fig. V.1 Exemple de dominance.

Une solution y domine (notée \succeq) une solution z (ou $y \succeq z$) si et seulement si

$$\forall i \in \{1, 2, \dots, n\}, f_i(y) \leq f_i(z) \text{ et } \exists i \in \{1, 2, \dots, n\}, f_i(y) < f_i(z).$$

Dans l'exemple de la figure (Fig. V.1), les points A, C, E ne sont pas dominés par d'autres points. Ils sont donc Pareto optimaux. Tandis que le point B est dominé par le point C et le point D est dominé par les points C et E. Les points A, C, E sont au même niveau, c'est-à-dire que d'un point de vue multi-objectif il n'est pas possible d'établir une préférence pour un point plutôt que pour un autre.

Cette relation de dominance est parfois nommée relation de dominance faible. Cette définition peut être modifiée pour définir une relation de dominance forte.

Une solution y domine fortement (notée \prec) une solution z (ou $y \prec z$) si la solution y est strictement meilleure que z sur tous les objectifs :

$$\forall i \in \{1, 2, \dots, n\}, f_i(y) < f_i(z).$$

Le vecteur idéal est le point de $F(X)$ composé de l'optimum pour chaque objectif pris séparément. La condition nécessaire et suffisante pour que ce vecteur idéal soit atteint (c'est-à-dire qu'il correspond à une solution réalisable) est que les fonctions objectif soient indépendantes. Si c'est le cas, il suffit de résoudre n problèmes mono-objectif. La solution d'un problème d'optimisation multi-objectif n'est donc pas une unique solution mais un ensemble de solutions appelé **ensemble Pareto optimal**.

Pour un problème d'optimisation multi-objectif donné, l'ensemble Pareto optimal (P^*) est défini comme suit :

$$P^* = \{ \forall x \in \Omega \mid \nexists x' \in \Omega, F(x') \preceq F(x) \} \quad \text{Eq. V.2}$$

L'image de l'ensemble Pareto optimal dans Y est appelée frontière Pareto ou surface de compromis.

Formellement, pour un problème d'optimisation multi-objectif donné et un ensemble Pareto optimal P^* , la frontière Pareto (PF^*) est définie comme suit :

$$PF^* = \{ u = (f_1(x), \dots, f_n(x)) \mid x \in P^* \} \quad \text{Eq. V.3}$$

Remarque :

Les problèmes étudiés en optimisation sont souvent difficiles et appartiennent à la classe de complexité NP-difficile, ce qui rend impossible la recherche de l'optimum global lorsque la taille du problème devient trop importante. Si un objectif d'un problème multi-objectif est NP-difficile alors le problème multi-objectif sera lui-même NP-difficile.

Il est à noter que même si tous les objectifs sont solubles en un temps polynomial, alors, sauf dans certains cas particuliers, le problème multi-objectif n'est pas soluble en un temps polynomial. Cela limite l'utilisation des méthodes exactes et encourage l'utilisation d'heuristiques dédiées et surtout de méta-heuristiques, notamment les méthodes à base de voisinage et les algorithmes génétiques. On ne cherche plus l'ensemble Pareto optimal mais une approximation de ce dernier. Pour un algorithme « A » donné, on parle alors de solutions potentiellement Pareto optimales qui forment l'ensemble potentiellement Pareto Optimal, Ehrgott [121].

3. Approches de résolution d'un problème de décision multi-objectif

La présence de plusieurs objectifs contradictoires et l'absence d'ordre total entre les solutions demandent la mise au point de stratégies capables de prendre en compte ces particularités. Cette section présente les approches qui permettent de rendre les problèmes d'optimisation multi-objectif solubles. Nous présentons d'abord deux classifications de ces stratégies qui reposent sur deux points de vue différents du problème : celui du décideur et celui du concepteur. Puis nous classerons les principales stratégies par rapport à la classification du concepteur.

4. Classification du point de vue du décideur

La solution d'un problème multi-objectif est un ensemble de solutions. Cependant, pour un problème réel, une seule solution pourra être déployée. Un choix par un décideur doit donc être effectué ; le décideur pouvant intervenir en amont de la résolution, après celle-ci, ou de manière interactive :

Ceci nous amène à classer les méthodes selon le rôle joué par le décideur [82]. Nous avons les familles suivantes :

- **Les méthodes à priori** : Ces méthodes consistent à modéliser un compromis sur les objectifs avant l'exécution de la méthode. Cela permet d'obtenir une méthode mono-objectif qui nous donne une unique solution et évite de générer différentes solutions du Front Pareto. L'avantage de ces méthodes est de faire une seule recherche de solution. L'inconvénient est de devoir modéliser le compromis avant d'obtenir un résultat et donc il est possible qu'après cette exécution, le décideur ne soit pas satisfait et doit réitérer la méthode avec un nouveau compromis.
- **Les méthodes progressives** : Ces méthodes consistent à réajuster au cours de l'exécution de la méthode le compromis posé avant cette exécution. Elle génère une solution intermédiaire pouvant être qualifiée par rapport aux objectifs. L'avantage de ces méthodes est de pouvoir réorienter les solutions vers le compromis réel souhaité. L'inconvénient est de nécessiter l'intervention du décideur tout au long de la recherche de solution.
- **Les méthodes à posteriori** : Ces méthodes consistent à chercher l'ensemble des solutions de manière à ce qu'elles soient réparties au mieux sur le Front Pareto. Les avantages de cette méthode sont de ne pas avoir à fixer le compromis entre les objectifs avant l'exécution et de proposer un nombre suffisant de solutions au décideur pour qu'il puisse choisir le compromis qu'il souhaite. L'inconvénient majeur est le temps d'exécution.

5. Classification du point de vue du concepteur

Une autre classification utilisée dans la littérature se place du point de vue de concepteur. Les méthodes sont classées dans trois catégories en fonction de la manière dont elles traitent la présence de plusieurs objectifs. Les trois catégories sont, approches scalaires, approches Pareto et approches non scalaires et non Pareto [122].

5.1.Approche scalaire

Ceux sont des méthodes qui "transforment" le problème de base multi-objectif en un problème mono-objectif. Ces méthodes consistent à connaître à priori les préférences du

décideur et fixe ainsi, avant l'exécution de la méthode, le compromis que l'on désire entre les différents objectifs. Dans ces méthodes, nous pouvons citer :

5.1.1. La méthode par agrégation

Cette méthode est la plus évidente. En effet, elle consiste à sommer l'ensemble des objectifs dans une fonction objectif globale. Chaque objectif peut avoir un certain poids, c'est-à-dire que selon le coefficient que l'on choisit pour un objectif, celui-ci peut se trouver favorisé ou non par rapport aux autres. L'avantage de cette méthode est de pouvoir utiliser par la suite toutes les méthodes de résolution de problèmes combinatoires mono-objectif. L'inconvénient majeur est de devoir déterminer les importances relatives des différents objectifs, ce qui nécessite de bien connaître le problème. Si ce n'est pas le cas, le décideur peut ne pas être satisfait du résultat et relance la recherche de solution avec d'autres poids et cela tant qu'il ne sera pas satisfait. L'avantage du temps d'exécution relativement rapide est donc perdu. Elle consiste à transformer un problème multi-objectif en un problème à un objectif en agrégeant les différents critères sous la forme d'une somme pondérée :

$$\left\{ \begin{array}{l} \text{Minimiser } f = \sum_{i=1}^m \omega_i \cdot f_i(x); \\ \text{tel que: } g(x) \leq 0 \\ \text{avec } x \in R^m \quad g(x) \in R^q \end{array} \right. \quad \text{Eq. V.4}$$

Les ω_i , appelés poids et $\sum_{i=1}^m \omega_i = 1$. Il est clair que la résolution d'un problème pour un vecteur de poids ω fixé ne permet de calculer que quelques solutions Pareto optimales. Pour obtenir un ensemble contenant un grand nombre de solutions Pareto optimales, il faut résoudre plusieurs fois le problème en changeant à chaque fois les valeurs de ω Liefoghe [123].

5.1.2. La méthode ε -contraintes

Fait également partie de cette catégorie de méthode. Elle consiste à optimiser un objectif en considérant tous les autres comme contraintes à respecter. Pour trouver plusieurs solutions à un même problème, on fait varier l'objectif à résoudre. L'inconvénient de cette méthode est que l'on optimise un seul objectif, les autres doivent justes être respectés. La méthode doit donc être utilisée plusieurs fois considérant successivement différents objectifs à optimiser pour trouver plusieurs solutions de compromis.

Une autre façon de transformer un problème d'optimisation multi-objectif en un problème mono-objectif est de convertir $m - 1$ des m objectifs du problème en

contraintes et d'optimiser séparément l'objectif restant. Le problème peut être reformulé de la manière suivante :

$$\left\{ \begin{array}{l} \text{Minimiser } g(x) \leq 0 \\ \text{tel que } f_1(x) \leq \varepsilon_1 \\ \vdots \\ f_{i-1}(x) \leq \varepsilon_{i-1} \\ f_{i+1}(x) \leq \varepsilon_{i+1} \\ \vdots \\ f_m(x) \leq \varepsilon_m \\ \text{et que } g(x) \leq 0 \\ \text{avec } x \in R^n, f(x) \in R^m, g(x) \in R^q \end{array} \right. \quad \text{Eq. V.5}$$

L'approche par ε -contrainte doit aussi être appliquée plusieurs fois en faisant varier le vecteur ε pour trouver un ensemble de points Pareto optimaux Xue [124].

5.2.Approches Pareto

Les approches Pareto utilisent directement la notion de dominance au sens de Pareto. Cette idée a été introduite initialement dans les algorithmes génétiques par Goldberg [94], pour résoudre les problèmes multi-objectif proposés dans la théorie. Ce concept permet de ne pas favoriser un compromis plutôt qu'un autre mais il permet de fournir une aide précieuse au décideur. L'utilisation de la dominance est principalement à la base du développement des algorithmes évolutionnaires multi-objectif.

- **La méthode MOGA :** (Multiple Objective Genetic Algorithm) [125] est le premier algorithme qui utilise la notion de dominance directement pour évaluer la performance des individus. Cependant, cette méthode ne permet pas, dans certains cas, d'obtenir une représentation complète du front de Pareto.
- **La méthode SPEA-II :** (Strength Pareto Evolutionary Algorithm II) [125] range les solutions non dominées dans une population externe entre la taille de l'archive et celle de la population courante. Cette méthode diffère de son prédécesseur (SPEA) qui est sensible à l'équilibre entre la taille de l'archive et celle de la population courante, par une taille de l'archive fixe et un calcul de la performance des individus plus raffiné.
- **La méthode NSGA-II :** (Non-Dominated Sorting Genetic Algorithm II) [119] est une des méthodes multi-objectif les plus utilisées. Cette méthode est la version améliorée de la méthode NSGA qui a utilisé, pour la première fois, le concept de classement par dominance. Dans cette méthode NSGA-II, les parents et les enfants sont classés selon le rang de dominance. Lorsque plus de la moitié des parents et des enfants confondus sont dominants, une mesure de densité des solutions est utilisée. Cette méthode NSGA-II ne dépend d'aucun paramètre de contrôle.

5.3.Approches non scalaires et non Pareto

Ces approches sont basées sur des populations de solutions. La recherche est réalisée en traitant séparément les différents objectifs.

5.3.1. Sélection parallèle :

Le premier travail consiste à utiliser des algorithmes génétiques pour résoudre des problèmes d'optimisation multi-objectif. L'algorithme développé par Vector Evaluated Genetic Algorithm (VEGA) sélectionne les individus de la population courante suivant chaque objectif indépendamment des autres (sélection parallèle).

A chaque génération, la population est divisée en un nombre de sous-populations qui est égale au nombre de fonctions objectif composant le vecteur objectif. Ainsi, dans le cas d'un problème à n objectifs résolu par un algorithme génétique travaillant sur k individus, n sous-populations de $\frac{k}{n}$ individus sont créées. Les n populations sont ensuite mélangées pour obtenir une nouvelle population à k individus sur laquelle les opérateurs génétiques sont appliqués Rogers [126].

5.3.2. La méthode lexicographique :

Cette méthode consiste à optimiser le problème selon un premier objectif en posant les autres en contraintes. Ceci nous permet d'obtenir une première solution. Ensuite ce premier objectif est transformé en contrainte d'égalité et l'objectif suivant est optimisé. Cette méthode est ensuite itérée sur l'ensemble des objectifs. La solution obtenue dépend de la hiérarchisation établie sur les différents objectifs. Il est donc important de bien connaître le problème à priori dans Coello [125].

6. VRP et multi-objectif

Les problèmes de tournées de véhicules académiques nécessitent généralement d'être adaptés lorsque l'on se trouve en présence d'un cas réel. Ces adaptations sont souvent des contraintes que l'on rajoute au modèle de départ [1], comme notamment les extensions vues précédemment. De plus, généralement dans les cas réels, il ne suffit pas d'optimiser un seul objectif, mais d'en prendre en compte plusieurs qui peuvent être contradictoires. Les objectifs les plus communs sont de minimiser la distance totale, le temps total de parcours, le coût global, la taille de la flotte, l'équilibrage des tournées, de maximiser la qualité de service ou le profit collecté, etc. De manière générale, Jozefowicz [127] propose dans sa thèse de classer les objectifs selon leur appartenance à une composante du problème : la tournée (coût, travail,...), les nœuds ou les arcs (fenêtres de temps, service du client,...) et les ressources (taille de la flotte, marchandises,...). Chaque objectif sera ainsi défini par son implication dans l'une de ses catégories.

Toujours selon Jozefowicz [128], l'étude des problèmes de tournées de véhicules multi-objectif est principalement due à trois motivations : l'extension des problèmes classiques académiques, la résolution de problèmes réels ou bien encore l'étude d'objectifs complémentaires sans abandonner pour autant l'objectif principal classique qui est celui de la minimisation de la distance total.

Il existe de nombreuses références sur cette problématique en littérature, notamment en étude de problème bi-objectif, dans cette partie nous allons répertorier certaines études de manière non exhaustive.

La plupart des études sur le problème de VRP multi-objectif concernent des études bi-objectifs [127, 129]. Dans ces études, une façon, relativement commune, de prendre en compte plusieurs objectifs consiste à les agréger dans une fonction objectif finale [130]. Une autre approche est apparue avec les colonies de fourmis. En effet, des recherches ont été menées de façon à utiliser une colonie par objectif comme dans [131]. Il existe de nombreuses autres méthodes de résolution, il semblerait toutefois que les méthodes évolutionnaires aient été le plus utilisées [127, 129] et ont prouvé leur efficacité, comme par exemple la méthode NSGA-II [119]. En effet, elles semblent bien se prêter à l'exploration et sont souvent hybridées afin d'intensifier les recherches. Par ailleurs d'autres méthodes peuvent être utilisées comme des heuristiques de construction et d'amélioration [130]. On peut également utiliser d'autres méta-heuristiques comme la méthode tabou [132] ou le recuit simulé [79].

En résumé, les méthodes multi-objectif pour le VRP ont évoluées depuis quelques années. Tandis qu'elles se limitaient à des méthodes scalaires, dorénavant elles tendent à exploiter les méta-heuristiques. Nous avons également pu observer que de nombreuses études utilisent les algorithmes évolutionnaires, souvent hybridés avec des méthodes heuristiques afin d'explorer de façon plus efficace l'espace de recherche.

7. L'approche multicritères proposée pour l'optimisation de notre problème

Après l'obtention de toutes les solutions non dominées, nous illustrons dans cet algorithme notre approche multicritères basée sur les algorithmes génétiques, proposée pour l'optimisation de notre problème « MD-FSMVRPTW ».

L'objectif principal des algorithmes génétiques dans cette approche est de proposer une bonne approximation de la frontière de Pareto. Ceci, en explorant l'espace de solution, afin d'obtenir les populations les plus diverses et les plus possibles.

Début

Etape 1: Création de la population initiale de taille = n ;

Etape 2: Affectation de tous les nœuds (clients) aux dépôts correspondants « création des clusters »;

Tanque nombre de générations non atteint **Faire**

Début

Etape 3: Création de la population intermédiaire de taille = $2n$ avec de nouveaux individus de "croisement ou mutation"

Etape 4: Application des procédures d'ordonnancement et de correction de capacité; «Appel les algorithmes **Algo IV.2, Algo IV.3** »

Etape 5 : Création de la deuxième population intermédiaire de taille = $2n * 2n$ représentant les différentes routes prises par chaque véhicule;

Etape 6 : Déterminer la valeur de fitness pour chaque individu dans la population $P_{\text{nœuds/véhicules}}$ (Le nombre de véhicules, la somme des retards et le coût de transport) pour obtenir par la suite la nouvelle population $P_{\text{pareto-dominance}}$;

Etape 7 : Ordonner la population $P_{\text{pareto-dominance}}$ en fonction des valeurs de fitness minimales (Le nombre de véhicules, la somme des retards et le coût de transport);

Etape 8 : Copier les solutions non dominées;

Etape 9 : Incrémenter le nombre de générations;

FinTQ

Fin.

Algo V.1 Structure de l'algorithme génétique multicritères proposé pour l'optimisation de notre problème FSMVRPTW

8. Conclusion

En générale le problème de transport est présenté comme un problème multi-objectif dans le cas réel. Toutefois, l'aspect construction de tournées dans un contexte multi-objectif n'a pas été vraiment traité. L'ensemble de ces recherches bibliographiques sert à confirmer l'importance du problème de tournées de véhicules.

Pour cela, nous avons évoqué les problèmes multi-objectif. Nous avons pu constater que la plupart des études pratiques utilisent soit l'agrégation des critères pour avoir au final une unique solution, soit un Front Pareto qui permet de mettre en avant les solutions non-dominées. Cet ensemble est intéressant puisqu'il permet de garder différentes propositions pour le décideur. Ce Front est particulièrement pratique et est bien adapté notamment pour les études multi-objectif, sur lesquels les solutions non-dominées restent pertinentes et visuelles. Pour obtenir ce Front Pareto, différentes méthodes peuvent être utilisées, nous pouvons constater qu'en pratique les algorithmes génétiques sont révélés très efficaces.

Finalement, nous avons conçu notre algorithme pour l'optimisation du MD-FSMVRPTW, et ceci en se basant sur le principe des deux méthodes efficaces « Les algorithmes génétiques et la notion de cluster ».

CHAPITRE 6 : SIMULATIONS ET RÉSULTATS

1. Introduction

Dans ce chapitre, nous allons nous intéresser à la présentation et à l'implémentation des instances de tests. Nous présentons en premier lieu ces instances, il s'agit des benchmarks de Cordeau et al. [17]. Ensuite, nous allons tester notre approche (ProGenClust) en utilisant ces instances de tests. Les simulations effectuées et les résultats acquis seront donc exposés en dernier lieu.

Toutes les expériences ont été réalisées sur un système d'exploitation linux « distribution : Ubuntu », un processeur i5, une mémoire de 4Go et un disque dur de 500Go. Dans le but d'évaluer les performances de notre approche, nous avons comparé nos résultats de calcul avec ceux de la littérature existante. Notre algorithme a été programmé sous MATLAB R2010b et les résultats obtenus ont été présentés en utilisant l'outil GUIMOO version 0.2.

2. Instances de tests utilisés pour l'évaluation expérimentale

Notre expérience de calcul est basée sur les ensembles de données composés de (23) MDVRPTW benchmarks qui peuvent être téléchargés à partir du Web VRP sur le site web [<http://neo.lcc.uma.es/vrp/vrp-instances/multiple-depotvrp-instances/>]. Nous avons comparé nos résultats obtenus suite à l'utilisation de l'approche Pareto avec ceux obtenus par la méthode Gen-Clust développé par Thangiah et al. [118].

La performance du MDVRPTW est évaluée en utilisant deux ensembles de 20 MDVRP benchmarks différents "les sommets des dépôts et des clients sont définis par des points dans le plan cartésien, où le coût de chaque arête (i, j) est la distance euclidienne entre les clients i et j ". Les spécifications des MDVRPTW benchmarks sont indiquées dans la Table VI.1. Où :

- Q_t : la capacité du véhicule de type t, ($t = X, Y, Z, U, V, W$);
- F_t : représente le coût fixe du véhicule de type t ;
- V_t : représente le coût variable du véhicule de type t.
- Client: représente le nombre de clients variant de 50 à 360.
- Dépôt: représente le nombre de dépôts utilisés qui varie entre 2 et 9.

Table VI.1. Les spécifications des 20 benchmarks utilisés

Problème	Q _X	F _X	V _X	Q _Y	F _Y	V _Y	Q _Z	F _Z	V _Z	Q _U	F _U	V _U	Q _V	F _V	V _V	Q _W	F _W	V _W
3	20	20	1.0	30	35	1.0	40	50	1.0	40	120	1.0	120	225	1.0			
4	60	1000	1.0	80	1500	1.0	150	3000	1.0									
5	20	20	1.0	30	35	1.0	40	50	1.0	70	120	1.0	120	225	1.0			
6	60	1000	1.0	80	1500	1.0	150	3000	1.0									
13	20	20	1.0	30	35	1.1	40	50	1.2	70	120	1.7	120	225	2.5	200	400	3.2
14	120	100	1.0	160	1500	1.1	300	3500	1.4									
15	50	100	1.0	100	250	1.6	160	450	2.0									
16	40	100	1.0	80	200	1.6	140	400	2.1									
17	50	25	1.0	120	80	1.2	200	150	1.5	350	320	1.8						
18	20	10	1.0	50	35	1.3	100	100	1.9	150	180	2.4	250	400	2.9	400	800	3.2
19	100	500	1.0	200	1200	1.4	300	2100	1.7									
20	60	100	1.0	140	300	1.7	200	500	2.0									

Chaque problème est présenté sous forme d'un fichier qui comprend : l'identifiant de chaque nœud, ces coordonnées, leurs fenêtres de temps, leur temps de service, les quantités à transporter, la capacité du transport, le type de véhicule, les contraintes de précedence et de succession.

Ces paramètres sont définis dans trois classes comme suit :

- La classe qui définit les paramètres de chaque client :
 - Id: identifiant du client ;
 - (x, y): coordonnées du client ;
 - q: quantités à transporter au client ;
 - (e, l): fenêtres de temps du client ;
 - s: temps de service d'un client ;
 - pred (i): prédécesseur du client i ;
 - succ (i): successeur du client i.
- La classe qui définit le dépôt :
 - (x, y): coordonnées du dépôt ;
 - (e, l): fenêtres de temps du dépôt.
- La classe qui définit les véhicules :
 - T: le type du véhicule ;
 - Q (t): la capacité maximal d'un véhicule de type T ;
 - V (t): la vitesse d'un véhicule de type T ;
 - A (t): Nombre de véhicules disponibles (type T).

3. Résultats expérimentaux

Dans cette section, nous présentons les meilleurs résultats de la littérature existante et la moyenne de nos résultats obtenus sur les vingt benchmarks. Pour obtenir un bon réglage des paramètres de l'Algorithme Génétique, plusieurs facteurs doivent être pris en considération comme la taille de la population, le schéma de codage, les critères de sélection, les probabilités des opérateurs génétiques et les techniques d'évaluation (conditionnement physique). Chaque optimisation basée sur les algorithmes génétiques a été effectuée avec des réglages de paramètres similaires, comme le montre la Table VI.2.

Table VI.2. Les paramètres expérimentaux

Les paramètres	Réglage
Taille de la population	300
Initialisation du chromosome	aléatoire
Nombre d'élite	3(1%)
Probabilité de croisement	0.60
Probabilité de la mutation intra- dépôt	0.20
Probabilité de la mutation inter- dépôt	0.25
Taux de tentative de mutation inter- dépôt	Chaque 10 générations

Dans les différents travaux liés au MDVRPTW trouvés dans la littérature, la plupart des chercheurs se concentrent sur un objectif, en particulier sur celui qui minimise la distance totale parcourue, en ignorant le nombre de véhicules utilisés et la somme totale des retards. Cependant, nous pouvons trouver des situations pratiques où la meilleure solution est celle avec un ou plusieurs véhicules de moins et un délai d'attente plus court que la meilleure solution trouvée, mais la distance totale parcourue est légèrement plus grande comme indiqué dans [118]. Les solutions devraient être préférées si elles nécessitent moins de véhicules, des délais plus courts et une distance minimale parcourue.

Pour de tels besoins, nous avons été motivés à utiliser l'approche de classement de Pareto (Pareto Ranking) pour avoir une meilleure base pour comparer la qualité de la solution de l'algorithme proposé avec celle de [118] qui a interprété le MDVRPTW comme un problème multi-objectif en considérant la minimisation de la distance totale parcourue, le nombre de véhicules utilisés et la somme totale des retards. Contrairement aux méthodes à objectif unique, lors de l'utilisation de l'approche Pareto Ranking, nous avons le choix entre plus d'une solution, selon ce que l'utilisateur souhaite : les solutions avec un nombre minimum de véhicules utilisés, un minimum de retard ou un coût de voyage minimum.

Le choix des résultats obtenus se fait à l'aide d'un outil nommé GUIMOO « <http://guimoo.gforge.inria.fr/> » qui permet une visualisation claire du front de Pareto. Ce dernier est donné par la (Fig. VI.1).

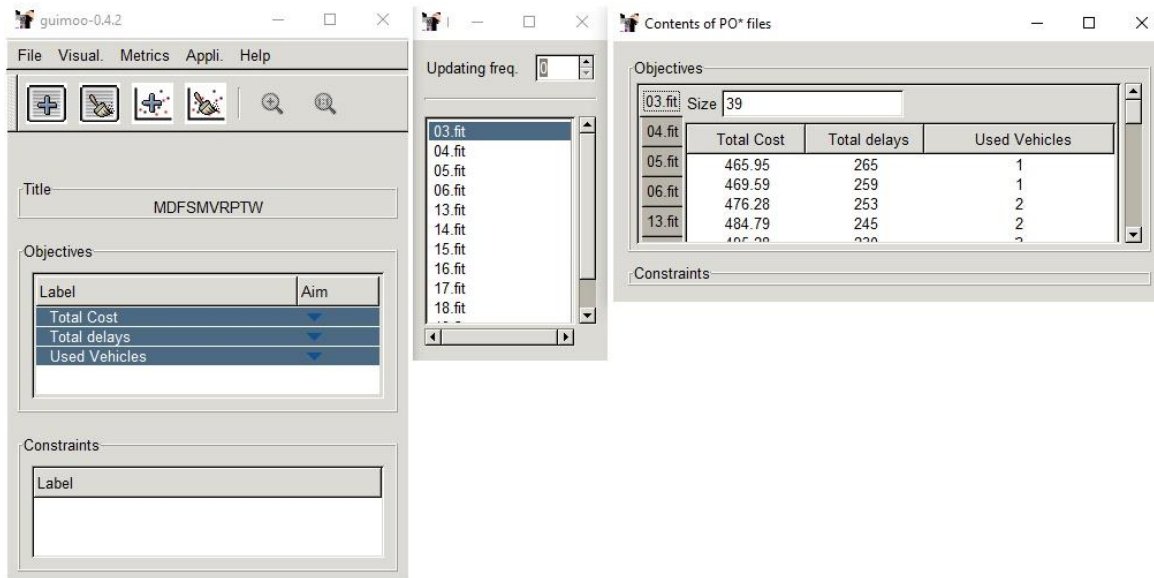
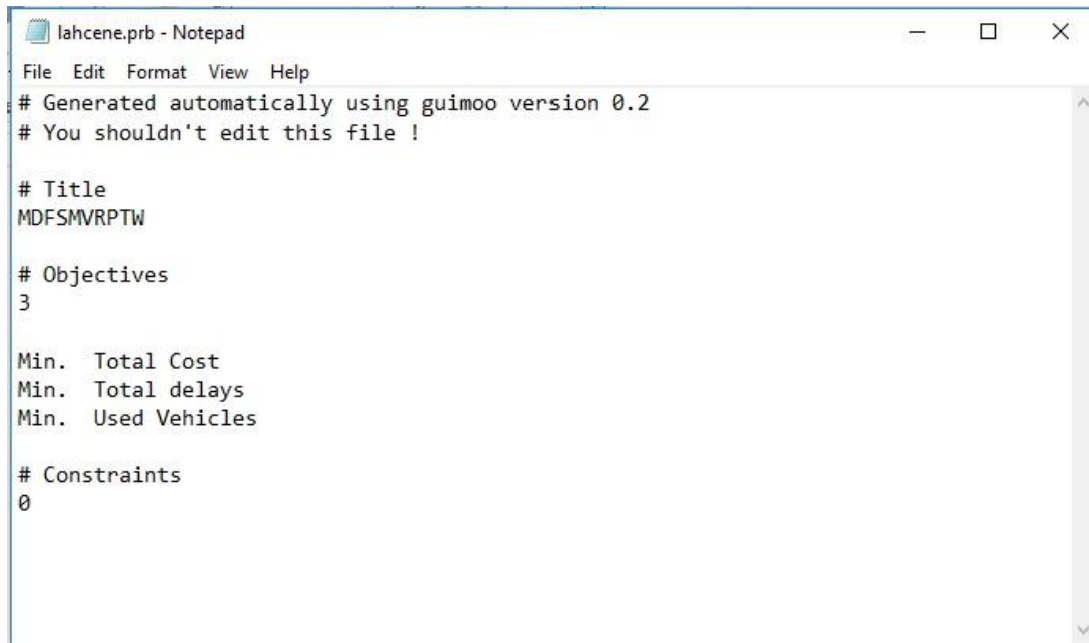


Fig. VI.1 Interface de GUIMOO

Nous avons commencé par créer un fichier extension (.prb), dans lequel nous avons défini le nombre d'objectifs et le type (min, max), (voir la (Fig. VI.2)). Après, nous avons créé pour chaque instance un fichier extension (.fit), afin d'introduire les résultats obtenus de l'optimisation multicritères des différents benchmarks (voir la (Fig. VI.3)). A la fin, la visualisation de la frontière de Pareto est donnée dans la (Fig. VI.4).



```

lahcene.prb - Notepad
File Edit Format View Help
# Generated automatically using guimoo version 0.2
# You shouldn't edit this file !

# Title
MDFSMVRPTW

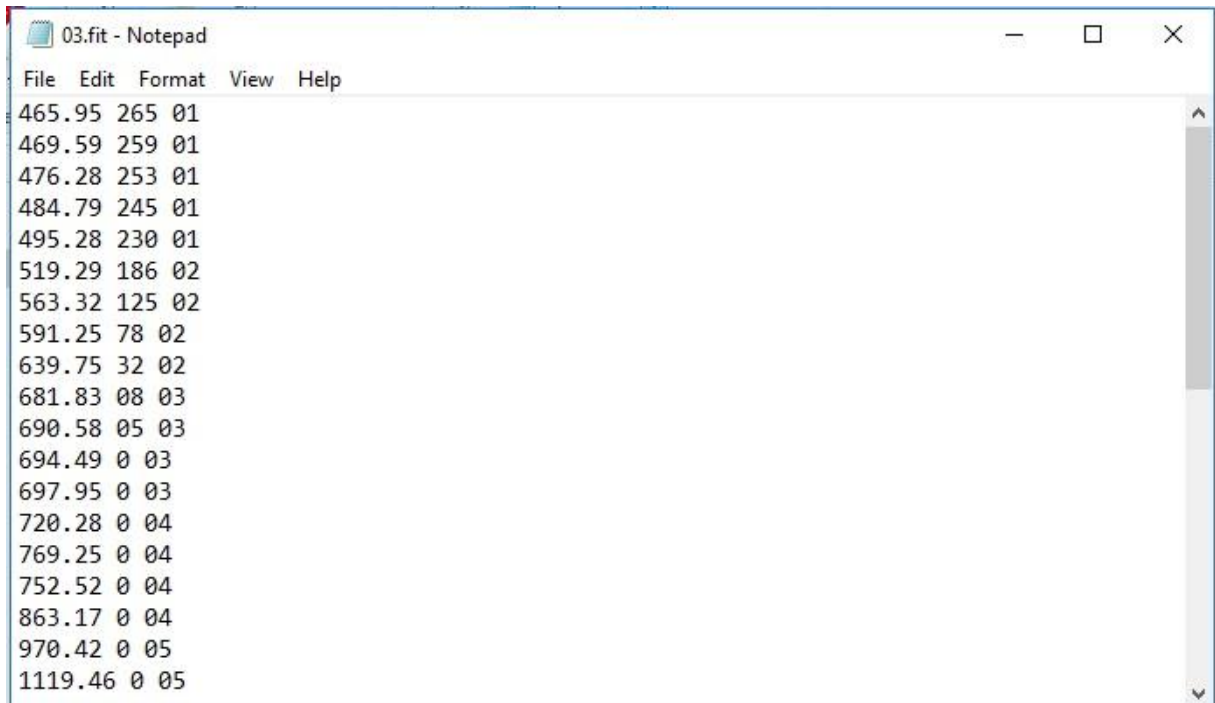
# Objectives
3

Min. Total Cost
Min. Total delays
Min. Used Vehicles

# Constraints
0

```

Fig. VI.2 Création d'un fichier pour le problème



```

03.fit - Notepad
File Edit Format View Help
465.95 265 01
469.59 259 01
476.28 253 01
484.79 245 01
495.28 230 01
519.29 186 02
563.32 125 02
591.25 78 02
639.75 32 02
681.83 08 03
690.58 05 03
694.49 0 03
697.95 0 03
720.28 0 04
769.25 0 04
752.52 0 04
863.17 0 04
970.42 0 05
1119.46 0 05

```

Fig. VI.3 Création un fichier des données

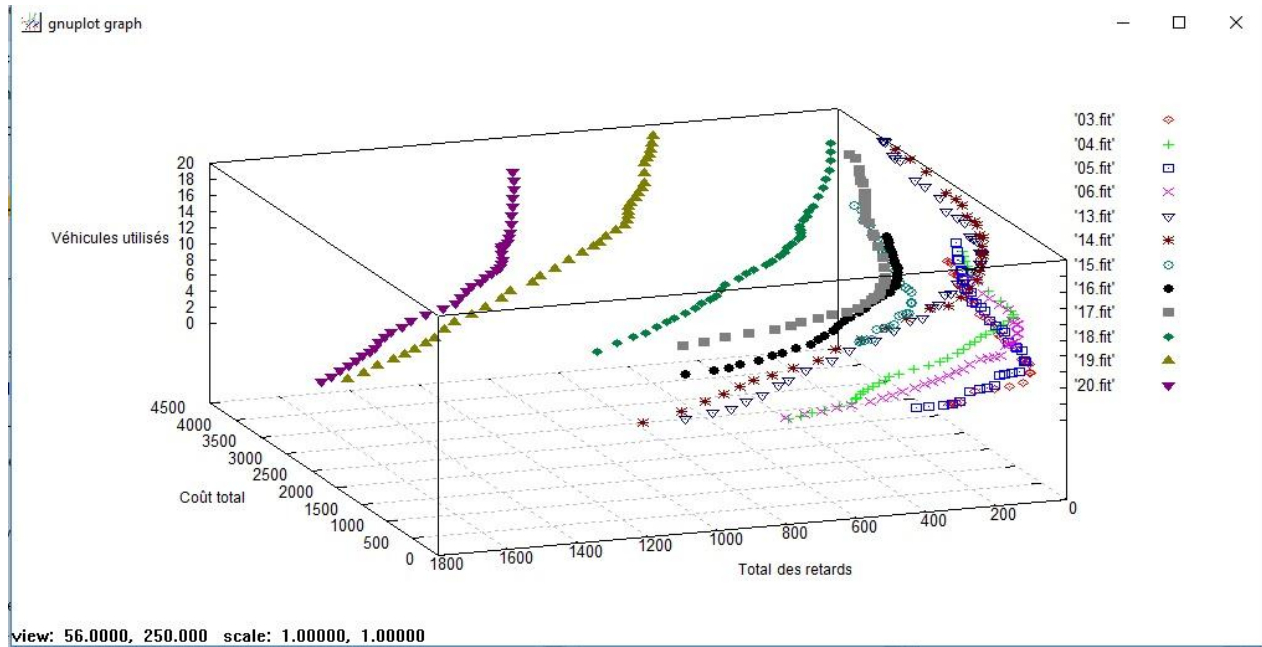


Fig. VI.4 Front de Pareto

Dans la figure (Fig. VI.4), chaque courbe représente les résultats obtenus suite à une optimisation multicritères basée sur une des instances Benchmark[17], l'ensemble de ces solutions sont réparties sur des Fronts Pareto dans un espace à 3 dimensions où chaque axe représente un des objectifs à minimiser. Si nous observons bien les fronts Pareto, on remarque que pour la majorité des instances on a obtenu des solutions dominées comme c'est le cas pour les instances 05, 06, 13, 14, 15 et 17.

Dans notre cas, nous avons utilisé le principe des méthodes à posteriori, qui ont comme avantage de ne pas avoir à fixer un compromis entre les différents objectifs avant l'exécution, et de nous proposer à la fin un nombre suffisant de solutions afin que nous puissions choisir le compromis que nous souhaitons. Les solutions (routes) optimales sélectionnées ont été représentées dans le tableau (Table VI.3).

Table VI.3. Les routes optimales résultantes de la simulation basée sur notre approche ProGenClust

Dépôt	Nombre véhicules disponibles	Nombre clients servis	Nombre de véhicules utilisés	Route optimale
Instance P03				
A(20,20)	3	10	2	A 4 34 46 67 A 26 17 51 6 68 75 A
B(30,40)	3	16	2	B 5 37 20 70 60 71 69 36 47 B 48 21 74 2 30 45 29 B
C(50,30)	3	21	3	C 35 8 52 27 15 57 13 54 19 C 14 59 66 65 11 C 7 58 72 31 10 38 53 C
D(60,50)	3	14	2	D 40 12 39 9 32 D 3 44 50 25 55 18 24 49 16 D
E(10,40)	3	14	2	E 43 41 56 23 63 33 73 1 E 62 28 61 22 64 42 E
Instance P04				
A(15,30)	8	50	8	A 41 23 67 39 56 A 96 6 89 27 28 53 58 A 92 61 16 86 38 43 15 A 57 42 14 44 91 100 98 37 97 A 21 72 74 75 22 2 A 73 4 25 55 54 12 26 40 A 59 99 93 85 A 87 95 94 13 A
B(50,20)	8	50	7	B 69 76 77 68 80 24 29 3 B 51 9 35 71 65 66 70 B 8 45 17 84 5 60 83 18 52 B 63 64 49 36 47 46 82 B 33 81 34 78 79 50 1 B 88 7 48 19 11 62 B 31 10 90 32 20 30 B
Instance P05				
A(20,40)	5	52	4	A 89 27 69 1 70 30 32 90 63 64 49 36 47 46 45 A 60 6 94 13 95 97 92 37 100 98 93 59 99 96 A 18 52 7 88 31 10 62 11 19 48 82 8 83 A 5 61 85 91 16 44 14 38 86 17 84 A
B(22,50)	5	48	4	B 76 50 33 81 9 51 20 66 65 71 35 34 78 29 24 B 77 3 79 68 80 B 12 28 53 58 2 57 87 42 43 15 41 22 73 21 40 26 B 4 72 74 75 56 23 67 39 25 55 54 B
Instance P06				
A(40,40)	6	34	6	A 85 A 91 44 38 86 16 A 93 59 98 100 A 37 97 87 2 57 15 43 42 14 A 92 95 13 94 6 96 99 A 5 60 83 8 45 17 84 61 A
B(10,25)	6	25	4	B 56 23 67 39 B 54 80 68 29 24 55 25 4 B 72 75 41 22 74 73 21 B 26 12 28 53 58 40 B
C(30,10)	6	41	6	C 51 9 71 65 35 34 78 50 C 62 19 47 48 C 70 30 20 66 32 10 C 33 81 79 3 77 76 C 88 7 82 46 36 49 64 11 63 90 C 31 52 18 89 27 69 1 C
Instance P13				
A(20,20)	5	40	4	A 3 11 19 27 35 34 26 18 10 2 A 8 16 24 37 29 21 13 5 A 7 15 23 31 39 65 73 38 30 22 14 6 A 1 9 17 25 33 36 28 20 12 4 A
B(30,50)	5	40	4	B 41 49 57 76 68 60 52 44 B 45 53 61 69 77 80 72 64 56 48 B 42 50 58 66 74 32 40 75 67 59 51 43 B 47 55 63 71 79 78 70 62 54 46 B

Instance P14				
A(23,45)	5	40	4	A 4 12 20 28 36 33 25 17 9 1 A 6 14 22 30 38 73 31 23 15 7 A 3 11 19 27 35 34 26 18 10 2 A 8 16 24 32 40 37 29 21 13 5 A
B(35,30)	5	40	4	B 48 56 64 72 80 79 71 63 55 47 B 42 50 58 66 74 39 65 57 49 41 B 43 51 59 67 75 77 69 61 53 45 B 44 52 60 68 76 78 70 62 54 46 B
Instance P15				
A(40,40)	5	39	4	A 3 11 19 27 34 26 18 10 2 A 1 9 17 25 33 36 28 20 12 4 A 15 23 31 39 74 66 58 50 49 57 65 73 38 30 22 14 A 6 7 8 5 A
B(50,22)	5	41	4	B 53 61 69 77 116 108 100 92 89 97 105 113 75 67 59 51 B 41 42 43 45 B 48 56 64 72 118 80 79 71 63 55 47 B 46 54 62 70 78 76 68 60 52 44 B
C(55,55)	5	39	4	C 86 94 102 110 119 111 103 95 87 C 83 82 81 84 C 85 93 101 109 117 120 112 104 96 88 C 90 98 106 114 159 151 143 135 136 144 152 160 115 107 99 91 C
D(25,45)	5	41	4	D 124 126 127 128 00 132 140 148 156 37 29 21 13 16 24 32 40 158 150 142 134 D 123 131 139 147 155 157 149 141 133 125 D 121 129 137 145 35 153 154 146 138 130 122 D
Instance P16				
A(15,35)	5	37	4	A 5 13 21 29 37 156 145 153 35 27 19 11 3 A 1 9 17 25 33 34 26 18 10 2 A 6 14 22 36 28 20 12 4 A 8 16 31 23 15 7 A
B(35,25)	5	43	4	B 43 51 59 67 75 113 105 97 69 61 53 45 B 41 49 57 65 73 38 30 39 74 66 58 50 42 B 47 55 63 71 79 64 56 48 B 46 54 62 70 78 76 68 60 52 44 B
C(30,50)	5	37	4	C 86 94 102 110 118 80 72 77 116 108 100 92 84 C 88 96 104 112 120 119 111 103 95 87 C 81 89 106 98 90 82 C 83 91 99 117 109 101 93 85 C
D(20,20)	5	43	4	D 126 134 142 150 158 40 32 24 148 140 132 124 D 122 130 138 146 154 137 129 121 D 125 133 141 149 157 155 147 139 131 123 D 127 135 143 151 159 114 107 115 160 152 144 136 128 D
Instance P17				
A(20,10)	5	40	4	A 1 9 17 25 33 36 28 20 12 4 A 7 15 23 31 73 38 30 22 14 6 A 2 10 18 26 34 35 27 19 11 3 A 5 13 21 29 37 156 32 24 16 8 A
B(30,35)	5	40	4	B 47 55 63 71 79 80 72 64 56 48 B 46 54 62 70 78 76 68 60 52 44 B 42 50 58 66 74 39 65 57 49 41 B 45 53 61 69 77 116 67 59 51 43 B
C(40,40)	5	40	4	C 86 94 102 110 118 119 111 103 95 87 C 82 90 98 106 114 159 107 99 91 83 C 84 92 100 108 75

				113 105 97 89 81 C 85 93 101 109 117 120 112 104 96 88 C
D(32,45)	5	40	4	D 128 136 144 152 160 115 151 143 135 127 D 126 134 142 150 158 40 148 140 132 124 D 121 129 137 145 153 154 146 138 130 122 D 125 133 141 149 157 155 147 139 131 123 D
Instance P18				
A(15,20)	5	31	3	A 6 14 22 30 73 38 36 28 20 12 4 A 2 3 5 8 7 A 10 18 26 34 239 231 214 222 230 238 33 25 17 9 1 A
B(50,20)	5	55	5	B 56 64 72 80 118 110 102 94 92 100 108 116 77 69 61 53 B 41 49 57 65 76 68 60 52 44 B 42 43 45 48 B 47 55 63 71 79 78 70 62 54 46 B 50 58 66 74 39 31 23 15 16 24 32 40 75 67 59 51 B
C(30,30)	5	25	3	C 86 84 81 82 C 88 96 104 112 120 119 111 103 95 87 C 83 91 99 107 160 115 117 109 101 93 85 C
D(22,45)	5	61	5	D 135 143 151 159 114 106 98 90 89 97 105 113 158 150 142 134 D 128 136 144 152 157 149 141 133 125 D 127 126 124 121 122 D 132 140 148 156 37 29 21 13 11 19 27 35 153 145 137 129 D 130 138 146 154 199 191 176 184 192 200 155 147 139 131 123 D
E(35,45)	5	33	3	E 167 168 175 183 166 164 E 161 169 177 185 235 193 194 186 178 170 162 E 163 171 179 187 195 197 189 181 173 165 E
F(10,10)	5	35	3	F 213 221 229 237 196 188 180 172 174 182 190 198 240 232 224 216 F 201 209 217 225 233 236 228 220 212 204 F 203 211 219 227 234 226 218 210 202 F
Instance P19				
A(15,30)	5	40	4	A 3 11 19 27 35 153 145 156 37 29 21 13 5 A 2 10 18 26 34 239 230 238 33 25 17 9 1 A 6 14 22 36 28 20 12 4 A 8 16 31 23 15 7 A
B(35,23)	5	43	4	B 46 54 62 70 78 76 68 60 52 44 B 48 56 64 79 71 63 55 47 B 43 51 59 67 75 113 105 97 69 61 53 45 B 42 50 58 66 74 39 30 38 73 65 57 49 41 B
C(25,25)	5	40	4	C 88 96 104 112 120 119 111 103 95 87 C 85 93 101 109 117 99 91 83 C 82 90 98 106 89 81 C 84 92 100 108 116 77 72 80 118 110 102 94 86 C
D(10,35)	5	45	4	D 127 135 143 151 159 114 107 115 160 152 144 136 128 D 123 131 139 157 149 141 133 125 D 121 129 137 146 138 130 122 D 126 134 142 150 158 40 32 24 148 140 132 124 D

E(50,10)	5	48	4	E 167 175 183 191 199 154 147 155 200 192 184 176 168 E 163 171 179 187 195 197 189 181 173 165 E 161 169 177 194 186 178 170 162 E 166 174 182 190 198 240 232 224 188 180 172 164 E
F(35,45)	5	24	3	F 201 209 217 225 233 234 226 218 210 202 F 206 214 222 236 228 220 212 204 F 208 216 231 223 215 207 F
Instance P20				
A(20,20)	5	40	4	A 1 9 17 25 239 34 26 18 10 2 A 6 14 22 30 38 36 28 20 12 4 A 8 16 24 32 74 39 31 23 15 7 A 3 11 19 27 35 153 29 21 13 5 A
B(40,40)	5	40	4	B 48 56 64 72 80 118 69 61 53 45 B 43 51 59 67 75 40 66 58 50 42 B 41 49 57 65 73 76 68 60 52 44 B 46 54 62 70 78 79 71 63 55 47 B
C(25,30)	5	40	4	C 86 94 102 110 77 116 108 100 92 84 C 88 96 104 112 120 119 111 103 95 87 C 83 91 99 107 115 117 109 101 93 85 C 81 89 97 105 159 114 106 98 90 82 C
D(35,20)	5	45	4	D 123 131 139 147 199 154 146 138 130 122 D 121 129 137 145 37 156 148 140 132 124 D 126 134 142 150 158 113 151 143 135 127 D 125 133 141 149 157 160 152 144 136 128 D
E(60,50)	5	45	4	E 163 171 179 187 195 197 189 181 173 165 E 161 169 177 185 193 194 186 178 170 162 E 166 174 182 190 237 196 188 180 172 164 E 168 176 184 192 200 155 191 183 175 167 E
F(60,35)	5	30	3	F 203 211 219 227 235 234 226 218 210 202 F 201 209 217 225 233 236 228 220 212 204 F 206 214 222 230 238 33 231 223 215 207 F

Les routes planifiées sont optimisées à l'aide de notre approche « ProGenClust » pour les douze instances et le nombre de clients servis, le nombre de véhicules utilisés, la longueur de la route et l'itinéraire optimal sont évalués. Ces paramètres sont présentés dans la Table VI.3, comme douze instances de références (p03, p04, p05, p06, p13, p14, p15, p16, p17, p18, p19 et p20). La figure ci-dessous « Fig. VI.5 » illustre un exemple d'optimisation de l'instance P04 par notre approche « ProGenClust », où, à partir des positions initiales des clients et des dépôts, tous les nœuds représentant des clients ont été affectés au plus proche dépôt en termes de distance euclidienne (Clustering), ce qui permet de former à la fin des clusters. Ensuite nous procédons à la phase de routage et d'ordonnancement, où les tournées traversent tous les emplacements des clients (affichés en points) à partir d'un lieu source « dépôt » et se terminent à la même source après avoir servi tous les clients.

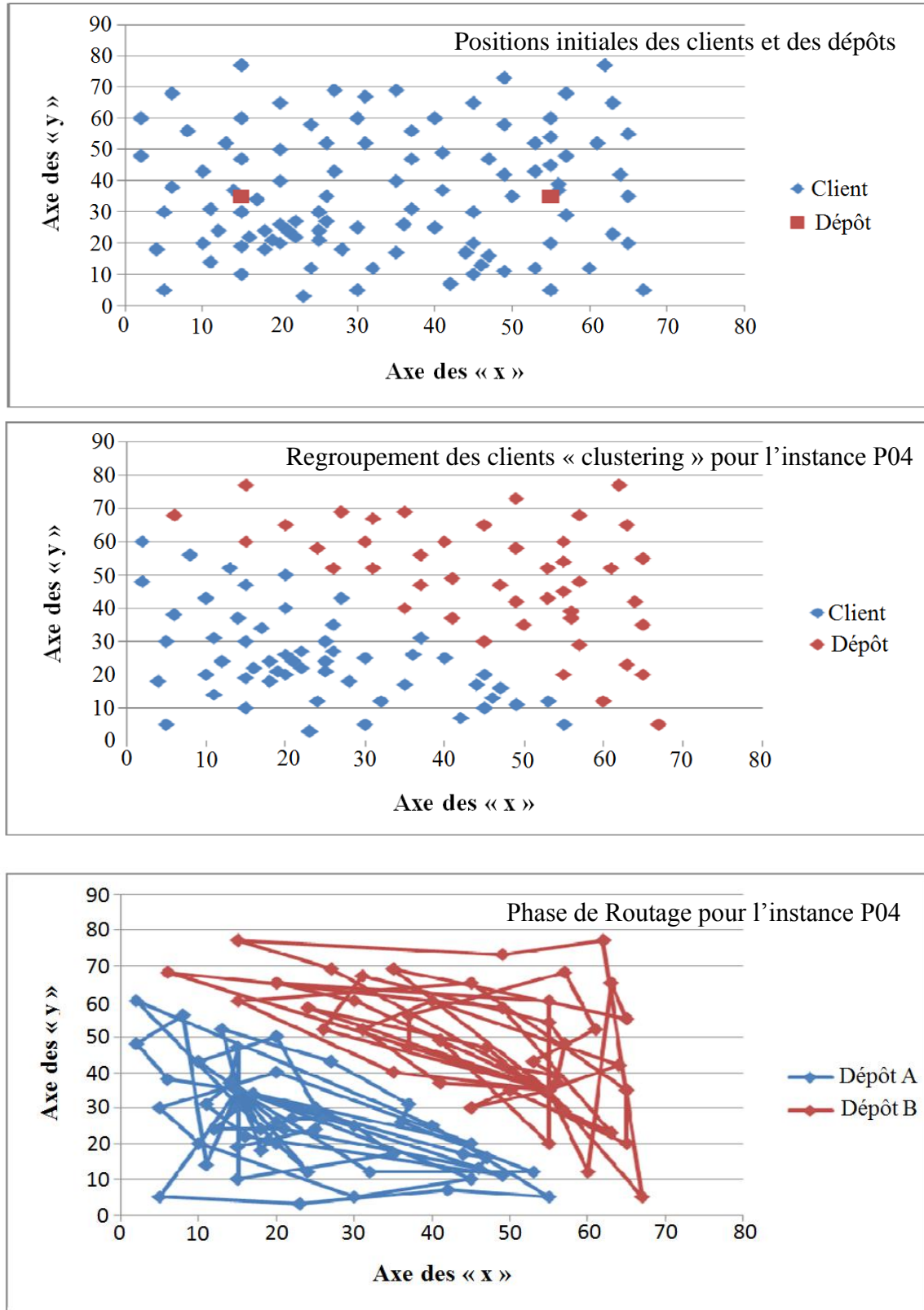


Fig. VI.5 Exemple de routes optimisées utilisant ProGenClust pour l'instance p04

Table VI.4. Moyenne des résultats pour FSMVRP avec coût fixe pour 12 problèmes de tests

Problèmes	Taillard (1999)		Gendreau et al. (2006)		Lima et al. (2010)		Choi and Tcha (2014)		ProGenClust (2017)		Véhicules utilisés
	Coût	Temps (s)	Coût	Temps (s)	Coût	Temps (s)	Coût	Temps (s)	Coût	Temps (s)	
3	694.49	N/A	694.49	164	696.15	89	694.49	0	694.49	0	03
4	1053.83	N/A	1056.01	253	1054.33	85	1056.01	1	1053.83	0	08
5	773.59	N/A	773.72	164	774.90	85	772.84	1	773.56	3	04
6	896.47	N/A	897.98	309	896.47	85	896.84	0	910.71	5	07
13	1340.78	470	1332.88	724	1318.95	559	1326.77	10	1318.95	22	06
14	1370.60	570	1368.98	1033	1368.62	669	1366.13	51	1368.52	45	14
15	2578.61	334	2575.68	901	2575.20	554	2573.92	10	2576.62	68	04
16	2616.25	349	2615.96	815	2601.36	507	2609.08	11	2601.36	110	06
17	2905.74	2072	2904.29	1022	2922.30	1517	2898.26	207	2897.26	99	05
18	4007.80	2744	4017.57	691	4026.00	1613	4000.78	70	4001.51	179	07
19	3956.40	12528	3973.50	1687	3986.30	2900	3956.08	1179	3958.24	678	12
20	4111.18	2117	4128.20	1421	4190.30	2383	4105.33	264	4097.06	1021	08
Moyenne	2192.15	2648	2194.94	765	2200.91	920	2196.38	150	2187.42	186	07
Déviaton	0.38%		0.40%		0.42%		0.41%		0.36%		
Nombre Des meilleures solutions	03		01		03		05		06		

La Table VI.4 présente les meilleurs résultats déjà obtenus pour la variante FSMVRP avec un coût fixe sur les 12 instances. Le coût de la solution et le temps de calcul pour chaque instance sont inclus pour chaque algorithme. La Table VI.4 présente aussi le pourcentage d'écart entre le résultat de chaque algorithme et les résultats des meilleures solutions connues dans la littérature. Les cinq algorithmes sont avérés très précis avec un écart moyen inférieur à 1% pour les 12 instances. Nous remarquons qu'en moyenne, notre algorithme génétique (ProGenClust) produit de meilleurs résultats (6/12 meilleures résultats atteintes) que ceux de Lima et al. [133], Taillard [134], et Gendreau et al. [27], et des résultats aussi bons que ceux de Choi et Tcha [22]. Les temps de calcul de chaque algorithme ont été rapportés par ces auteurs et présentés dans cette même table.

Maintenant, nous discutons la performance de l'algorithme génétique proposé en utilisant la Procédure de classement de Pareto (Pareto Ranking) par rapport à l'algorithme génétique « GenClust » proposé par Thangiah et al., [118] et résumons les résultats dans la Table VI.5 en utilisant un format similaire à celui trouvé dans [118]. Les coûts d'itinéraire sont mesurés par la distance euclidienne moyenne. La colonne intitulée GenClust donne les meilleures solutions publiées dans [118] tandis que la colonne ProGenClust donne les meilleures solutions obtenues par notre algorithme génétique après 10 exécutions.

Table VI.5. Comparaison de notre algorithme génétique « ProGenClust » avec l'algorithme génétique connu « GenClust » [118]

Inst.	Client.	Dépôt	GenClust(km)	ProGenClust (km)	C_M	V_M	T_M
3	75	5	689.15	694.49	-2.7	+1	-4.5
4	100	2	1062.38	1053.83	-0.1	0	-1.6
5	100	2	754.84	773.56	+1.1	0	+2.9
6	100	3	976.02	910.71	-2.3	-1	-8.5
13	80	2	1318.95	1318.95	0.0	0	0
14	80	2	1360.12	1368.52	+0.5	0	+1.9
15	160	4	3059.15	2576.62	-3.6	+1	-2.4
16	160	4	2719.98	2601.36	-1.6	0	-4.7
17	160	4	2894.69	2899.26	-0.2	0	-2.5
18	240	6	5462.90	4001.51	-9.8	+2	-0.5
19	240	6	3956.61	3958.24	+0.6	0	+2.3
20	240	6	4344.81	4097.06	-5.4	-3	-18.6
Nombre des meilleures solutions			4	5			

La colonne intitulée V_M présente la différence entre le nombre de véhicules utilisés par notre approche ProGenClust et l'approche GenClust. Un nombre négatif indique une diminution, un nombre positif indique une augmentation. Notre Algorithme génétique a obtenu le même nombre de véhicules que GenClust [118] dans sept problèmes, une réduction de 1-3 véhicules dans deux problèmes et une augmentation de 1-2 véhicules dans 3 problèmes.

La colonne C_M représente les écarts de distance en termes de pourcentages entre l'algorithme génétique proposé et GenClust. Une valeur négative signifie que notre algorithme génétique a eu un coût inférieur, et une valeur positive signifie que notre algorithme génétique a obtenu un coût plus élevé que GenClust. Notre algorithme génétique a réduit les coûts dans huit problèmes, avec une réduction maximale des coûts de (-9,8%). Cependant, nous pouvons noter que l'algorithme génétique proposée trouve de meilleures solutions que Genclust dans (6 et 20), où il a une réduction des distances et du nombre de véhicules en même temps. En outre, il convient de noter que dans l'instance 13, notre algorithme a eu le même coût que GenClust pour le même nombre de véhicules.

La dernière colonne T_M donne la somme totale des retards en termes de pourcentages entre l'algorithme génétique proposé et GenClust. Une valeur négative signifie que notre algorithme a eu un délai plus court et une valeur positive signifie qu'il a obtenu un délai plus long que GenClust. Notre algorithme génétique a obtenu un retard plus court dans huit problèmes par rapport à GenClust [118].

Les valeurs en gras dans les colonnes intitulées GenClust et ProGenClust indiquent les cas où un algorithme dépasse un autre en minimisant la distance totale parcourue (avec le même nombre de véhicules et un délai plus court pour les deux algorithmes

ProGenClust et GenClust) où il a minimisé le coût, le nombre de véhicules ou la somme totale des retards.

Enfin, après avoir comparé les performances de notre algorithme génétique avec ceux de GenClust [118] sur les différents MDVRPTW benchmarks en termes de distance parcourue, de nombre de véhicules et de somme totale de retard, nous constatons que notre approche a révélé cinq nouveaux meilleurs résultats par rapport au travail effectué par [118].

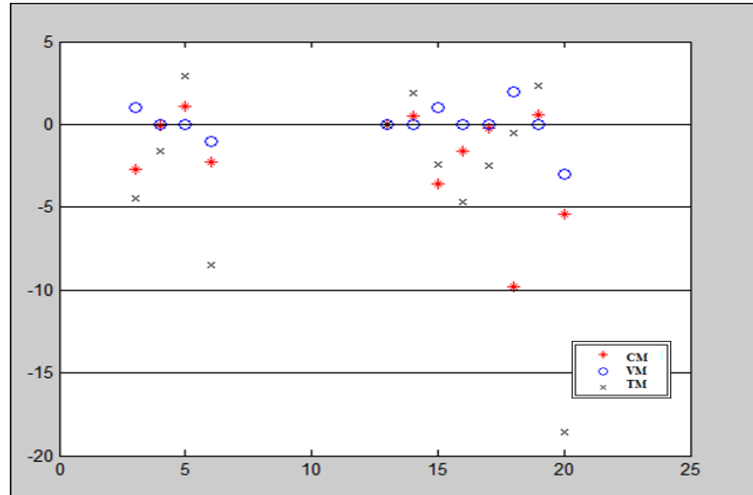


Fig. VI.6 Le nuage de points représentant les écarts entre les résultats de ProGenClust et GenClust [118]

La figure (Fig. VI.6) montre le nuage de points qui représente les différences entre les résultats de notre algorithme génétique par rapport à l'une des solutions les plus connues dans la littérature "GenClust [118]" en termes de distance optimale, somme totale moyenne des retards et le nombre de véhicules utilisés.

Nous notons qu'il existe trois types de points (métriques) dans la Figure (Fig. VI.6). Les points bleus (○) représentent la différence entre le nombre de véhicules utilisés (V_M) par notre ProGenClust approche et GenClust, les points rouges (✱) représentent les écarts de distance (C_M) en termes de pourcentage entre ProGenClust et GenClust et les points gris (×) représentent la somme totale des retards (T_M) en termes de pourcentage entre ProGenClust et GenClust.

Les points situés au-dessous de l'axe des x de la figure (Fig. VI.6) représentent les cas où notre approche est meilleure que l'algorithme GenClust, en revanche ceux situés au-dessus de l'axe illustrent des échantillons où l'approche GenClust est légèrement meilleure que la nôtre, et les points situés sur l'axe représentent des instances où les deux approches sont similaires.

La figure (Fig. VI.6) montre clairement que le nombre de points où notre approche est meilleure que GenClust (16 points) est supérieur au nombre de points où GenClust est meilleur (9 points), ce qui reflète l'efficacité de l'algorithme proposé.

4. Conclusion

Dans ce chapitre, nous avons présenté les résultats des simulations générés par notre approche « ProGenClust » pour le MDFSMVRPTW. Ces derniers sont basés sur trois approches de base : les algorithmes génétiques, la notion de Clustering (regroupement) et l'approche de classement Pareto (pareto ranking).

Après avoir effectués différents tests et évaluations sur notre approche, les résultats de simulation obtenus ont été répartis sur des Fronts Pareto représentés par l'outil GUIMOO. Les résultats sélectionnés ont été récapitulés sous forme de tableau et comparés aux résultats des solutions les plus connues dans la littérature.

Sur les 12 instances de Cordeau et al. [17], nous notons que notre approche a atteint (06 fois) les résultats des meilleurs solutions connues et trouve aussi 02 meilleurs résultats jamais atteints. Il offre également une performance très compétitive en termes de coût moyen de la solution et de temps moyen.

Ces approches donnent de bons résultats, principalement au niveau de la minimisation de la somme des retards où nous avons obtenu, dans plusieurs cas, un retard presque nul.

CONCLUSION GÉNÉRALE & PERSPECTIVES

Dans le but de faciliter la tâche de distribution des produits dans un réseau routier, tel que celui de l'Algérie, nous avons proposé dans ce travail de la modéliser sous forme d'un problème de tournées de véhicules (VRP), et pour éviter la mauvaise gestion du trajet qui est essentiellement due à la mauvaise prise de décision par le conducteur, nous avons opté pour la conception et la réalisation d'un système d'information d'aide à la décision (SIAD) permettant de sélectionner les meilleurs chemins à prendre par les véhicules lors de leurs tâches de distribution. Notre système se sert des cartes du GPS pour tracer les chemins possibles et définir les positions des véhicules.

Dans cette thèse, nous avons commencé par présenter un état de l'art sur les problèmes relatifs à l'optimisation dans les sciences de l'ingénieur en général. Ensuite, nous avons cité les solutions existantes des problèmes d'optimisation multi-objectif et leurs intérêts dans le domaine logistique et les méthodes d'optimisation multi-objectif basées sur la dominance Pareto. Cette étude a fait l'objet d'une recherche bibliographique approfondie.

Ceci nous a permis de positionner la problématique selon deux approches. Dans la première approche, le problème multi-objectif est souvent ramené à la résolution d'un problème mono-objectif, avec le risque toutefois d'éluder l'intérêt du problème d'optimisation multi-objectif. La deuxième approche adopte quant à elle un point de vue global, en prenant en compte l'ensemble des critères d'optimisation en utilisant la notion de dominance au sens de Pareto. Cette dernière a été retenue.

Avant la résolution de la problématique posée, nous avons formulé les développements mathématiques inhérents à notre problème d'optimisation multi-objectif basé sur la variante MD-FSMVRPTW caractérisé par : la présence de plusieurs dépôts, une flotte composée de véhicules hétérogènes et un intervalle de temps à être respecté, ... Cette formulation mathématique comprend toutes les conditions de routage, ainsi que les relations des fonctions « objectif », les contraintes liées au type, la capacité des véhicules utilisés, l'intervalle de temps, ...

Ensuite, nous avons présenté un domaine de recherche relativement récent qui s'intéresse à l'optimisation multi-objectif des problèmes d'optimisations dans le cas du problème de tournée de véhicules. Dans notre cas, nous avons utilisé une technique inspirée d'une hybridation entre les algorithmes génétiques et deux algorithmes de voisinages afin d'appliquer la notion de clustering, dont le but est d'améliorer la phase de génération de la population initiale, ainsi que d'avoir un meilleur ordonnancement. Nous avons aussi appliqué une approche de résolution multicritères au sens Pareto optimal, que nous avons implémenté sous MATLAB. Notre démarche nous a permis de mettre en évidence la facilité d'exploitation de ce genre d'algorithmes de résolution et leur efficacité dans le traitement des problèmes complexes après avoir défini leurs opérateurs génétiques tels que le codage des individus, la sélection basée sur la notion de

regroupement, le croisement et la mutation, notamment en optimisation multi-objectif de la variante MD-FSMVRPTW.

A la lumière des résultats obtenus, ces derniers ont été comparés aux résultats des solutions les plus connues dans la littérature. Sur les 12 instances de Cordeau et al. [17], nous avons constaté que notre approche a atteint plusieurs fois les meilleurs résultats connus et elle a produit de meilleurs résultats pour 02 instances qui n'ont jamais été atteint auparavant. Cette approche offre également une performance très compétitive en termes de coût moyen de la solution et de temps moyen.

Ces approches donnent de bons résultats, principalement au niveau de la minimisation de la somme des retards où nous avons obtenu, dans plusieurs cas, un retard presque nul.

D'un autre côté, les résultats obtenus apportent une nette amélioration en les comparants à ceux obtenus par la méthode GenClust développée dans [118] « vu qu'elle est la méthode la plus proche à notre approche en terme d'outil d'optimisation et métriques utilisées » avec une somme des retards minimisée où nous avons obtenu, dans plusieurs cas, un retard presque nul.

Enfin, comme perspectives de nos travaux de recherche, nous envisageons de :

- Intégrer d'autres algorithmes, tels que les essais de particules, brain inspired algorithms, la recherche tabou ou les colonies de fourmis pour la résolution du MD-FSMVRPTW,
- Modéliser et optimiser la même variante «MD-FSMVRP-TW» avec des demandes incertaines.
- Développer des méthodes d'insertion des nœuds dynamiques, à savoir l'intégration des demandes urgentes nécessitant un temps de réaction assez rapide,
- Implémenter un Système de dialogue orale homme/machine en langage maternelle « Arabe »,
- Tester nos approches sur des cas réels de transport de marchandises.

BIBLIOGRAPHIE

- [1] P. Toth, and D. Vigo, *Vehicle routing: problems, methods, and applications*: Society for Industrial and Applied Mathematics, 2014.
- [2] G. B. Dantzig, and J. H. Ramser, "The truck dispatching problem," *Management science*, vol. 6, no. 1, pp. 80-91, 1959.
- [3] F. L. Hitchcock, "The distribution of a product from several sources to numerous localities," *Studies in Applied Mathematics*, vol. 20, no. 1-4, pp. 224-230, 1941.
- [4] A. A. Javid, and N. Azad, "Incorporating location, routing and inventory decisions in supply chain network design," *Transportation Research Part E: Logistics and Transportation Review*, vol. 46, no. 5, pp. 582-597, 2010.
- [5] P. Oberlin, S. Rathinam, and S. Darbha, "Today's traveling salesman problem," *IEEE robotics & automation magazine*, vol. 17, no. 4, pp. 70-77, 2010.
- [6] D. Espinouse, Penz, "Problèmes combinatoires classiques," *Recherche opérationnelle et réseaux*, vol. 20, no. 6, pp. 06 - 20, 2002.
- [7] P. Schwerin, and G. Wäscher, "The bin-packing problem: A problem generator and some numerical experiments with FFD packing and MTP," *International Transactions in Operational Research*, vol. 4, no. 5-6, pp. 377-389, 1997.
- [8] J. K. Lenstra, and A. Kan, "Complexity of vehicle routing and scheduling problems," *Networks*, vol. 11, no. 2, pp. 221-227, 1981.
- [9] M. L. Fisher, and R. Jaikumar, "A generalized assignment heuristic for vehicle routing," *Networks*, vol. 11, no. 2, pp. 109-124, 1981.
- [10] R. Spliet, and A. F. Gabor, "The time window assignment vehicle routing problem," *Transportation Science*, vol. 49, no. 4, pp. 721-731, 2014.
- [11] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations research*, vol. 35, no. 2, pp. 254-265, 2007.
- [12] C. Prodhon, "A hybrid evolutionary algorithm for the periodic location-routing problem," *European Journal of Operational Research*, vol. 210, no. 2, pp. 204-212, 2011.
- [13] R. Russell, and W. Igo, "An assignment routing problem," *Networks*, vol. 9, no. 1, pp. 1-17, 1979.
- [14] M. Mourgaya, and F. Vanderbeck, "Problème de tournées de véhicules multipériodiques: Classification et heuristique pour la planification tactique," *RECHERCHE OPERATIONNELLE-PARIS-*, vol. 40, no. 2, pp. 169, 2006.
- [15] P. Francis, K. Smilowitz, and M. Tzur, "The period vehicle routing problem with service choice," *Transportation science*, vol. 40, no. 4, pp. 439-454, 2006.
- [16] J.-F. Cordeau, G. Laporte, and A. Mercier, "A unified tabu search heuristic for vehicle routing problems with time windows," *Journal of the Operational research society*, vol. 52, no. 8, pp. 928-936, 2001.
- [17] J.-F. Cordeau, and G. Laporte, "The dial-a-ride problem (DARP): Variants, modeling issues and algorithms," *4OR: A Quarterly Journal of Operations Research*, vol. 1, no. 2, pp. 89-101, 2003.
- [18] M. Dror, and P. Trudeau, "Split delivery routing," *Naval Research Logistics (NRL)*, vol. 37, no. 3, pp. 383-402, 1990.
- [19] P. H. V. Penna, A. Subramanian, and L. S. Ochi, "An iterated local search heuristic for the heterogeneous fleet vehicle routing problem," *Journal of Heuristics*, pp. 1-32, 2013.
- [20] B. Golden, A. Assad, L. Levy, and F. Gheysens, "The fleet size and mix vehicle routing problem," *Computers & Operations Research*, vol. 11, no. 1, pp. 49-66, 1984.
- [21] H. Yaman, "Formulations and valid inequalities for the heterogeneous vehicle routing problem," *Mathematical Programming*, vol. 106, no. 2, pp. 365-390, 2006.
- [22] E. Choi, and D.-W. Tcha, "A column generation approach to the heterogeneous fleet vehicle routing problem," *Computers & Operations Research*, vol. 34, no. 7, pp. 2080-2095, 2014.
- [23] R. Baldacci, N. Christofides, and A. Mingozzi, "An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts," *Mathematical Programming*, vol. 115, no. 2, pp. 351-385, 2008.
- [24] A. Pessoa, E. Uchoa, and M. Poggi de Aragão, "A robust branch-cut-and-price algorithm for the heterogeneous fleet vehicle routing problem," *Networks*, vol. 54, no. 4, pp. 167-177, 2009.
- [25] R. Baldacci, and A. Mingozzi, "A unified exact method for solving different classes of vehicle routing problems," *Mathematical Programming*, vol. 120, no. 2, pp. 347-380, 2009.
- [26] L. S. Ochi, D. S. Vianna, L. M. Drummond, and A. O. Victor, "An evolutionary hybrid metaheuristic for solving the vehicle routing problem with heterogeneous fleet." in *European Conference on Genetic Programming*, pp. 187-195, 1998.
- [27] M. Gendreau, G. Laporte, C. Musaraganyi, and É. D. Taillard, "A tabu search heuristic for the heterogeneous fleet vehicle routing problem," *Computers & Operations Research*, vol. 26, no. 12, pp. 1153-1173, 2006.
- [28] J. Renaud, and F. F. Boctor, "A sweep-based algorithm for the fleet size and mix vehicle routing problem," *European Journal of Operational Research*, vol. 140, no. 3, pp. 618-628, 2002.

- [29] Y. Lee, J. Kim, K. Kang, and K. Kim, "A heuristic for vehicle fleet mix problem using tabu search and set partitioning," *Journal of the Operational Research Society*, vol. 59, no. 6, pp. 833-841, 2008.
- [30] J. Brandão, "A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem," *European journal of operational research*, vol. 195, no. 3, pp. 716-728, 2009.
- [31] C. Prins, "Two memetic algorithms for heterogeneous fleet vehicle routing problems," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 6, pp. 916-928, 2009.
- [32] A. Imran, S. Salhi, and N. A. Wassan, "A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem," *European Journal of Operational Research*, vol. 197, no. 2, pp. 509-518, 2009.
- [33] J.-M. Belenguer, E. Benavent, P. Lacomme, and C. Prins, "Lower and upper bounds for the mixed capacitated arc routing problem," *Computers & Operations Research*, vol. 33, no. 12, pp. 3363-3383, 2006.
- [34] P. Lacomme, C. Prins, and W. Ramdane-Cherif, "Competitive memetic algorithms for arc routing problems," *Annals of Operations Research*, vol. 131, no. 1, pp. 159-185, 2004.
- [35] P. Lacomme, C. Prins, and W. Ramdane-Chérif, "Evolutionary algorithms for periodic arc routing problems," *European Journal of Operational Research*, vol. 165, no. 2, pp. 535-553, 2005.
- [36] W. Ramdane-Cherif, "Evolutionary algorithms for capacitated arc routing problems with time windows," *IFAC Proceedings Volumes*, vol. 39, no. 3, pp. 321-326, 2006.
- [37] I. H. Dridi, "Optimisation heuristique pour la résolution du m-PDPTW statique et dynamique," Ecole Centrale de Lille; École Nationale d'Ingénieurs de Tunis (Tunisie), 2010.
- [38] R. Baldacci, E. Bartolini, A. Mingozzi, and R. Roberti, "An exact solution framework for a broad class of vehicle routing problems," *Computational Management Science*, vol. 7, no. 3, pp. 229-268, 2010.
- [39] F. Hernandez, "Méthodes de résolution exactes pour le problème de routage de véhicules avec fenêtres de temps et routes multiples," Montpellier 2, 2010.
- [40] G. Laporte, and Y. Nobert, "Exact algorithms for the vehicle routing problem," *North-Holland Mathematics Studies*, vol. 132, pp. 147-184, 1987.
- [41] M. L. Fisher, "Optimal solution of vehicle routing problems using minimum k-trees," *Operations research*, vol. 42, no. 4, pp. 626-642, 1994.
- [42] D. Naddef, and G. Rinaldi, "Branch-and-cut algorithms for the capacitated VRP." in *The vehicle routing problem*, pp. 53-84, 2002.
- [43] M. Padberg, and G. Rinaldi, "A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems," *SIAM review*, vol. 33, no. 1, pp. 60-100, 1991.
- [44] J. Privé, J. Renaud, F. Bector, and G. Laporte, "Solving a vehicle-routing problem arising in soft-drink distribution," *Journal of the Operational Research Society*, vol. 57, no. 9, pp. 1045-1052, 2006.
- [45] S. Ropke, "Formulations and branch-and-cut algorithms for the generalized vehicle routing problem," 2014.
- [46] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "Heuristics for multi-attribute vehicle routing problems: A survey and synthesis," *European Journal of Operational Research*, vol. 231, no. 1, pp. 1-21, 2013.
- [47] I. Ioachim, S. Gelinas, F. Soumis, and J. Desrosiers, "A dynamic programming algorithm for the shortest path problem with time windows and linear node costs," *Networks*, vol. 31, no. 3, pp. 193-204, 1998.
- [48] G. Clarke, and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations research*, vol. 12, no. 4, pp. 568-581, 1964.
- [49] T. Gaskell, "Bases for vehicle fleet scheduling," *Journal of the Operational Research Society*, vol. 18, no. 3, pp. 281-295, 1967.
- [50] P. Wark, and J. Holt, "A repeated matching heuristic for the vehicle routeing problem," *Journal of the Operational Research Society*, pp. 1156-1167, 1994.
- [51] R. Mole, and S. Jameson, "A sequential route-building algorithm employing a generalised savings criterion," *Journal of the Operational Research Society*, vol. 27, no. 2, pp. 503-511, 1976.
- [52] J.-Y. Potvin, and J.-M. Rousseau, "A parallel route building algorithm for the vehicle routing and scheduling problem with time windows," *European Journal of Operational Research*, vol. 66, no. 3, pp. 331-340, 2003.
- [53] K. Fukunaga, *Introduction to statistical pattern recognition*: Academic press, 2013.
- [54] P. Fränti, and J. Kivijärvi, "Randomised local search algorithm for the clustering problem," *Pattern Analysis & Applications*, vol. 3, no. 4, pp. 358-369, 2000.
- [55] C.-F. Tsai, C.-W. Tsai, and C.-C. Tseng, "A new hybrid heuristic approach for solving large traveling salesman problem," *Information Sciences*, vol. 166, no. 1-4, pp. 67-81, 2004.
- [56] X. Zhang, Y. Li, R. Kotagiri, L. Wu, Z. Tari, and M. Cheriet, "KRNN: k Rare-class Nearest Neighbour classification," *Pattern Recognition*, vol. 62, pp. 33-44, 2017.
- [57] Z. Pan, Y. Wang, and W. Ku, "A new k-harmonic nearest neighbor classifier based on the multi-local means," *Expert Systems with Applications*, vol. 67, pp. 115-125, 2017.
- [58] M. A. Mohammed, M. K. A. Ghani, R. I. Hamed, and D. A. Ibrahim, "Review on Nasopharyngeal Carcinoma: concepts, methods of analysis, segmentation, classification, prediction and impact: a review of the research literature," *Journal of Computational Science*, vol. 21, pp. 283-298, 2017.
- [59] M. G. Alfaiakawi, I. Ahmad, and S. Hamdan, "Harmony-search algorithm for 2D nearest neighbor quantum circuits realization," *Expert Systems with Applications*, vol. 61, pp. 16-27, 2016.

- [60] B. E. Gillett, and L. R. Miller, "A heuristic algorithm for the vehicle-dispatch problem," *Operations research*, vol. 22, no. 2, pp. 340-349, 1974.
- [61] J. Renaud, F. F. Boctor, and G. Laporte, "An improved petal heuristic for the vehicle routing problem," *Journal of the operational Research Society*, vol. 47, no. 2, pp. 329-336, 2006.
- [62] J. E. Beasley, "Route first—cluster second methods for vehicle routing," *Omega*, vol. 11, no. 4, pp. 403-408, 1983.
- [63] P. Berkhin, "A survey of clustering data mining techniques," *Grouping multidimensional data*, pp. 25-71: Springer, 2006.
- [64] J. MacQueen, "Some methods for classification and analysis of multivariate observations." in Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, pp. 281-297, 1967.
- [65] R. Xu, and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on neural networks*, vol. 16, no. 3, pp. 645-678, 2005.
- [66] Y. Yunel, "Development of k-means clustering algorithm using genetic algorithm," Istanbul Technical University, 2015.
- [67] I. Or, "TRAVELING SALESMAN TYPE COMBINATORIAL PROBLEMS AND THEIR RELATION TO THE LOGISTICS OF REGIONAL BLOOD BANKING," 1977.
- [68] M. Gendreau, A. Hertz, and G. Laporte, "New insertion and postoptimization procedures for the traveling salesman problem," *Operations Research*, vol. 40, no. 6, pp. 1086-1094, 1992.
- [69] A. Van Breedam, *An Analysis of the Behavior of Heuristics for the Vehicle Routing Problem for a Selection of Problems with Vehicle-related, Customer-related, and Time-related Constraints: RUCA*, 1994.
- [70] G. A. Kindervater, and M. W. Savelsbergh, "Vehicle routing: handling edge exchanges," *Local search in combinatorial optimization*, pp. 337-360, 2007.
- [71] L. Zeng, H. Ong, and K. Ng, "An assignment-based local search method for solving vehicle routing problems," *Asia-Pacific Journal of Operational Research*, vol. 22, no. 01, pp. 85-104, 2005.
- [72] I. H. Osman, and G. Laporte, "Metaheuristics: A bibliography," Springer, 1996.
- [73] S. Voß, S. Martello, I. H. Osman, and C. Roucairol, *Meta-heuristics: Advances and trends in local search paradigms for optimization*: Springer Science & Business Media, 2012.
- [74] T. Stützle, "Local search algorithms for combinatorial problems," *Darmstadt University of Technology PhD Thesis*, vol. 20, 1998.
- [75] Y. Okitonyumbe, and B. E.-L. Ulungu, "Nouvelle caractérisation des solutions efficaces des problèmes d'optimisation combinatoire multi-objectif," 2013.
- [76] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: A survey," *Applied Soft Computing*, vol. 11, no. 6, pp. 4135-4151, 2011.
- [77] V. Černý, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of optimization theory and applications*, vol. 45, no. 1, pp. 41-51, 1985.
- [78] K. A. Dowsland, and J. M. Thompson, "Simulated annealing," *Handbook of natural computing*, pp. 1623-1655: Springer, 2012.
- [79] O. Chebbi, E. Fatnassi, and H. Kaabi, "On Integrating Simulated Annealing Within Parallel Genetic Algorithm: An On-Demand Transportation Case Application." in International Conference on Hybrid Intelligent Systems, pp. 153-163, 2016.
- [80] C.-H. Chen, C.-J. Ting, and P.-C. Chang, "Applying a hybrid ant colony system to the vehicle routing problem," *Computational Science and Its Applications—ICCSA 2005*, pp. 817-817, 2005.
- [81] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & operations research*, vol. 13, no. 5, pp. 533-549, 1986.
- [82] O. Bräysy, and M. Gendreau, "Tabu search heuristics for the vehicle routing problem with time windows," *Top*, vol. 10, no. 2, pp. 211-237, 2002.
- [83] É. Taillard, "Parallel iterative search methods for vehicle routing problems," *Networks*, vol. 23, no. 8, pp. 661-673, 2006.
- [84] M. Gendreau, A. Hertz, and G. Laporte, "A tabu search heuristic for the vehicle routing problem," *Management science*, vol. 40, no. 10, pp. 1276-1290, 1994.
- [85] P. Toth, and D. Vigo, "The granular tabu search and its application to the vehicle-routing problem," *Informatics Journal on computing*, vol. 15, no. 4, pp. 333-346, 2013.
- [86] C. D. Tarantilis, and C. T. Kiranoudis, "BoneRoute: An adaptive memory-based method for effective fleet management," *Annals of operations Research*, vol. 115, no. 1, pp. 227-241, 2007.
- [87] C. Rego, and C. Roucairol, "Using tabu search for solving a dynamic multi-terminal truck dispatching problem," *European Journal of Operational Research*, vol. 83, no. 2, pp. 411-429, 1995.
- [88] S. Ichoua, M. Gendreau, and J.-Y. Potvin, "Exploiting knowledge about future demands for real-time vehicle dispatching," *Transportation Science*, vol. 40, no. 2, pp. 211-225, 2006.
- [89] P. Hansen, and N. Mladenović, "An introduction to variable neighborhood search," *Meta-heuristics*, pp. 433-458: Springer, 2009.
- [90] P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms," *Caltech concurrent computation program, C3P Report*, vol. 826, pp. 1989, 1989.

- [91] P. Shaw, "Using constraint programming and local search methods to solve vehicle routing problems." in International Conference on Principles and Practice of Constraint Programming, pp. 417-431, 2008.
- [92] D. Pisinger, and S. Ropke, "A general heuristic for vehicle routing problems," *Computers & operations research*, vol. 34, no. 8, pp. 2403-2435, 2014.
- [93] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*: MIT press, 1975.
- [94] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*: Addison-Wesley Longman Publishing Co., Inc., 1989.
- [95] O. Bräysy, and M. Gendreau, "Vehicle routing problem with time windows, Part II: Metaheuristics," *Transportation science*, vol. 39, no. 1, pp. 119-139, 2005.
- [96] B. Ombuki, B. J. Ross, and F. Hanshar, "Multi-objective genetic algorithms for vehicle routing problem with time windows," *Applied Intelligence*, vol. 24, no. 1, pp. 17-30, 2006.
- [97] G. Vaira, and O. Kurasova, "Genetic algorithms and VRP: the behaviour of a crossover operator," *Baltic Journal of Modern Computing*, vol. 1, no. 3-4, pp. 161-185, 2013.
- [98] C. Prins, "A simple and effective evolutionary algorithm for the vehicle routing problem," *Computers & Operations Research*, vol. 31, no. 12, pp. 1985-2002, 2014.
- [99] A. Haghani, and S. Jung, "A dynamic vehicle routing problem with time-dependent travel times," *Computers & operations research*, vol. 32, no. 11, pp. 2959-2986, 2005.
- [100] S. Salhi, N. Wassan, and M. Hajarati, "The fleet size and mix vehicle routing problem with backhauls: Formulation and set partitioning-based heuristics," *Transportation Research Part E: Logistics and Transportation Review*, vol. 56, pp. 22-35, 2013.
- [101] M. Skok, D. Skrlec, and S. Krajcar, "The genetic algorithm method for multiple depot capacitated vehicle routing problem solving." in Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on, pp. 520-526, 2008.
- [102] N. E. A. Ghani, S. S. R. Shariff, and S. M. Zahari, "An Alternative Algorithm for Vehicle Routing Problem with Time Windows for Daily Deliveries," *Advances in Pure Mathematics*, vol. 6, no. 05, pp. 342, 2016.
- [103] G. Jeon, H. R. Leep, and J. Y. Shim, "A vehicle routing problem solved by using a hybrid genetic algorithm," *Computers & Industrial Engineering*, vol. 53, no. 4, pp. 680-692, 2007.
- [104] W. Ho, G. T. Ho, P. Ji, and H. C. Lau, "A hybrid genetic algorithm for the multi-depot vehicle routing problem," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 4, pp. 548-557, 2008.
- [105] P. Chen, and X. Xu, "A hybrid algorithm for multi-depot vehicle routing problem." in Service Operations and Logistics, and Informatics, 2008. IEEE/SOLI 2008. IEEE International Conference on, pp. 2031-2034, 2008.
- [106] H. C. Lau, T. Chan, W. Tsui, and W. Pang, "Application of genetic algorithms to solve the multidepot vehicle routing problem," *IEEE transactions on automation science and engineering*, vol. 7, no. 2, pp. 383-392, 2010.
- [107] J. van Hemert, and J. La Poutre, "Dynamic routing problems with fruitful regions: Models and evolutionary computation." in Parallel Problem Solving from Nature-PPSN VIII, pp. 692-701, 2014.
- [108] A. Maniezzo, "Distributed optimization by ant colonies." in Toward a practice of autonomous systems: proceedings of the First European Conference on Artificial Life, p. 134, 1992.
- [109] M. Dorigo, V. Maniezzo, and A. Colomni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29-41, 1996.
- [110] B. Bullnheimer, R. F. Hartl, and C. Strauss, "An improved ant System algorithm for the vehicle Routing Problem," *Annals of operations research*, vol. 89, pp. 319-328, 1999.
- [111] X. Tan, X. Luo, W. Chen, and J. Zhang, "Ant colony system for optimizing vehicle routing problem with time windows." in Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on, pp. 209-214, 2005.
- [112] R. Montemanni, L. M. Gambardella, A. E. Rizzoli, and A. V. Donati, "Ant colony system for a dynamic vehicle routing problem," *Journal of Combinatorial Optimization*, vol. 10, no. 4, pp. 327-343, 2005.
- [113] H. Omar, "Optimisation multicritère de la distribution de produits dans une économie: cas d'If ri," Université Abderrahmane Mira de Béjaia, 2012.
- [114] J. Renaud, F. F. Boctor, and G. Laporte, "An improved petal heuristic for the vehicle routing problem," *Journal of the operational Research Society*, vol. 47, no. 2, pp. 329-336, 1996.
- [115] D. B. Fogel, *Evolutionary computation: the fossil record*: Wiley-IEEE Press, 1998.
- [116] M. Khouja, Z. Michalewicz, and M. Wilmot, "The use of genetic algorithms to solve the economic lot size scheduling problem," *European Journal of Operational Research*, vol. 110, no. 3, pp. 509-524, 1998.
- [117] J. Li, Y. Li, and P. M. Pardalos, "Multi-depot vehicle routing problem with time windows under shared depot resources," *Journal of Combinatorial Optimization*, vol. 31, no. 2, pp. 515-532, 2016.
- [118] S. R. Thangiah, and S. Salhi, "Genetic clustering: an adaptive heuristic for the multidepot vehicle routing problem," *Applied Artificial Intelligence*, vol. 15, no. 4, pp. 361-383, 2011.
- [119] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182-197, 2002.

- [120] C. Silvano, W. Fornaciari, and E. Villar, *Multi-objective design space exploration of multiprocessor SoC architectures*: Springer, 2014.
- [121] M. Ehrgott, "Approximation algorithms for combinatorial multicriteria optimization problems," *International Transactions in Operational Research*, vol. 7, no. 1, pp. 5-31, 2000.
- [122] J. Knowles, D. Corne, and K. Deb, *Multiobjective problem solving from nature: from concepts to applications*: Springer Science & Business Media, 2007.
- [123] A. Liefvooghe, "Métaheuristiques pour l'optimisation multiobjectif: Approches coopératives, prise en compte de l'incertitude et application en logistique," Université des Sciences et Technologie de Lille-Lille I, 2009.
- [124] X. Xue, K. W. E. Cheng, T. W. Ng, and N. C. Cheung, "Multi-objective optimization design of in-wheel switched reluctance motors in electric vehicles," *IEEE Transactions on industrial electronics*, vol. 57, no. 9, pp. 2980-2987, 2010.
- [125] M. G. C. Tapia, and C. A. C. Coello, "Applications of multi-objective evolutionary algorithms in economics and finance: A survey," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pp. 532-539, 2007.
- [126] J. L. Rogers, "A parallel approach to optimum actuator selection with a genetic algorithm," 2000.
- [127] N. Jozefowicz, "Modélisation et résolution approchée de problèmes de tournées multi-objectif," Lille 1, 2004.
- [128] N. Jozefowicz, F. Semet, and E.-G. Talbi, "Multi-objective vehicle routing problems," *European journal of operational research*, vol. 189, no. 2, pp. 293-309, 2008.
- [129] B. M. Ombuki-Berman, A. Runka, and F. Hanshar, "Waste collection vehicle routing problem with time windows using multi-objective genetic algorithms," in *Proceedings of the Third IASTED International Conference on Computational Intelligence*, pp. 91-97, 2007.
- [130] C.-M. Liu, T.-C. Chang, and L.-F. Huang, "Multi-objective heuristics for the vehicle routing problem," *International Journal of Operations Research*, vol. 3, no. 3, pp. 173-181, 2006.
- [131] Q. Jun, J. Wang, and B.-j. Zheng, "A hybrid multi-objective algorithm for dynamic vehicle routing problems," *Computational Science-ICCS 2008*, pp. 674-681, 2008.
- [132] A. Beham, "Parallel tabu search and the multiobjective capacitated vehicle routing problem with soft time windows," *Computer Aided Systems Theory-EUROCAST 2007*, pp. 829-836, 2007.
- [133] C. d. R. Lima, M. C. Goldberg, and E. F. G. Goldberg, "A memetic algorithm for the heterogeneous fleet vehicle routing problem," *Electronic Notes in Discrete Mathematics*, vol. 18, pp. 171-176, 2010.
- [134] É. D. Taillard, "A heuristic column generation method for the heterogeneous fleet VRP," *RAIRO-Operations Research*, vol. 33, no. 1, pp. 1-14, 1999.