Modèle SPG 14

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA
RECHERCHE SCIENTIFIQUE

UNIVERSITE DE MOULOUD MAMMERI, TIZI OUZOU

FACULTE DES : SCIENCES
DEPARTEMENT : MATHEMATIQUES (RO)

THÈSE DE MAGISTER
OPTION : RECHERCHE OPÉRATIONNELLE ET OPTIMISASION

Présenté par :
Meriem AKLI

Sujet :

PROBLÈME DE TOURNÉES DE VÉHICULES AVEC CONTRAINTES ET FENÊTRE DE TEMPS

Devant le jury d'examen composé de :

Mr Mohamed Aidene ;	Professeur ;	UMMTO ;	Président.
Mr Méziane Aïder ;	Professeur ;	USTHB	Rapporteur.
Mr Brahim Oukacha ;	Maître de Conférences classe "A" ;	UMMTO	Examinateur.
Mr Bachir Sadi ;	Maître de Conférences classe "A" ;	UMMTO	Examinateur.
Mme Kahina Louadj ;	Maître de Conférences classe "B" ;	UMMTO	Examinatrice.

Soutenue le : 07/03/2013

Table des matières

Remerciements

Avant tout je dis AL HAMDOU LI ***ALLAH*** *qui ma donné la volonté, la force et le courage pour réaliser ce modeste mémoire.*

Mes remerciements les plus vifs et chaleureux, empreints d'une reconnaissance ineffable, vont à mon directeur de mémoire, Monsieur AIDER MÉZIANE, Professeur à USTHB, pour sa disponibilité, sa confiance et son soutien ainsi que ses orientations judicieuses.

Je remercie particulièrement Monsieur AIDENE MOHAMED pour la compréhension qu'il a montrée à mon égard. Qu'il trouve ici l'expression de ma profonde reconnaissance pour m'avoir fait l'honneur de présider mon jury de mémoire.

Je remercie également les membres de jury, Monsieur OUKACHA BRAHIM, Monsieur SADI BACHIR et Madame LOUADJ KAHINA d'avoir accepté d'examiner mon mémoire.

Pour finir, je souhaite adresser mes sincères remerciements à tous ceux qui ont contribué, de près ou de loin, à la réalisation de ce mémoire et qui m'ont accompagnée durant ces trois années de travail.

Dédicaces

Je dédie ce modeste travail à la mémoire de mon cher frère avec lequel j'aurais tant aimé partager ce moment, ainsi qu'à mes chers parents, qui on tout fait pour mener ce travail à bout, et à qui je souhaite aussi une longue vie.

Introduction générale

Le terme logistique vient du mot grec "LOGISTIKOS", qui signifie "relatif au raisonnement logique". Au départ, il était exclusif au glossaire militaire et actuellement le mot logistique est largement utilisé en économie. En effet, une chaine logistique est un ensemble d'opérateurs garantissant la bonne circulation de marchandises, de l'approvisionnement en matière première à la commercialisation du produit fini prêt à la consommation, passant par le processus de production et les différentes étapes de stockage et de transport. Par conséquent, la logistique comprend l'ensemble des processus liés à la bonne circulation des flux physiques et administratifs, afin de satisfaire les exigences des clients en terme de qualité, coût et particulièrement en terme de temps de livraison, quelle que soit la situation géographique.

L'évolution du marché mondial, caractérisée par la diversification des sources d'approvisionnement, la spécialisation des sites de production, les échanges inter-usines et parfois même internationaux, la création de sites de stockage intermédiaires et la commercialisation des produits, fait apparaitre de nouveaux problèmes auxquels le manager doit faire face. DEn outre, les débats politiques sur la protection de l'environnement et la pollution par les émissions des gaz à effet de serre ($CO, CO_2 \ldots$) et le pétrole qui se fait de plus en plus rare ainsi que l'augmentation des prix de revient des énergies non renouvelables ne fait que renforcer l'inquiétude des managers. Plus de 30% des dépenses d'une chaine logistique sont consacrés au transport, ce taux ne fera qu'augmenter avec le temps.

Donc l'optimisation de la chaine de transport est plus que vitale pour l'épanouissement de toute économie. C'est le paramètre de base, le secret de la réussite, sur lequel le manager doit investir pour se distinguer des autres. Il doit innover sans cesse, créer, des méthodes

lui permettant l'organisation optimale de sa chaine de transport en concordance avec l'ensemble de sa chaine logistique.

Le problème de transport est célèbre en optimisation combinatoire. Le problème de base et probablement le plus étudié, est le Problème du Voyageur de Commerce (PVC) dit aussi Traveling Salesman Problem (TSP), qui a pour objectif, la visite aux moindres coûts d'un ensemble de clients, une et une seule fois, avec un seul véhicule. La difficulté de ce problème est de trouver l'ordre dans lequel chacun des clients sera visité, en minimisant un certain critère (temps ou coût du parcours, ou bien longueur totale parcourue ...).

L'extension du nombre de véhicules de un à une flotte de véhicules définit un autre problème de transport. Il est dit Problème de Tournées de Véhicules ou Vehicul Routing Problem (VRP). En pratique, il modélise autant de problèmes que le PVC. L'enrichissement du PVC par des contraintes relatives au nombre de véhicules, à leurs charges, ou par des contraintes relatives aux clients, à leurs demandes, leurs fenêtres temporelles, définit de nombreuses variantes.

La présente étude est axée sur les problèmes d'optimisation combinatoire et leurs méthodes de résolution, en particulier, les problèmes de tournées de véhicules. Ces problèmes, sont qualifiés de difficiles, ils appartiennent à la classe des problèmes NP-DURS. La classe NP-DURS. La classe NP-DURS est la plus importante des des classes de problèmes d'optimisation combinatoire. Des problèmes du quotidien, tels que les problèmes de distribution et ou de ramassage, les problèmes d'affectation de tâches aux individus, les problèmes d'emploi du temps ..., sont d'autres problèmes d'optimisation combinatoire de la classe des problèmes NP-DURS.

L'objectif de ce travail est la résolution d'un problème de tournées de véhicules classique, auquel nous rajoutons les contraintes de temps. Le problème ainsi défini est noté VRPTW. De nombreux problèmes du processus de transport se modélisent sous la forme de VRPTW, nous citons : la distribution et ou collecte de produits agro-alimentaires périssables tels que le lait et ses dérivés, le pain, les fruits et légumes, les viandes, l'eau potable, le ramassage du personnel ou scolaire.... Le problème de ramassage des déchets(ménagers, hospitaliers ou industriels pour traitement et ou recyclage) représente un marché florissant ces

dernières années, c'est un exemple très concret. D'autres applications du VRPTW, tels que les problèmes de tournées de maintenance (système d'alarme), de visites de médecins ou infirmiers...

La résolution du VRPTW signifie la détermination des tournées de chacun des véhicules, de sorte à minimiser, un seul critère parmi les suivants : la somme des longueurs ou coûts des tournées ou bien la somme des temps de parcours de chacune des tournées ou même le nombre de véhicules utilisés pour la satisfaction des demandes d'un ensemble de clients géographiquemnet dispersés. Dans le cas de la prise en compte de plus d'un critère à minimiser ou à maximiser, on parle d'optimisation multi-objectif, ce cas n'est pas développé dans ce mémoire.

Le problème de tournées de véhicules (VRP) et ses variantes suscitent, ces dernières décennies, la curiosité de nombreux chercheurs (voir bibliographie),qui accordent un intérêt particulier à sa modélisation et sa résolution. Il est impossible de dresser un bilan exhaustif à propos des travaux menés sur les approches de résolution du VRP et ses variantes. Les aboutissements de ces travaux varient selon les approches adoptées, des résultats satisfaisants ont été obtenus pour la majorité des travaux. Quelques uns d'entre eux ont été consultés lors de la réalisation de ce mémoire et les paragraphes suivants montrent trois axes principaux de ces approches de résolutions : Approches exactes, Approches heuristique, Approches hybrides.
Les méthodes de résolution exactes fournissent une (ou des) solution(s) optimale(s) à des instances du VRP de taille relativement petite. Parmi les méthodes exactes les plus utilisées nous citons : la programmation dynamique, la programmation linéaire et linéaire en nombres entiers et les approches de Branch and Bound. Les grandes instances du VRP ne peuvent pas être résolues par des méthodes exactes, car le nombre de solutions réalisables est une fonction exponentielle en la taille du problème. Il est impossible, d'en faire une exploration complète. Le seul moyen de les résoudre est d'utiliser des heuristiques.

Les approches heuristiques permettent de traiter des instances de grande dimmenssion en déterminant , en un temps raisonable une "bonne solution". Afin de trouver cette "bonne solution", l'ensemble des solutions réalisables doit être exploré de façon équilibrée entre intensification et diversification.

Un grand nombre d'heuristiques a été élaboré pour la résolution du VRP et de ses variantes. Ces heuristiques se répartissent en plusieurs catégories distinctes : La catégorie des méthodes de recherche locale telles que la méthode du Recuit Simulé, la Recherche Taboue ..., qui généralement, assurent une recherche intensifiée dans le voisinage de la solution courante. La catégorie des approches évolutionnaires tels que les algorithmes génétiques, la programmation génétique, la stratégie évolutionnaire, la programmation évolutionniste, qui permettent, avec un bon paramétrage, une recherche aussi diversifiée qu'intensifiée. Les approches heuristiques ou méta-heuristiques inspirées de phénomènes naturels forment aussi une catégorie assez vaste des approches de résolution du VRP, parmi lesquelles : Les Colonies de Fourmies, Les Colonies d'Abeilles, l'Optimisation par Essaims Particulaires ....

Parmi les approches citées, nous relevons que certaines assurent une recherche intensifiée et d'autres permettent une recherche diversifiée alors qu'une bonne solution est obtenue par un processus de recherche efficace qui doit être doté d'un mécanisme de diversification et d'intensification. La diversification permet d'éviter une convergence prématurée vers un optimum local et l'intensification garantit que la solution obtenue est la meilleure dans un voisinage bien défini.

De ce fait, une idée d'hybridation des approches est apparue. En effet, l'hybridation est la nouvelle tendance des méthodes de résolution en général, et du VRP en particulier. Elles réunissent les avantages des différentes méta-heuristiques et/ou méthodes exactes dans un seul processus de résolution, de sorte qu'il soit plus performant que les approches qui le constituent.

La plus populaire des stratégies d'hybridation combinent des méthodes de recherche locales es méta-heuristiques à population. Une autre manière d'hybrider consiste en une utilisation parallèle de la même méta-heuristique avec des paramètres variables. Un autre type d'hybridation combine une méta-heuristique et une méthode exacte.

Ce mémoire est constitué de quatre chapitres. Les trois premiers sont consacrés aux résultats et études théoriques, et le quatrième est un résultat du cas pratique considéré dans ce memoire. Une brève description de ces chapitres est faite dans les paragraphes suivants :

Dans le premier chapitre, nous présentons les concepts de base de l'optimisation combinatoire, en particulier le VRP. Ensuite, nous présentons un état d'art du VRP. Le deuxième chapitre fait l'objet d'une description des méthodes de résolution exactes et approchées des problèmes NP-DUR. Le troisième chapitre est consacré aux méthodes de résolution hybrides. Dans le quatrième chapitre, nous décrivons l'environnement de programmation et l'implémentation du logiciel. Le travail est cloturé par une analyse des résultats et une conclusion dans laquelle nous évaluons l'approche de résolution programmée et donnons quelques des perspectives de recherche.

1

Optimisation Combinatoire et Problème de Tournées de Véhicules

1.1 Introduction

Un problème d'optimisation combinatoire, est un problème mathématique qui consiste à déterminer la meilleure solution parmie un ensemble fini de solutions réalisables. De nombreux problèmes font partie de cette branche et les plus réputés sont cités brièvement dans ce chapitre. Parmi eux nous retrouvons le VRP, pour lequel nous accordons une attention particulière.

Nous donnons une description détaillée des composantes et des paramètres ainsi que des formulations mathématiques des Problèmes de Tournées de Véhicules classique (VRP) et énumèrons quelques unes de ses variantes en nous attardons sur la présentation du VRPTW. Nous terminons le chapitre par la présentation de la complexité algorithmique des problèmes d'optimisation combinatoire, en général, et du VRPTW, en particulier.

1.2 Optimisation Combinatroire

L'optimisation ou programmation mathématique représente le pilier principal de la recherche opérationnelle. Un problème d'optimisation consiste à choisir ou à chercher un élément ou une configuration parmi un ensemble de même structure, en optimisant un certain critère défini préalablement. Tout problème d'optimisation peut être formulé comme suit :

$$(P) = \begin{cases} \textit{Optimiser } \mathrm{Z} = & C.x \\ & x \in X \end{cases} \qquad (1.1)$$

Dans la formulation (1.1), X est un sous-ensemble de l'ensemble S des solutions réalisables du problème de base, satisfaisant les contraintes supplémentaires spécifiques au problème (P) de (1.1).

L'optimisation consiste à minimiser ou à maximiser (nous passons la minimisation d'un problème à la maximisation et inversement, en utilisant la relation (1.2).

$$min(Z(x)) = -\max(-Z(x)) \qquad (1.2)$$

Dans tout ce qui suit, et sans perte de généralité, nous considérons uniquement les problèmes de minimisation.

Généralement, nous parlons de problèmes d'optimisation combinatoire lorsque nous avons à examiner un nombre fini de combinaisons. Souvent la résolution de ces problèmes se heurte à la croissance exponentielle du nombre de combinaisons à examiner, en fonction de leurs taille.

Une des conséquences de cette croissance exponentielle, est l'incapacité de tout opérateur, aussi puissant qu'il soit, à énumérer ou simplement à examiner exhaustivement l'ensemble des combinaisons qui en résulte. Il en a découlé l'idée de classification et de comparaison entre les performances de ces processus de résolution qui a donné naissance à la théorie de la complexité des algorithmes de résolution des problèmes d'optimisation combinatoire. La compléxité algorithmique est développée à la fin de ce chapitre.

1.3 Problèmes célèbres en Optimisation Combinatoire

L'optimisation combinatoire englobe une classe importante des problèmes de la recherche opérationnelle. Il nous est impossible de les énumérer de manière exhaustive mais nous en citons les plus fréquents.

Problème du Sac à Dos
L'énoncé de ce problème fameux en optimisation combinatoire est simple : étant donné un ensemble d'objets chacun possédant un poids et une valeur et étant donné une capacité pour le sac, quels objets doit on mettre dans le sac de manière à maximiser la valeur totale sans dépasser la capacité du sac.

Problème de Bin Packing
Il s'agit de trouver le nombre minimum de boites (bins) de capacité fixe pour emballer un ensemble d'éléments de tailles différentes sans que la capacité des boites soit excédée.

Problème d'Affectation
L'objectif, d'un problème d'affectation, est d'affecter un ensemble d'éléments (objets, tâches, endroits, ...) à un ensemble d'individus (machines, endroits, ...) de sorte que le produit du flot circulant entre ces différents endroits et la distance les séparant soit minimisée. Par exemple, la répartition de bâtiments dans un espace donné en fonction du nombre de personnes amenées à circuler entre ces bâtiments, ou la répartition des modules électroniques sur une carte en fonction du nombre de connexions les liant les uns aux autres.

Problème d'Emploi du Temps
Ce problème a plusieurs variantes, selon le domaine d'application, leur point commun est d'élaborer la planification de l'activité de l'individu au cours du temps. La solution est affichée sous forme de tableau à deux dimensions dans lequel on associe les lignes aux personnes et les colonnes aux périodes horaires.

Problème d'Ordonnancement des Taches
Il consiste à définir dans quel ordre, exécuter un ensemble de tâches T sur un ensemble de machines M, de sort qu'une machine exécute une et une seule tâche et que l'ensemble des

tâche soit exécuté en un minimum de temps.

Problèmes Liés à la Théorie des Graphes

Nous citons les problèmes de couplage, de couverture de sommets, de coloration de graphes, de stable mmaximum, le problème de voyageur de commerce (PVC) et le problème de tournées de véhicules (VRP), qui a pour objectif le service d'un ensemble des clients en parcourant une longueur minimale. Le VRP sera étudié de façon détaillée (présentation, méthode de résolution et application) tout au long de ce mémoire.

1.4 Problème de Tournées de Véhicules

1.4.1 Problème de Voyageur de Commerce(PVC)

Il est dit aussi Traveling Salesman Problem (TSP) ou PVC. C'est un problème classique de la recherche opérationnelle. Le voyageur du commerce désire visiter un certain nombre de villes (ou clients). Il doit débuter et finir sa tournée par la même ville de départ, en visitant chacune des autres villes, une et une seule fois, l'objectif étant de trouver la tournée qui minimise la distance totale parcourue. Le PVC permet la formulation de nombreuses situations réelles. Depuis son apparition il n'a pas cessé d'attirer l'attention des chercheurs.

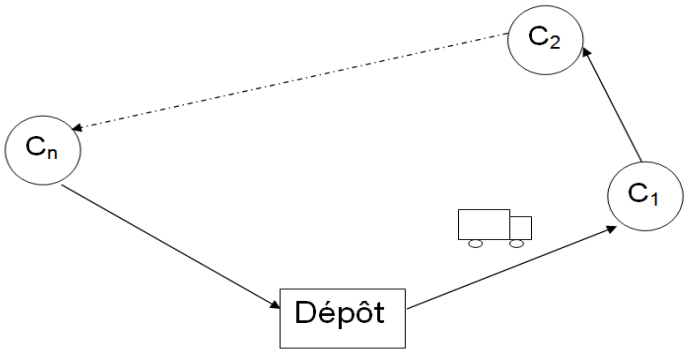

Fig. 1.1 – PVC

Soit $G = (X, U)$ un graphe, où X est l'ensemble des sommets et U l'ensemble des arrêts (ou arcs si le graphe est orienté). Dantzig et al en 1954 [8]ont proposé une formulation mathématique du PVC, où ils définissent :

- Une variable binaire x_{ij} qui prend la valeur de 1 si l'arc (i, j) est utilisé dans la tournée et 0 sinon, et
- C_{ij} le coût du parcours de l'arc (i, j).

Ainsi nous avons la formulation suivante :

$$\text{Minimiser} \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}x_{ij} \tag{1.3}$$

Sous

$$\sum_{i\in X} x_{ij} = 1; \qquad \forall j \in X \tag{1.4}$$

$$\sum_{j\in X} x_{ij} = 1; \qquad \forall i \in X \tag{1.5}$$

$$\sum_{i,j\in S} \leq |S| - 1; \ \forall S \subset X; 2 \leq |S| - 2 \tag{1.6}$$

$$x_{ij} \in 0, 1 \qquad \forall i \in X, \forall j \in X \tag{1.7}$$

La (1.3) formule l'objectif du PVC qui est la minimisation de la longueur totale parcourue par le voyageur. Les contraintes (1.4) et (1.5) assurent que le voyageur passe une et une seule fois par chaque sommet (client). La contrainte (1.6) garantit l'élimination de la formation de sous-tour.

Remarque (1)
Le voyageur de commerce peut être remplacer par un véhicule, postier, médecin, appareil ou machine, ... et les villes peuvent être remplacer par des clients, machines, hôpitaux, malades, foyers....

1.4.2 Problème de Tournées de Véhicules (*VRP*)

Le VRP (Vehicle Routing Problem) a été proposé pour la toute première fois par Lord Hamilton en 1859 [56], puis réintroduit par Dantzig and Ramser en 1959 [8]. Depuis, le VRP a fait l'objet d'études intensives. Le problème de tournées de véhicules n'est qu'une

extension classique M-PVC du PVC.

La version basique du VRP s'énonce comme suit : une flotte de véhicules, basée dans un ou plusieurs dépôt(s), doit assurer des tournées entre plusieurs clients (ou villes) ayant demandés une certaine marchandise ou service. L'ensemble des clients visités par un véhicule désigne la tournée de celui-ci et chaque tournée commence et se termine au dépôt. Chaque client doit être desservi une et une seule fois et par un et un seul véhicule. L'objectif du VRP est de minimiser la somme des distances parcourues ou le temps total de parcours des tournées des véhicules tout en satisfaisant la demande des clients.

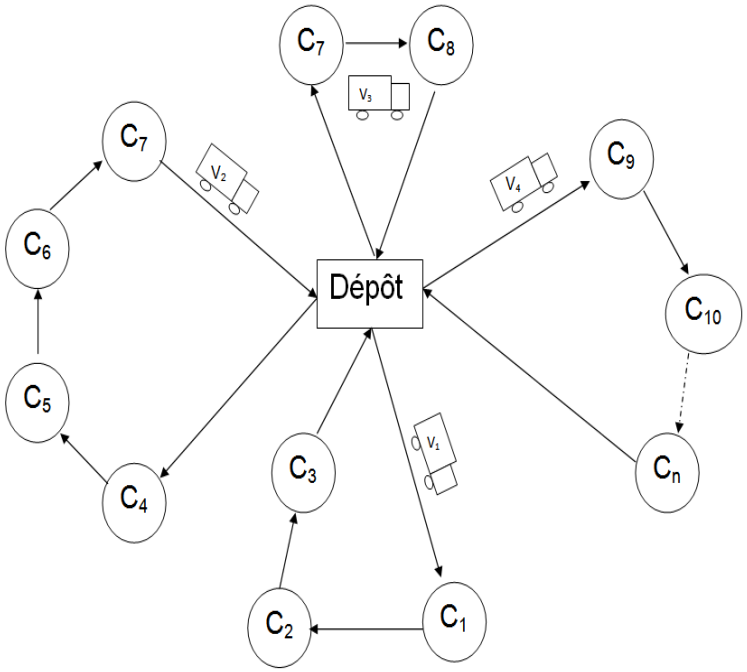

FIG. 1.2 – VRP

1.5 Champs d'Application

Les applications du PVC et VRP sont aussi diverses que variées. Toute entreprise industrielle désire améliorer l'efficacité de sa chaine logistique, pour assurer une production de biens ou de services au moindre coût et une fluidité d'écoulement de sa marchandise. En effet, le problème de tournées de véhicules est un maillon principal dans le domaine de la

logistique. Vu l'augmentation du coût de l'énergie, près de 30% du coût de production d'un bien ou service sont dûs aux frais de transport.

Le problème de tournées de véhicules fait partie de notre quotidien, à commencer par le ramassage scolaire, ramassage du personnel, le ramassage des déchets ménagères, la cueillette de lait cru, la distribution des journaux, de courrier et de denrées alimentaires telles que le lait, le pain, l'eau .... Aussi les services ambulatoires et la distribution urgente de produits pharmaceutiques font partie des problèmes de tournées de véhicules.

Ces deux derniers ont la particularité de servir les clients selon la priorité de l'urgence et l'objectif n'est pas de minimiser la distance totale parcourue mais de servir le maximum de clients avant que ce ne soit trop tard. Nous attirons l'attention sur le fait que tous ces problèmes sont contraints par le temps, et dans les problèmes de ramassage une contrainte de temps est plus rigide que pour les autres les problèmes de distribution.

L'informatique et la robotique sont d'autres domaines ou le VRP est largement utilisé. En effet, le routage de l'information dans les réseaux informatiques à connexion distante ou par câble est fondé essentiellement sur le VRP. La planification du calendrier aérien est une autre application du VRP.

En dépit de l'intérêt que porte l'homme à ce problème, à ce jour, il se trouve incapable de résoudre avec exactitude certaines de ses instances. Les grandes instances du VRP s'avèrent difficiles ou complexes, parfois même "impossibles", à résoudre avec un calculateur quelle que soit la puissanace de celui-ci.

1.6 Paramètres du VRP

Le VRP est caractérisé par le réseau qui le constitue, la clientèle à servir, la flotte de véhicules et la fonction objectif.

1.6.1 Réseau

Le réseau de transport permet la circulation des flux d'individus, de frêt ou des informations. Il est en quelque sorte le squelette d'un système visant à établir une forme de communication. Il peut être schématisé sous la forme d'un graphe complet,symétrique ou asymétrique. Les sommets représentent les clients, caractérisés par sa position géographique (les coordonées (x, y)) et les arêtes représentent les chemins reliant les différents clients. Sous certaines contraintes, ce graphe peut être orienté.

1.6.2 Clientèle

Le client est caractérisé par sa demande qui peut être une demande d'un service ou de produits (marchandises), ces produits peuvent être d'un seul ou de plusieurs types. La demande totale des clients d'une même tournée ne doit pas excéder la capacité Q du véhicule. Aussi, il est caractérisé par sa position dans l'espace. Enfin la demande peut être déterministe (quantité demandée par le client est fixe et connue par le distributeur) ou incertaine (stochastique).

1.6.3 Flotte de véhicules

Le premier critère de la flotte est sa taille (le nombre de véhicules la composant), le second est son homogénéité (les véhicules sont caractérisés par la même capacité d'emport et le même coût de transport) ou hétérogénéité (les véhicules ont des capacités d'emport et/ou coûts de transport différents).

1.6.4 Fonction objectif

Les objectifs les plus courants sont soit la minimisation du nombre de véhicules utilisés soit la minimisation de la distance totale parcourue par les véhicules. D'autres objectifs peuvent être considérés :

- ✓ Minimisation du temps total du parcours de la tournée, du temps d'attente, du temps du retard, du temps de service ....
- ✓ Minimisation du nombre de véhicules.

- ✓ Minimisation du coût total de la tournée, coût fixe à savoir l'amortissement du matériel (véhicule ou autre), salaire des chauffeurs, frais des véhicules .... et en coût variable on cite les pénalités liées aux retards notamment pour le VRPTW ....
- ✓ Maximisation du gain engendré par la tournée dans le cas de collecte de produits chez des clients.
- ✓ Maximisation de la qualité de service.
- ✓ Maximisation du chargement des véhicules utilisés pour les tournées.

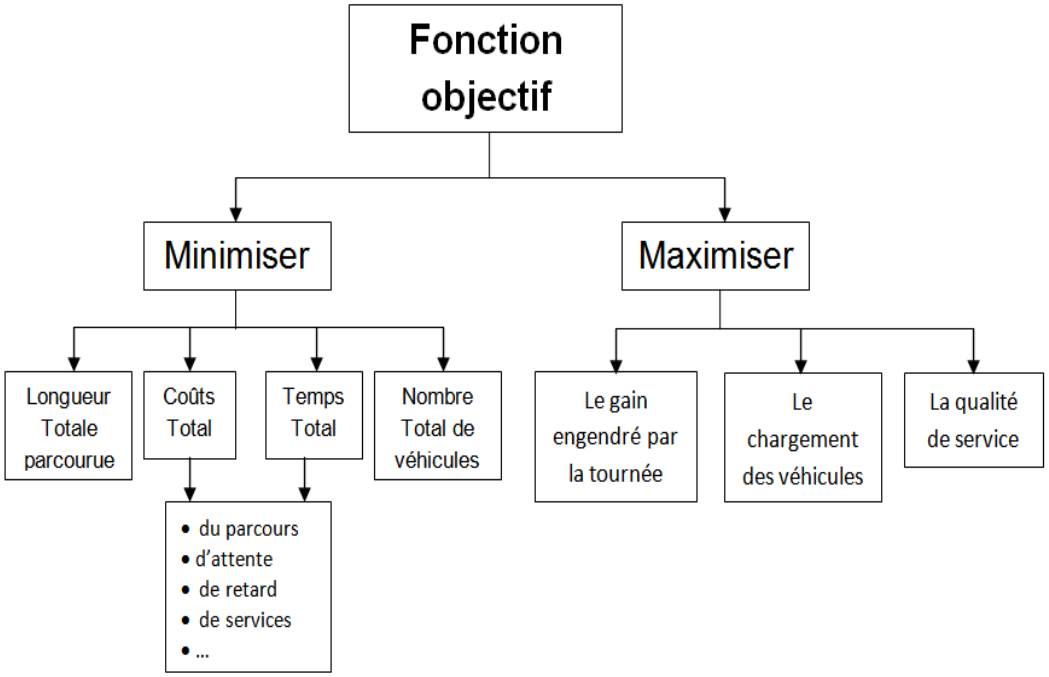

FIG. 1.3 – Différentes Fonctions Objectif

Remarque (2)

Notons que l'objectif peut être porté sur les resources à titre d'exemple la minimisation du nombre de véhicules ou l'utilisation maximale de la capacité du véhicule. Comme il peut être porté sur les tournées en minimisant leurs longueurs ou leur temps ou coût de parcours. Il peut en être porté aussi sur les arcs ou les neuds en minimisant le temps d'attente

par exemple ou maximisant la satifaction des clients ....

En pratique, le VRP est plus complexe, car souvent, il faut considérer simultanément plusieurs objectifs pouvant être contradictoires. Par exemple, la minimisation du nombre de véhicules de la flotte et la mamximisation de la longueur totale parcourue. En effet la diminution du nombre de véhicules engendre le plus souvent une diminution de la distance totale parcourue et inversement. Ce conflit des problèmes multi-objectif peut être résolu par la pondération de Fonctions Objectif et la recherche de l'équilibre de NASH (détermination de l'optimum de Paréto)[68].

1.7 Variantes du VRP

La variation des paramètres du VRP, la suppression et/ou l'ajout ou bien la combinaison de contraintes du VRP classique permettent de définir un ensemble de variantes du VRP. Dans ce qui suit, nous présentons une répartition des variantes du VRP selon le type de contraintes.

1.7.1 VRP à Contraintes Liées à la Flotte de Véhicules

✓ **VRP-C** (Capacitated Vehicle Routing Problem)
C'est un problème de tournées de véhicules avec des contraintes de capacités. Les véhicules ont une capacité d'emport limitée (quantité, volume, poids,...). Ceci se traduit par le fait que la somme des demandes des clients appartenant à une tournée ne doit pas dépasser la capacité du véhicule. En pratique tous les problèmes de tournées de véhicules sont à capacité limitée seulement le degré d'influence de cette contrainte varie. Par exemple, les problèmes de transport de marchandise sont très sensibles à la capacité alors que le problème de distribution de courrier l'est moins.

✓ **VRP-FL** (Vehicle Routing Problem with Full Truckload)
C'est un VRP avec utilisation complète de la capacité du véhicule.

✓ **VRP-HF** (Vehicle Routing Problem with Heterogeneous Fleet)
Pour ce problème la flotte est composée de véhicules de types différents, qui se dis-

tinguent par la capacité, la puissance, le coût de transport, ...

✓ **O-VRP** (Open Vehicle Routing Problem)
Ce problème est identique au VRP, seulement le véhicule est libre de rejoindre ou pas le dépôt après la fin de la tournée. S'il choisit de reprendre le dépôt, il doit reprendre le parcours de la tournée dans le sens inverse.

✓ **VRP-B** (Vehicle Routing Problem Back)
Le VRP-B est un problème où le retour du véhicule au dépôt est exigé. Le véhicule doit rejoindre le dépôt aussi-tôt que le dernier client ait été servi sans reprendre le parcours de la tournée.

1.7.2 VRP à Contraintes Liées à la Demande des Clients

✓ **VRP à Demande Déterministe**
C'est un problème fréquent pour les entreprises qui font des livraisons sur commande. En effet, le livreur connait avant son départ du dépôt la quantité à livrer à chacun de ses clients.

✓ **VRP à Demande Stochastique**
Contrairement au précédent, dans le VRP à demande stochastique, le livreur ne connait pas la quantité à livrer au client il la découvre au moment de le servir. Il estime approximativement la demande de chaque client par une fonction stochastique.

✓ **S-VRP** (Split Delivery Vehicle Routing Problem)
Ce problème consiste à visiter un client plusieurs fois afin de satisfaire entièrement sa demande. Exceptionnellement pour ce problème, la demande du client peut être supérieure à la capacité du véhicule.

✓ **VRP-PD** (Vehicle Routing Problem Pick-up and Deliveries)
C'est un problème de tournées de véhicules avec collecte et livraison. Avec ce genre de problème la durée du service est comptabilisée deux fois, car on doit effectuer une collecte et une livraison ou inversement.

1.7.3 VRP à Contraintes Liées aux Dépôts

✓ **VRP-MD** (Multi-Depôt Vehicle Routing Problem)
Les véhicules peuvent s'approvisionner de plusieurs dépôts.

✓ **VRP-1P** (Vehicle Routing Problem)
Les véhicules doivent s'approvisionner d'un seul dépôt.

On peut passer du VRP-MD au VRP-1D ou invercement par centralisation respectivement division du dépôt(s).

1.7.4 VRP à Contraintes Liées aux Produits

✓ **MP-VRP** (Problème de Tournées de Véhicule à Produits Multiples)
Une gamme de produits doit être livrée aux différents clients par chaque véhicule en une seule tournée.

✓ **1-VRP** (Problème de Tournées de Véhicule à un Seul Produit)
Un seul produit doit être livré aux différents clients par chaque véhicule en une seule tournée.

1.7.5 VRP à Contraintes Liées au Temps

✓ **PVRP** (Periodic Vehicle Routing Problem)
Dans le problème de tournées de véhicules périodique, chaque client est périodiquement visité selon une certaine planification prédéfinie.

✓ **VRP** à Temps de Service Déterministe
La durée du service est connue par le livreur avant d'entamer la tournée.

✓ **VRP** à Temps de Service Stochastique
Le livreur ne connait pas la durée du service des clients, il l'a découvre au moment de les servir. Il peut définir une fonction stochastique pour l'approximer.

✓ **VRP-TW**(Vehicle Routing Problem with Time Windows)

Le VRPTW est un problème de tournées de véhicules avec fenêtre de temps. Chaque client doit être servi dans un intervalle de temps défini, connu d'avance par le livreur et toute violation de cette contrainte peut engendrer une pénalité. Lorsque la contrainte de fenêtre de temps n'est pas satisfaite, soit on rejette la solution si on considère le cas rigide ou bien on construit une fonction de pénalité qui sera rajoutée ou combinée avec la fonction objectif pour le cas relâché. En réalité, c'est un problème très fréquent. La distribution des produits périssables (le lait, la viande ...), de journaux, services ambulatoires, ... sont des examples pratiques du VRPTW. Dans cette classe de problèmes, on distingue deux sous-classes : le VRPTW rigide où le service doit impérativement être effectué dans la fenêtre de temps et le VRPTW relâché où le retard ou l'avence engendre uniquement une pénalité. Lorsqu'on rajoute on définit une variante du VRPTW. Nons citons le CVRPTW, le PDPTW, le 1-PDPTW,le m-PDPTW ....

1.7.6 Problèmes de Tournées de Véhicules Fréquants

✓ **VRP Statique**

Le VRP Statique est un problème dont toutes les composantes sont connues, fixées avant d'entamer la moindre tournée. Ce cas est obtenu soit par une étude rigoureuse de stationnarisation de l'ensemle des paramètres soit parce que le problème modélise à l'origine un phénomène statique.

✓ **DVRP** (Dynamic Vehicle Routing Problem)

Á l'opposé du VRP Statique, le VRP Dynamique a au moins une composante dynamique ou changeante au cours de son exécution. On peut avoir la demande qui ou la fenêtre de temps de servitude qui varient ou bien le nombre de clients à servir qui change. Dans ce cas, le problème devient plus complexe.

✓ **SVRP** (Stochastic Vehicle Routing Problem)

Le SVRP est un cas particulier des problèmes dynamiques. En effet, la variation de la composante considerée suit une loi probabiliste, donc la variation est mise sous forme d'un modèle mathématique, ce qui permet une bonne maitrise de la variation

qui facilite la résolution.

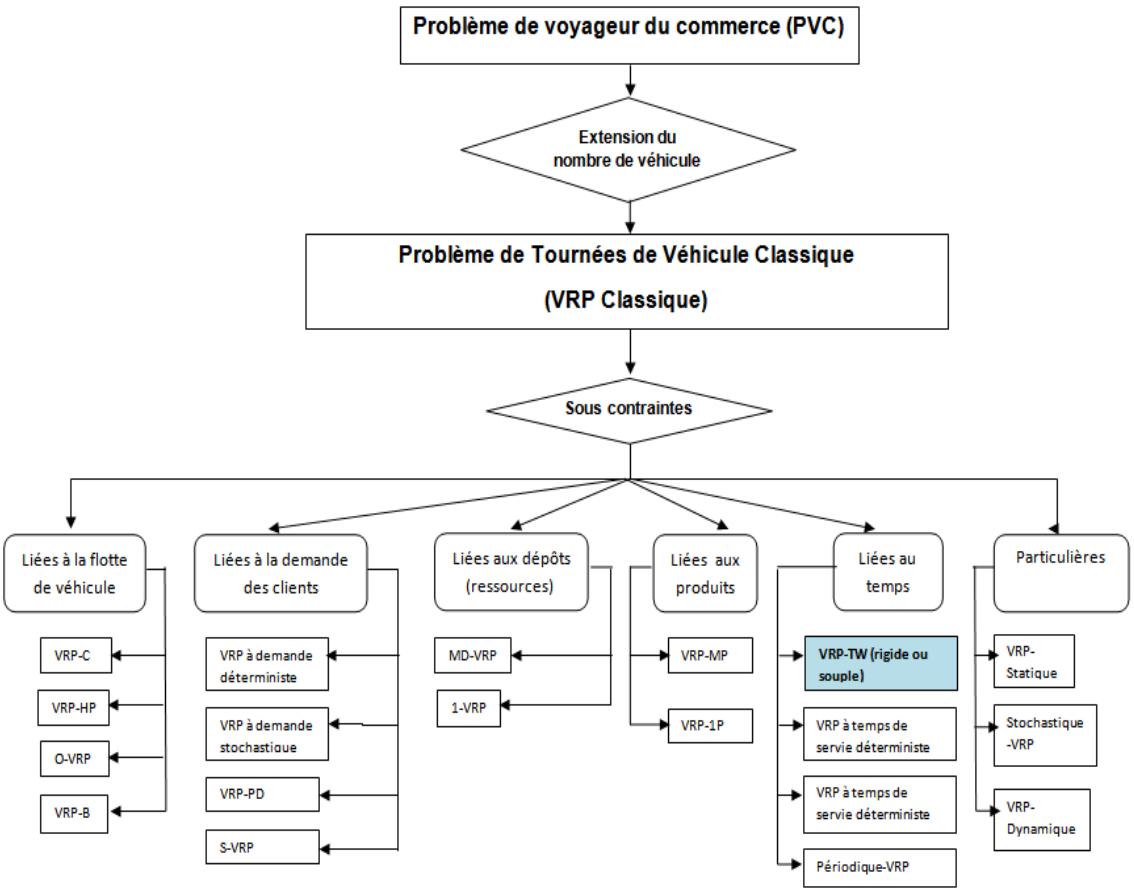

FIG. 1.4 – Variantes du VRP

Remarque (3)

En réalité, on peut avoir un problème avec une combinaison de contraintes. Par exemple, une contrainte de capacité et de fenêtre de temps ou une contrainte de dépôts multiples et collecte et livraison. Ce type de combinaison de contraintes rend le problème plus représentatif du phénomène réel mais plus difficile à résoudre.

1.8 Formulation mathématique du VRP

De nombreuses formulations du VRP existent en littérature. Le point commun de toutes ces formulations est la représentation du problème de tournées de véhicules sous forme d'un graphe orienté.

La représentation graphique du VRP classique est décrite comme suit : soit $G = (X, U)$ un graphe orienté où :

- X, l'ensemble des sommets du graphe G, représente les clients et le(s) dépôt(s) du VRP, $X = N \bigcup \{0\}$ et $(|X| = n + 1)$;
- U, l'ensemble des arcs du graphe G, représente les chemins reliant les clients entre eux et au dépôt du VRP, $(|U| = ((n + 1)n)/2)$;
- Une pondération peut être effectuée sur les sommets (resp. arêtes) pour définir la quantité demandée par le client (resp. distance séparant deux clients ou le temps de déplacement. . .).

1.8.1 Formulation mathématique

De nombreuses formulations mathématiques du VRP classique existent dans la littérature. Les deux formulations présentées ci-dessous sont parmi les plus fréquentes.

Avant d'entamer les formulations, il est nécessaire de définir les paramètres suivants :

n : nombre de clients ;
m : nombre de véhicules $(m \leq n)$;
Q_k : capacité du $k^{ème}$ véhicule ;
T_k : temps maximal de la tournée du $k^{ème}$ véhicule ;
as_j : nombre d'arcs sortants du sommet j ;
ae_j : nombre d'arcs entrants du sommet j ;
q_j : demande du $j^{ème}$ client $d_0 = 0$;
s_{kj} : durée du service du $j^{ème}$ client par le $k^{ème}$ véhicule $(s_0^k = 0)$;
t_{ij} : temps nécessaire pour passer du $i^{ème}$ client au $j^{ème}$ client $(t_{jj}^k = +\infty)$;
d_{ij} : distance séparant $i^{ème}$ client du $j^{ème}$ client $(d_{jj}^k = +\infty)$;
C_{ij} : le coût du déplacement du $i^{ème}$ client au $j^{ème}$ client.

1.8.2 Formulation 1

La formulation mathématique du VRP classique la plus communément utilisée dans la littérature est celle adoptée par Laporte [1992], Rego and Roucairol [1994], Toth and Vigo

[2001 a], Crainic and Semet [2006]. En effet, elle nécessite la définition de $n \times n$ variables de décision de type binaire, à trois indices, suivantes :

$$x_{ij}^k = \begin{cases} 1 & \text{si l'arc } (i,j) \text{ est parcouru par } k^{ème} \text{ véhicule} \\ 0 & \text{sinon} \end{cases} \quad (1.8)$$

Autrement dit : $x_{ij}^k \in \{0,1\};\ i = 0, \cdots, n\,;\ j = 0, \cdots, n\,;\ k = 1, \cdots, m \quad (1.9)$

Le problème se modélise comme suit :

$$\text{Minimiser} \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij} \sum_{k=1}^{m} x_{ij}^k \quad (1.10)$$

Sous

$$\sum_{i=1}^{n}\sum_{k=1}^{m} x_{ij}^k = 1;\ j = 2, \cdots, n \quad (1.11)$$

$$\sum_{j=1}^{n}\sum_{k=1}^{m} x_{ij}^k = 1;\ i = 1, \cdots, n \quad (1.12)$$

$$\sum_{i=1}^{n} x_{ip}^k - \sum_{j=1}^{n} x_{pj}^k = 0;\ k = 1, \cdots, m; p = 1, \cdots, n \quad (1.13)$$

$$\sum_{i=1}^{n} q_i (\sum_{j=1}^{n} x_{ij}^k) \leq Q_k;\ k = 1, \cdots, m \quad (1.14)$$

$$\sum_{i=1}^{n} s_i^k x_{ij}^k + \sum_{i=0}^{n}\sum_{j=0}^{n} t_{ij}^k x_{ij}^k \leq T_k;\ k = 1, \cdots, m \quad (1.15)$$

$$\sum_{j=1}^{n} x_{0j}^k \leq 1;\ k = 1, \cdots, m \quad (1.16)$$

$$\sum_{i=1}^{n} x_{i0}^k \leq 1;\ k = 1, \cdots, m \quad (1.17)$$

$$\sum_{i \in S}\sum_{j \in S} x_{ij}^k \leq |S| - 1\,; \quad \text{pour tout } S \subset N;\ et\ 2 \leq |S| \leq (n-1) \quad (1.18)$$

La formule (1.10) représente la fonction à optimiser ou la fonction objectif du VRP classique. Généralement, l'objectif est de trouver le minimum du coût global des tournées. Les formules de (1.11) à (1.18) représentent les contraints du problème :

Les contraintes (1.11) et (1.12) assurent que chaque client n'est servi qu'une et seule fois et par un et un seul véhicule. La contrainte (1.13) assure la continuité de la tournée : non seulement un véhicule doit passer une et une seule fois chez un client (1.11) et (1.12)

mais il doit impérativement le quiter une fois le service est achevé. La contrainte (1.14) assure que la capacité du véhicule ne sera pas dépassée. La contrainte (1.15) assure que la durée totale d'une tournée ne dépassera jamais sa durée totale maximale. Les contraintes (1.16) et (1.17) assurent le non dépassement de la disponibilité de chacun des véhicules, un véhicule ne sort du dépôt et n'y revient qu'une seule fois. La dernière contrainte (1.18) garantit l'élimination de sous-tours.

1.8.3 Formulation 2

C'est une formulation proposée par Fisher and Jaikumar, en (1978) et (1981), est basée essentiellement sur deux catégories de variables binaires, une à trois indices et une autre à deux indices :

$$x_{ij}^k = \begin{cases} 1 & \text{si l'arc } (ij) \text{ est parcouru par } k^{ème} \text{ véhicule} \quad (1.19) \\ 0 & \text{sinon} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & \text{si le } i^{ème} \text{ client est servi par le } k^{ème} \text{ véhicule} \quad (1.20) \\ 0 & \text{sinon} \end{cases}$$

La formule précédente est équivalente à : $y_{ik} \in \{0,1\};\ i = 1, \cdots, n\,;\ k = 1, \cdots, m \quad (1.21)$

Le problème se modélise comme suit :

$$\text{Minimiser} \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij} \sum_{k=1}^{m} x_{ij}^k \quad (1.22)$$

Sous

$$\sum_{k=1}^{n} y_{ik} = 1;\ i = 1, \cdots, n \quad (1.23)$$

$$\sum_{k=1}^{m} y_{0k} = m; \quad (1.24)$$

$$\sum_{i=1}^{n} q_i y_{ik} \leq Q_k;\ k = 1, \cdots, m \quad (1.25)$$

$$\sum_{j=1}^{n} x_{ij}^k = \sum_{j=1}^{n} x_{ji}^k = y_{ik};\ k = 1, \cdots, m \quad i = 1, \cdots, n \quad (1.26)$$

$$\sum_{i \subset S}\sum_{i \subset S} x_{ij}^k \leq |S| - 1;\ \forall S \subseteq \{2, \cdots, n\}\ et\ k = 1, \cdots m \quad (1.27)$$

La formule en (1.22) est équivalente à la formule (1.10), elle représente la fonction à optimiser. Les formules de (1.23) à (1.27) représentent les contraintes du problème. La contrainte (1.27) est équivalente aux contraintes (1.11) et (1.12), elle assure que chaque client n'est servi qu'une et seule fois et par un et un seul véhicule. La contrainte (1.24) assure que tous les véhicules se trouvent au dépôt. La contrainte (1.25) assure que la capacité de chacun des véhicules ne soit pas dépassée. La contrainte (1.26) permet de lier les deux types de contraintes d'une part, et d'autre part, elle assure que le véhiculen quitte le client aussi-tôt que ce dernier ait été servi. La dernière contrainte (1.19) garantit l'élimination de sous-tour.

Remarque (4)
La formulation précédente représente le VRP classique auquel on a rajouté deux contraintes : une de capacité d'emport des véhicules et une autre de temps total de parcours de chacun des véhicules.

1.9 Complexité algorithmique

La complexité d'un problème quelconque sous-entend l'estimation, dans le pire des cas, du nombre d'instructions à effectuer afin de résoudre une instance du problème. La complexité algorithmique sert à mesurer l'efficacité d'un algorithme de résolution d'un problème. En effet, plus le nombre d'instructions est important plus le problème est complexe et donc la complexité algorithmique est proportionnelle au nombre d'instructions du problème. On en déduit alors que plus l'algorithme est complexe moins il est efficace.

Plusieurs types de complexité peuvent être distingués, on cite la complexité constante $o(1)$ qui est indépendante de la taille du problème, la complexité logarithmique $o(\log(n))$, la complexité linéaire $o(n)$, la complexité quadratique $o(n^2)$, la complexité cubique $o(n^3)$, la complexité polynomiale $o(n^p)$, la complexité exponentielle $o(\exp n)$, et enfin la complexité factorielle $o(n!)$.

Selon le critère de complexité, nous distinguons essentiellement quatre classes de problèmes :

- ✓ **Classe P**
  Dans cette classe on trouve l'ensemble des problèmes pouvant être résolus, en temps

polynomial, par une machine de ***Turing***. La résolution de n'importe quelle instance des problèmes de cette classe se fait en un temps et espace polynomiale d'où l'appelation de classe de problèmes polynomiaux ou la Classe P.

✓ **Classe P-Complet**

Dans cette classe, on trouve l'ensemble des problèmes polynomiaux et l'ensemble des problèmes non polynômiaux où tout problème polynomial peut se réduire en un temps poly-logarithmique avec ou sans l'utilisation d'une machine de calcul déterministe.

✓ **Classe NP**

Cette classe englobe l'ensemble des problèmes qui peuvent être résolus en un temps polynomial sur une machine de Turing.

✓ **Classe NP-DUR**

Cette classe est dite aussi NP-Difficile. Elle regroupe l'ensemble des problèmes dont la complexité est exponentielle ou doublement exponentielle. Cette classe de problème se distingue par la difficulté de résolution optimale ou exacte des grandes instances, en un temps polynomial. Á cet effet, on se contente de trouver des solutions dites de bonne qualité en un temps polynomial. Parmi les problèmes de cette classe, on cite les problèmes de partitionnement d'arbres, des connexités dans un graphe de routage, ... Le problème de tournées de véhicules est un des problèmes de routage. Lenstra et Rinnooy Kan dans [Lenstra and Rinnooy Kan,1981] ont prouvé que le VRP est un problème NP-DUR.

Le problème de tournées de véhicules est un problème NP-DUR. Sa résolution par une méthode exacte s'avère inappropriée pour les instances de grandes taille. Il est donc inévitable de procéder à sa résolution par des approches heuristiques, qui fournissent des solutions réalisables et appréciables en un temps raisonable.

1.10 Conclusion

La majorité des problèmes d'optimisation combinatoire appartiennent à la classe NP-DURS. Leur résolution ne peut se faire que de manière approchée, car à ce jour, il n'existe aucun algorithme permettant leur résolution en un temps polynomial.

Les approches heuristiques qui seront développées dans le chapitre suivant constitent une alternative aux problèmes de résolutions.

2

Approches de Résolution Exactes et Méta-Heuristiques

2.1 Introduction

Ce chapitre est consacré intégralement aux approches de résolution de problèmes d'Optimisation Combinatoire de type NP-DURS. Nous commençons par la présentation des approches exactes, en citant quelques résultats célèbres, ensuite, nous passons aux approches heuristiques et aux méta-heuristiques.

Nous donnons une description générale des approches heuristiques et méta-heuristiques et nous présentons leurs concepts communs. Puis, nous distingons les différents types d'approches selon le nombre de solutions manipulées à chaque itération (S-Méta-heuristique et P-Métaheuristique), la stratégie de recherche (méthodes de voisinages) ou la source d'inspiration (colonie de fourmis, ...). Nous terminons par une conclusion sur les avantages et inconvenients de chacune des approches.

2.2 Approches exactes

Les méthodes exactes, dites aussi complètes, assurent la détermination de solutions optimales pour les problèmes d'OC, suite à l'exploration exhaustive par énumération de l'ensemble des solutions réalisables de ces problèmes.

Durant les soixante dernières années, une large gamme de méthodes exactes a été conçue pour la résolution des problèmes d'OC en général et particulièrement, les problèmes du VRP, qui ne cessent d'attirer l'attention des chercheurs. Parmi les aboutissements les plus récurrents nous citons : Programmation Linéaire, Programmation Dynamique, Approches Branch and Bound ....

2.2.1 Programmation Linéaire

C'est une approche dédiée exclusivement aux programmes linéaires de type :

$$(P) = \begin{cases} Min\ \mathrm{Z} = & C.x \\ S.C. & \\ & A.x = b \\ & x \geq 0 \end{cases} \qquad (2.1)$$

avec
n : nombre de variables $(x = (x_1, \cdots, x_n))$;
m : nombre de contraintes $m \leq n$;
$A = (a_{ij})$; $i = 1, \cdots, m$; $j = 1, \cdots, n$: matrice des contraintes $(m \times n)$ et $rang(A) = m$;
$C = (C_1, \cdots, C_n)$: vecteur ligne des profits (ou gains) ;
$b = (b_1, \cdots, b_m)$: vecteur colonne des seconds membres.

La résolution de ce problème consiste à affecter des valeurs au vecteur $x = (x_1, \cdots, x_n)$ de sorte que la fonction Z soit à son minimum, sans violation de la moindre contrainte.

En 1947 G. Dantzig, a développé la méthode du simplexe. C'est la méthode de résolution de programmes linéaires la plus utilisée à nos jours. Elle s'avère pratique, simple et efficace jusqu'à ce que Klee et Minty, en 1972, mettent en évidence des exemples pour lesquels la

complexité de l'algorithme du Simplexe croit exponentiellement avec la taille du problème à résoudre, raison pour laquelle les chercheurs se sont remis à explorer de nouvelles pistes de la recherche de méthodes de résolution efficaces.

Les explorations les plus fructueuses sont celles qui ont abouti à la méthode des ellipsoïdes élaborée par Khachian en 1979. Cette méthode consiste à utiliser une suite d'ellipsoïdes de volumes décroissants mais contenant à chaque itération la solution optimale du programme linéaire à résoudre. La méthode du point intérieur développée par Karmakar, en 1984, basée sur le principe de recherche de points à l'intérieur du polyèdre des contraintes permettant le passage rapide au sommet optimal. Parmi les dérivées de cette dernière, on cite la méthode de barrière qui se trouve plus efficace que la méthode du simplexe pour certains problèmes de grande taille. Il y a aussi l'algorithme de génération de colonnes qui est une technique de résolution exacte très sophistiquée, appliquée efficacement à des problèmes de grandes instances. On peut également l'adapter et l'insérer dans un algorithme de branchement.

Ces méthodes ne sont pas appropriées aux programmes linéaires en nombres entiers, mais peuvent être ulilisées dans des schémas de résolution basés sur le principe de l'arborescence et celui des coupes.

2.2.2 Programmation Dynamique

C'est une méthode basée sur le principe de Richard Bellman (1957), qui stipule que "une sous-stratégie d'une stratégie optimale est elle-même optimale". Eilon, Wastson-Gandy et Christofides (1971) ont appliqué ce principe pour la résolution du VRPS. Ils l'ont subdivisé en sous-problèmes simples et faciles à résoudre. Les sous problèmes obtenus sont de taille réduite et peuvent être taités. Ils sont liés au problème principal par des contraintes suplémentaires. Leur résolution séquentielle suivie d'une remontée graduelle au problème principal assure la détermination de la solution du problème principal.

Le point fort de cette méthode est le fait d'éviter d'évaluer deux fois la même fonction, généralement en utilisant une table de résultats obtenus, remplie à fur et à mesure que l'on résout les sous-problèmes et c'est ainsi qu'on gagne un temps considérable.

2.2.3 Approche de Branch and Bound

L'approche de Branch and Bound permet une exploration intégrale du domaine des solutions réalisables S d'une façon arborescente, où S est la racine de l'arbre et ses feuilles sont des sous-ensembles de S qui seront à leur tour des racines de nouvelles feuilles.

Dans l'approche de Branch and Bound, on distingue deux étapes principales : séparation et évaluation.

L'étape de séparation permet le passage de la racine à la feuille. Elle partitionne progressivement les ensembles non stériles des solutions en sous-ensembles de taille réduite tout en formant un recouvrement de l'ensemble de solutions S, ensuite on évalue chacun des sous-ensembles obtenus en leurs associant la valeur du coût des solutions leurs appartenant. Le parcours de l'arborescence se fait de trois manières différentes :

- **En largeur**, le sommet à séparer est le premier créé ;
- **En profondeur**, le sommet à séparer est le dernier créé ;
- **En choisissant** la stratégie du **meilleur** d'abord.

La stratégie du parcours joue un rôle déterminant sur l'efficacité de l'algorithme. En pratique la stratégie du parcours en profondeur est la stratégie la plus répandue. En effet, elle permet de trouver très rapidement les solutions réalisables dont l'évaluation affine la borne supérieure et permet de couper plus vite les branches stériles.

L'injection de l'algorithme de génération de colonnes dans l'algorithme de Branch and Bound génère l'algorithme de Branch and Price, qui s'avère efficace pour les problèmes du VRP d'instances dépassant 50 villes.

Christofied, Mingozzi et Toth ont appliqué la méthode de séparation et évaluation au VRPTW-larges à contraintes de capacités de véhicules variables. La meilleure instance résolue est de 53 clients et 08 véhicules. Après résolution du problème, le nombre maximal de véhicules dans une tournée est de 15 [53].

2.2.4 Avantages et Inconvénients

Les méthodes exactes garantissent l'optimalité des solutions qu'elles fournissent. En effet, la détermination de la solution optimale se fait par un parcours exhaustif du domaine des solutions réalisables du problème.

Pour les problèmes d'OC, la taille du domaine des solutions réalisables d'un problème croit exponentiellement avec sa taille, donc le parcours exhaustif ce domaine des solutions réalisables est impossible à partir d'une certaine taille du problème.

2.3 Approches Heuristiques et Méta-Heuristiques

La majorité des problèmes d'optimisation combinatoire font partie de la classe NP-DUR, y compris le VRP abordé dans le chapitre précédent. La complexité des problèmes rend les méthodes de résolution exactes inefficaces, et vu la fréquence et récurrence de ces problèmes en pratique, les chercheurs ont été menés à développer des méthodes approchées pour résoudre ces problèmes complexes. En effet, les méthodes approchées donnent des solutions appréciables à des problèmes de complexité importante en un temps raisonnable.

2.3.1 Heuristiques

Le terme heuristique vient du verbe grec heuriskein ($\varepsilon\upsilon\rho\iota\sigma\kappa\varepsilon\iota\nu$) signifiant **trouver**. Une heuristique permet de trouver une "bonne" solution, en un temps raisonnable, seulement elle n'offre aucune garantie sur l'optimalité de la solution trouvée.
Selon Feignebaum et Feldman (1963), une heuristique est une règle d'estimation, une stratégie, une astuce ou bien une simplification, qui limite la recherche de bonnes solutions dans l'espace des configurations de façon ahurissante.

La performance d'une heuristique apparait essentiellement, en sa complexité spatiale et temporelle, et en sa simplicité et facilité d'implémentation. Sa flexibilité et son débit de génération de solutions sont aussi des critères de performance d'une approche Heuristique. Néanmoins, il est impossible d'évaluer théoriquement avec exactitude les performances d'une heuristique.

Selon la processus de génération de solutions, nous distingons principalement trois types d'heuristiques :

- **Heuristiques constructives :** À fur et à mesure d'itérer le processus, la solution se construit. La solution ne peut être complètement définie qu'à la fin du processus. Par exemple nous citons les algorithmes gloutons, le plus proche voisin ou le plus lointain voisin....
- **Heuristiques d'amélioration :** Elles nécessitent une solution de départ, qui s'améliore aux cours du déroulement de l'algorithme, comme les algorithmes de recherche locale.
- **Heuristiques de deux phases :** Elles consistent en premier à générer une ou plusieurs solutions, auxquelles on applique une procédure d'amélioration.

2.3.2 Méta-Heuristiques

Méta est un préfixe signifiant "au-delà", "plus que ça", " à un niveau supérieur", lorsqu'il est rajouté à l'heuristique il donne Meta-Heuristique qui veut dire trouver au-delà, ou trouver plus que ça. Avant l'apparition du terme Méta-heuristiques, on parlait plutôt des heuristiques modernes.

Les Méta-heuristiques sont des stratégies d'exploration du domaine de solutions réalsables à la recherche de la solution "optimale" ou presque. C'est des approches guidées intelligemment par l'usage de simples procédures de recherche locale et/ou des processus d'apprentissage complexes de sorte à éviter le bouclage ou les optimums locaux .... Elles sont d'ordre général, s'adaptent à tous types de problèmes sans perte d'efficacité.

Comme les heuristiques, les Méta-Heuristiques n'offrent pas de garantie sur l'optimalité, bien qu'on ait pu démontrer la convergence de certaines d'entre elles vers l'optimum global. Non déterministes, elles incorporent souvent un principe stochastique pour surmonter l'explosion combinatoire.

2.3.3 Concepts Communs des Méta-Heuristiques

L'application de toute Méta-Heuristique à un problème requiert la formulation du problème sous la forme d'un modèle mathématique et la définition de(s) objectif(s).

a) Modélisation du Problème

La modélisation d'un problème consiste à déterminer le modèle mathématique le plus représentatif du problème réel. Il doit être simple et précis. Pour celà on doit codifier les données de sorte à faciliter leurs manipulations. Le choix du codage des données se fait selon le type du problème, de données et de la méthode envisagée pour la résolution du problème.

Par exemple, le codage binaire pour les problèmes d'affectation, vecteur d'entiers pour les PLE, vecteur de réels pour les PL ou codage probabiliste ou aléatoire lorsque la variable prend ses valeurs dans l'intervalle $[0, 1]$ .... L'efficacité d'un codage est fortement liée à l'opérateur de recherche appliqué. En effet, lorsqu'on définit un codage on doit tenir compte du nombre de fois que cet opérateur sera réitérer et du nombre de fois que la solution sera évaluée.

Un codage efficace doit être complet, représentatif de toutes les solutions réalisables du problème. Connexe, il ne doit pas empêcher le passage d'une solution à une autre dans le domaine des solutions réalisables. Un codage est efficace lorsqu'il facilite la manipulation de données, permet une réduction de la complexité temporelle et spatiale. En effet, un même problème peut être codifier de deux façons différentes et l'évaluation de la performence d'un codage se fait par la comparaison des critères cités précédemment.

b) Fonctions Objectifs

La fonction objectif, dite aussi fonction coût, associe à chaque solution réalisable une valeur. Par fois, elle est facile à déterminer comme le cas du problème de sac à dos, PVC ou encore le problème d'affectation..., pour ces cas elle prend sa valeur dans $\mathbf{R}$. Pour le cas d'un problème de satisfiabilité elle est soit vraie soit fausse donc deux valeurs seulement sont possibles 0 ou 1. D'autres, peuvent être plus complexes, donc soit on lui applique une transformation de sorte à la simplifier, soit on définie une seconde fonction objectif comparable à la première.

La fonction objectif permet de guider la recherche, particulièrement, la recherche locale. L'usage du gradient de la fonction est très fréquent lors de l'utilisation des méthodes de recherche locale notamment les méthodes du recuit simulé et recherche taboue ou il est

évalué pratiquement à chaque itération.

La fonction objectif permet non seulement d'évaluer la solution mais aussi d'évaluer l'approche de résolution. En effet, l'analyse du comportement de la fonction objectif lors de l'application de deux méthodes de résolutions à un problème permet d'évaluer l'efficacité relative de ces approches.

c) Contraintes

Les contraintes peuvent être définies comme des restrictions ou limitations des valeurs, des variables de décision, combinées. Elles peuvent être sous forme d'une équation d'égalité ou inégalité, linéaire ou non linéaire.

Les contraintes sont des stratégies qui agissent principalement, sur les variables de décisions et donc agissent indirectement sur la fonction objectif. Un classement de ces contraintes, selon les stratégies quelles représentent, a été ainsi proposé :

- **Stratégie de rejet** : cette stratégie est employée lorsqu'on veut extraire un ensemble très réduit de solutions non réalisables d'un grand ensemble de solutions réalisables, particulièrement lorsqu'elles sont sur la frontière.

- **Stratégie de pénalisation** : lors de violation d'une contrainte définissant le domaine de réalisabilité, une modification est apportée sur la fonction objectif : cette modification peut être compensation par une para métrisation statique, dynamique ou adaptative.

- **Stratégie de codage** : permet de spécifier les valeurs des variables, par exemple lorsqu'on choisi un codage binaire on limite les valeurs des variables à 0 et 1.

- **Stratégie réparation** : elle est utilisée lorsque l'opérateur de recherche génère des solutions non réalisables, avec une stratégie de réparation on rond ces solutions non réalisables.

- **Stratégie préservation** : cette classe de stratégie est adaptée pour des problèmes

spécifiques dont les opérateurs de recherche produisent des solutions réalisables qui seront préservées.

d) Critère d'Arrêt

Comme tout processus itératif, les méta-heuristiques sont dotées d'un critère d'arrêt. Il peut être une limitation du nombre d'itération, l'atteinte d'un certain taux d'amélioration d'un paramètre ou de fonction .... Il assure la finitude du processus de recherche et lui évite de tourner indéfiniment (cyclage).

2.4 S-Méta-Heuristiques

Les S-Méta-Heuristiques dites aussi Méta-Heuristiques de voisinages ou bien tout simplement, les Méthodes de Recherche Locale, sont des méthodes d'amélioration itératives. En effet, elles fournissent à chaque itération une solution du voisinage de la solution de départ de sorte que la valeur de la fonction objectif de cette nouvelle solution soit au moins aussi bonne que la valeur de la fonction objectif de la solution de départ. Le pseudo algorithme suivant indique la structure générale de la Méthode de Recherche Locale de base :

Algorithme 1 Recherche Locale de base

1: **Initialisation :** Insérer s_0
2: $N(s_0)$
3: $s^* \leftarrow s_0$
4: **Tant que** critère d'arrêt non satisfait
5: **faire** Trouver s
6: **si** $f(s) < f(s_0)$
7: **alors** $s_0 \leftarrow s$
8: $s^* \leftarrow s$
9: **fsi**
10: **fait**

D'après ce pseudo algorithme, nous observons que la Méthode de Recherche Locale est basée essentiellement sur une solution de départ, une structure de voisinage et un critère d'arrêt.

2.4.1 Solution de Départ (Initiale)

Pour un problème d'OC donné, la génération de la solution initiale, se fait par l'unes des trois approches suivantes :

- ✓ Approche aléatoire Elle permet de générer une solution initiale rapidement, mais cette dernière peut être très loin de la solution optimale, ce qui engendre un nombre plus important d'itérations de la Méta-heuristique, donc une convergence lente de l'algorithme de résolution.
- ✓ Approche gloutonne Elle permet la convergence rapide de l'algorithme de résolution, vers la solution optimale, pour de nombreux problèmes d'OC, mais s'avers moins efficace pour les problèmes TPS.
- ✓ Approche hybride C'est une approche qui combine, au moins, deux approches. Á titre d'exemple une approche hybrideissue de la combinaison entre une approche aléatoire et une glotonne....

Un compromis entre la qualité de la solution initiale (l'éloignement ou rapprochement de la solution optimale) et temps d'exécution (long ou court), doit être établi. Ce compromis dépend essentiellement de l'efficacité de l'approche utilisée pour l'initialisation du problème et des propriétés de l'algorithme d'amélioration.

2.4.2 Structure du Voisinage

Soient s est un élément de S ($s \in S$) et S un l'espace normé. On appelle voisinage de s dans S toute partie de S contenant la boule ouverte de centre s de rayon ε.
On définit la fonction N :

$$\begin{aligned} N : S &\rightarrow S \\ s &\rightarrow N(s) \subset S \end{aligned} \qquad (2.2)$$

Où $N(s) = \{s' \in S \quad tq : \quad \| s - s' \| \leq \varepsilon; \varepsilon > 0\}$ (2.3)
et $\| \ . \ \|$ est une norme de l'espace S déterminant la distance entre s et s' et $N(s)$ est l'ensemble des solutions admissibles, voisines de s obtenues en appliquant la norme $\| \ . \ \|$ dans l'espace normé S.

Ce mouvement peut prendre plusieurs formes selon la nature de l'espace S (continu, discret, binaire ...) et le type de problème ainsi que la méthode de résolution : il peut être une norme si S est un espace normé ou une simple permutation, inversion, déplacement, complémentation ou bien insertion, pour un codage de type chaine de caractères ou de variables binaires.

Le choix du mouvement déterminant le voisinage est une étape cruciale dans la résolution d'un problème d'OC avec une méthode de recherche locale. En effet, l'efficience de cette dernière est liée à la taille du voisinage.

Le nombre de voisins, dit aussi taille ou diamètre du voisinage, influe sur la complexité de l'approche de résolution. En effet, lorsque le voisinage est restreint le temps de son exploration est réduit ce qui diminue la complexité temporelle.

a)Types de voisinages

Un voisinage est défini par un mouvement. Le choix du mouvement varie selon le type de codage considéré. Les mouvements les plus fréquents sur un codage binaire ou chaines de caractères sont :

- **Complémentation** (remplacement), pour les solutions codées en binaire (une chaîne en 0, 1), on remplace un 0 quelconque de la chaîne par son complémentaire 1. Par exemple 100111001 devient 100110001, qui créé un voisinage de taille n, elle est d'ordre polynomial.
  Dans ce qui suit, on considère un codage de la forme d'une chaine de caractères.
- **Échange** : il consiste à intervertir les caractères situés en deux positions données de la chaîne. Ainsi $A\underline{B}CD\underline{E}FG$ devient $A\underline{E}CD\underline{B}FG$ par échange des positions 2 et 5. Ce mouvement donne un voisinage de taille $\frac{n(n-1)}{2}$, il est en $o(n^2)$.
- **Insertion-décalage** : elle consiste à choisir l'élément x_j et la position i dans laquelle il sera inséré, ensuite décaler les éléments des positions $i, i+1, \cdots, j$. Tel que $ABCD\underline{E}FG$ insertion décalage $A\underline{E}BCDFG$. La taille du voisinage obtenu par l'application de ce mouvement est $(n-1)$ elle est d'ordre polynomial.
- **Inversion** : elle consiste à choisir deux positions i et $i+1$, puis à inverser l'ordre d'écriture des caractères situés aux positions. Tel que $ABC\underline{ED}FG$ inversion $ABCE\underline{DF}G$.

Ce mouvement donne un voisinage de taille $(n-1)$, il est en $o(n^2)$.

Les mouvements de complémentarité et d'échange sont généralement les plus utilisés. Ils permettent de définir un voisinage de taille réduite, et de complexité d'ordre polynomial.

b)Évaluation du voisinage

Le choix du mouvement à effectuer dépend de la nature du problème, du type de codage des solutions et de l'approche de résolution. Le choix du mouvement influe sur sur la taille du voisinage et donc sur l'efficacité de l'algorithme. En conclusion, la taille du voisinage est un des critères d'évaluation du voisinage. En effet, ce dernier peut avoir une influence positive ou négative sur l'efficacité de l'approche.

Néanmoins, il est parfois nécessaire d'utiliser un voisinage large afin de garantir l'existence d'un optimum local ou d'une solution meilleure que s. Le choix d'un voisinage large assure la diversification des solutions.

2.4.3 Critère d'Arrêt

Le critère d'arrêt est commun à l'ensemble des méthodes de résolution itératives (Heuristiques, Méta-Heuristiques, hybrides . . .). En effet, il permet d'éviter le cyclage du processus. Le critère d'arrêt se défini de deux manières :

a)Statique

Fixer préalablement le nombre d'itérations à effectuer. Généralement la détermination de ce nombre se fait sur la base des résultats empiriques. En algorithmique, cela se traduit par l'insertion d'un compteur.

b)Dynamique

Varie selon une certaine formule qui dépend des résultats des itérations précédentes d'une fonction aléatoire.

2.5 Variantes de S-Méta-Heuristiques

Afin d'échapper à l'optimum local, des manoeuvres ont été créées. En effet, ces manoeuvres permettent d'échapper aux optimums locaux d'une part et d'autre part elle accélérent le processus de recherche de solution.

Dans ce qui suit, nous présentons les maneuvres ou stratégies qui permettent d'échapper aux optimums locaux. Lorsqu'elles sont appliquées à des méthode de recherche locale, elles définissent les variantes de la Méta-Heuristique de voisinage.

2.5.1 Approches du Recuit Simulé (Simulated Annealing)

En 1983, Kirkpatrick and al. ont proposé une méthode de résolution itérative tout à fait nouvelle. Elle est dite Méthode du Recuit Simulé (RS). Elle permet la résolution efficace des problèmes complexes tels que le traitement d'images, conception des réseaux électroniques, l'organisation des réseaux informatiques, reconnaissance de formes (codes postaux), ou même les problèmes du voyageur de commerce ramassage et/ou de collecte de biens .... Ils se sont inspirés du phénomène de la thermodynamique appliqué en physique statique.

En effet, la température représente un paramètre de commande permettant au physicien de modifier l'état d'un matériau. Un matériau porté à une température assai élevée, devient liquide donc le degré de liberté (désordre) des molécules, le constituant, est augmenté. Partant de cet état, plusieurs structures peuvent apparaitre selon la façon dont la température est abaissée. Un état métastable, molécules peu ordonnées ou structurées, est atteint. L'abaissement brutale de la température fait apparaitre des défauts dans le matériau. Pour l'obtention d'une structure solide, organisée et cristalline, un réchauffement local doit être effectué suivi d'un refroidissement progressif afin d'atteindre un état parfaitement stable, un minimum absolu d'énergie interne.

Par analogie à ce processus, on applique la méthode du recuit simulé à des problèmes d'OC NP-difficiles pour échapper à un optimum local, état métastable, et ce en acceptant, avec une certaine probabilité, la dégradation de la fonction objectif, qui est en physique statique le degré d'énergie interne ou le degré de liberté de molécules décri appliqué, afin de trouver une solution plus efficace que celle en cours.

La probabilité P d'acceptation de passage d'une solution s à une autre s' est dite probabilité de Boltzmann, au faite, elle est en fonction de la température T et de la constante de Boltzmann k_b :

$$P = \exp\left(-\frac{f(s')-f(s)}{k_b.t}\right) \qquad (2.4)$$

D'après la formule (2.4) la baisse de la température augmente la probabilité P d'acceptation de passage d'une solution s à une autre s'. En effet, la température décroît au fil du temps de sorte que la recherche tend à s'intensifier vers la fin de l'algorithme.

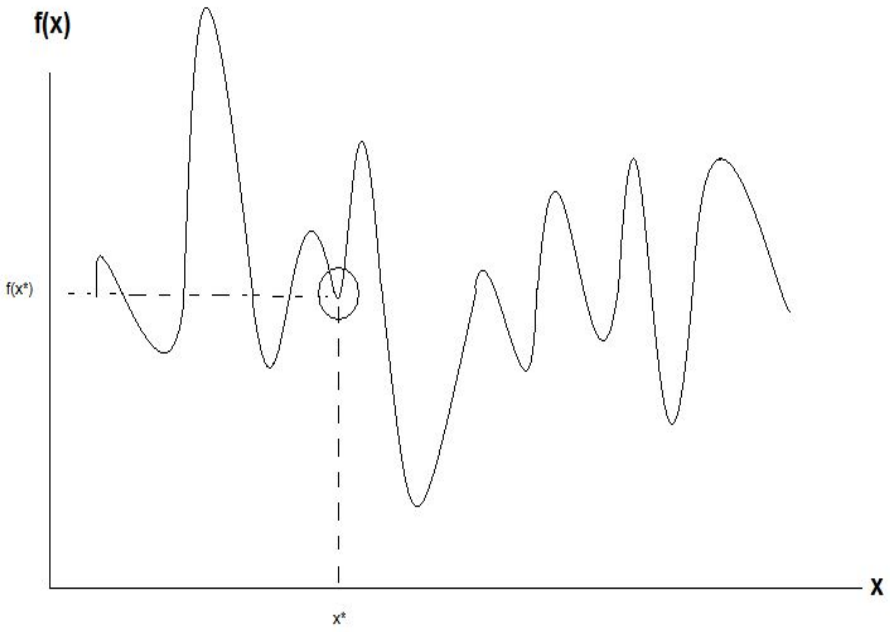

FIG. 2.1 – Recuit simulé

a) Convergence de l'Approche du Recuit Simulé

De nombreux travaux ont été menés sur la convergence de la méthode du RS, principalement Aarst al.(1985), Hajek (1988) et Hajek and al.(1989), qui, sous certaines conditions, ont abouti théoriquement à la convergence de la méthode du RS en probabilité vers l'optimum global.

b) Avantages et Inconvénients de l'Approche du Recuit Simulé

La méthode du RS est une méthode générale, en effet elle s'adapte quasiment à touts problèmes de type NP-difficiles. Elle est pratique et facile à programmer, aussi permet de fournir des solutions de qualité appréciable en un temps raisonnable.

Les nombreux paramètres composants la méthode peuvent constituer un inconvénient, bien que les valeurs publiées permettent, généralement, un fonctionnement efficace de la méthode. Il y a un aspect empirique et aléatoire que les études théoriques s'efforcent de négliger. En pratique, par fois le temps d'exécution est excessif, afin d'y remédier le choix des paramètres de la méthode doit être judicieux, particulièrement, la fonction de décroissance de la température.

La mise en œuvre interactive, la parallélisation de l'algorithme et la prise en considération des progrès effectués par la physique statique dans l'étude des milieux désordonnés peuvent être des directions de recherche prometteuses pour améliorer l'efficacité de cette méthode en terme de temps d'exécution.

Il se trouve que, en pratique, cette méthode est efficace pour les problèmes d'ordonnancement, néanmoins elle reste aussi efficace pour les problèmes du VRP.

c) Approches dérivées du recuit simulé

De nombreuses méthodes similaires à la méthode RS ont été proposées en littérature, nous citons :

- ✓ Threshold accepting, qui est plus rapide que le RS car la génération aléatoire des nombres et fonctions exponentielles réduisent considérablement le temps de calcul.
- ✓ Demon algorithms, elle est basée essentiellement sur la variation de la fonction objectif, une valeur initiale à laquelle on doit rajouter ou soustraire la variation de la fonction objectif selon que cette dernière soit supérieure ou inférieure à la valeur initiale.

d) Algorithme du Recuit simulé

Algorithme 2 Recuit simulé

Entrées Inserer
Programme de refroidisement
$s \leftarrow s_0$/*générer la solution initiale*/
$T \leftarrow T_{max}$/*générer la température initiale*/
Répéter
Répéter/*à une température fixée*/
Choisir aléatoirement s' voisin de s
$\nabla E \leftarrow f(s') - f(s)$
si $\nabla E \leq 0$
alors $s \leftarrow s'$/*solution améliorée*/
sinon $s \leftarrow s'$;
avec une probabilité de exp $\frac{-\nabla E}{T}$
fsi
jusqu'à satisfaction de l'équilibre
/*fixer le nombre d'itération à exécuter pour chaque température*/
$T \leftarrow g(T)$ /*réduire la température*/
jusqu'à satisfaction du critère d'arrêt/*$T < T_{min}$*/
Sortie afficher la meilleure solution trouvée

2.5.2 Méthode de Recherche Taboue "Tabu Search"

L'idée fondamentale de la recherche taboue est de conserver une trace des solutions récemment visitées dans une liste taboue. Cette liste est utilisée lors des déplacements dans le voisinage de manière à ne pas revenir sur des solutions déjà visitées. Pour cela, lors de la recherche locale, le voisinage de la solution courante est réduit aux solutions ne faisant pas partie de la liste taboue. Une étude plus approfondie est présentée dans le Chapitre 3.

2.5.3 Autres variantes des Méthodes de Recerche Locale

a) Méthode de Recherche Locale à Voisinage Variable *(RLVV)*

Avec cette méthode on démarre d'une solution initiale s_0 et d'un ensemble de voisinage ordonné selon un certain critère $V = \{V_1, \cdots, V_{k_{max}}\}$, ensuite à chaque itération on effectue une recherche locale dans un voisinage V_k : si on trouve une solution s^* meilleure que s_0 dans un voisinage k alors on reprend la recherche à partir de s^* dans le même voisinage k sinon on passe à un voisinage d'ordre supérieur $k+1$ et ainsi de suite jusqu'à satisfaction du critère d'arrêt.

Algorithme 3 Recherche Locale à Voisinage Variable

Entrées Inserer
s_0 /*générer la solution initiale*/
$V = V_1, \cdots, V_{k_{max}}$/*générer la structure du voisinage*/
$k \leftarrow 1$
$s^* \leftarrow s_0$
Tant que $k \leq k_{max}$
faire $s \leftarrow RechercheLocale(s_0, V_k)$
si$f(s) < f(s_0)$
alors$s_0 \leftarrow s$
$s^* \leftarrow s$
sinon $k \leftarrow k+1$
fsi
fait

b) Méthode de Recerche Locale Guidée*(RLG)*

Avec la méthode de recherche locale guidée on échappe à l'optimum locacal par le changement de la fonction objectif et le rajout d'une fonction dite de pénalité qui prend une valeur positive lors du changement de la fonction objectif. Ensuite, une nouvelle recherche locale est lancée en utilisant la fonction de coût modifiée. Un état de l'art de la recherche locale guidée et de ces applications est réalisé dans [19].

c) Méthode Recerche Locale de Briutage *(RLB)*

La méthode de bruitage a été introduite sur des problèmes dont la donnée comporte un ensemble de nombres réels, par exemple, le voyageur de commerce (arêtes values par des réels) et le problème d'affectation quadratique (matrices de réels). Cette méthode fait appel à une notion de bruitage de la donnée 13 qui est définie comme suit : la donnée bruitée est produite à partir de la donnée initiale en ajoutant à chacun des réels une composante calculée comme le produit de trois éléments : **1)** une fonction aléatoire à valeurs dans $[0, 1]$, **2)** un paramètre permettant de contrôler le niveau du bruit, **3)** le plus grand des réels concernés, afin de normaliser le niveau du bruit par rapport à la donnée.

d) Méthode de Recerche Locale Itérée *(RLI)*

La méthode RLI est une méthode dotée d'un mécanisme permettant au processus de résolution d'échapper aux optimums locaux. Ce mécanisme consiste à alterner une méthode de recherche locale et une perturbation. On commence par l'application d'une méthode de recherche locale, en suite on perturbe la solution obtenue par l'application de mouvements aléatoires pouvant même dégrader la solution courante et puis on réapplique une recherche locale à la solution obtenue par la perturbation. Le processus est ainsi réitéré jusqu'à satisfaction d'un certain critère d'arrêt.

Algorithme 4 Recherche Locale Itérée

Entrées Inserer

s_0 /*générer la solution initiale*/

$s^* \leftarrow recherchelocale(s_0)$/*appliquer une recherche locale às_0*/

4: **Répéter**

$s' \leftarrow perturbe(s_*, hystoriquedelarecherche)$

/*pérturber s^* solution issue de la recherche locale appliqué à s_0*/

$s'' \leftarrow recherchelocale(s')$ /*appliquer une recherche locale

8: pour la solution issue de la pérturbation*/

$s^* \leftarrow accepter(s^*, s'', hystoriquedelarecherche)$ /*critère d'acceptation*/

jusqu'à satisfaction du critère d'arrêt /*$T < T_{min}$*/

Sortie afficher la meilleure solution trouveée

L'exécution de l'heuristique comporte plusieurs étapes. Chaque étape consiste à calculer

un bruitage de la donnée, puis à effectuer une descente tenant compte de la valeur de la fonction de coût calculée à partir de la donnée bruitée. Le niveau du bruit est décrémenté au début de chaque nouvelle étape, et la descente s'effectue à partir de la configuration résultante de l'étape précédente. Il existe deux variantes pour le bruitage : **1)** chaque descente sur la donnée bruitée est suivie par une descente effectuée sur la donnée non bruitée. L'objectif consiste à mieux tenir compte de la donnée réelle puisqu'un véritable optimum local est alors atteint ; **2)** la configuration courante est régulièrement remplacée par la meilleure configuration obtenue depuis le début. Cette méthode a été appliquée avec succès au problème du voyageur de commerce et a donnée de très bons résultats selon les auteurs.

2.5.4 Évaluation des Méta-Heuristiques de Voisinage

- **L'avantage** des Méta-Heuristiques de voisinage est de fournir une solution "satisfaisante" en un temps raisonnable, sans l'exploration exhaustive de l'ensemble de solutions réalisables du problème. En plus, ce sont des approches générales, qui peuvent être adaptées à tout type de problèmes d'OC sans difficulté de programmation.
- **L'inconvénient** majeur des Méta-Heuristiques de voisinage est le fait qu'elles s'arrêtent au premier optimum local rencontré.

2.6 Approches Évolutionnistes

Les premiers résultats des travaux sur la génétique, étude de l'hérédité, ont été introduit par Johann Gregor Mendel, au 19^{eme} siècle. En 1859, Charles Robert Darwin présente son livre " On the Origin of Species " sur la théorie de l'évolution. Selon C.R.Darwin, les mécanismes à l'origine de l'évolution des êtres vivants reposent sur la compétitivité qui sélectionne les individus les plus adaptés à leur milieu pour assurer la transmission et la descendance des caractéristiques utiles qui ont permis la survie des parents (préservation de la race). De nos jours, aucune théorie ne remet en cause ou ne contredit la théorie d'évolution de C.R.Darwin.

Durant les dernières années, de nombreux chercheurs mathématiciens se sont inspirés de la théorie d'évolution de C.R.Darwin afin de développer des méthodes évolutionnistes, dites

aussi darwinistes, qui permettraient la résolution des problèmes complexes. L'apparition des calculateurs électroniques durant les années cinquante a permis à ces chercheurs de rendre ces méthodes accessibles à la simulation. Tester sur des problèmes d'ingénierie s'est avéré peu satisfaisant à cause des connaissances insuffisantes de la génétique et des calculateurs électroniques qui étaient peu performants.

Dès les années soixante dix, des calculateurs électroniques plus performants commencent à être plus accessibles, ce qui a incité de nombreux chercheurs à entreprendre parallèlement de nouvelles tentatives d'élaboration de méthodes évolutionnistes pour la résolution des problèmes complexes.

Les approches évolutionnistes sont des processus itératifs, de recherche stochastique, appliqués avec succès pour les problèmes d'optimisation complexes. Le principe général de ces algorithmes est de choisir, selon une valeur de fitness donnée, des individus, de la population d'individus de départ, qui seront regroupés deux par deux ou plus, recombinés de sorte à former de nouveaux individus, héritant des meilleures caractéristiques de leurs parents. Puis ces nouveaux individus remplaceront les individus parents les plus mauvais dans la population de solution de départ, selon un certain critère.

Ces dernières années, une large gamme d'approches évolutionnistes a été développée, nous citons : la programmation évolutionniste, la stratégie évolutionniste, les algorithmes génétiques et la programmation génétique.

2.6.1 Algorithme Génétique

Les algorithmes génétiques forment la catégorie la plus connue et la plus utilisée de la classe des algorithmes évolutionnistes. Introduits pour la première fois par J. Holland au début des années 1970. Á cette époque peu de chercheurs s'intéressaient à l'exploration de cette piste faute du coût d'exécution élevé. L'apparition des calculateurs de hautes performances au début des années 1990, a suscité un regain d'intérêt des chercheurs pour cette catégorie d'algorithmes. L'apparition du livre " Genetic Algorithms in Search Optimisition and Machine Learning " de D.E.Geoldberg, en 1989, a été le déclic pour le début d'une nouvelle ère de développement des algorithmes génétiques. Depuis, de nombreuses améliorations et

adaptations ont été apportées, allant jusqu'à dériver des méthodes des algorithmes selon le type de problèmes à traiter.

Généralement, les AG sont spécifiques pour un encodage binaire des individus, une chaine de bits sert à représenter chacun des individus. Cette chaine de bits peut être assimilée au génotype de l'individu, à partir duquel on peut calculer son phénotype qui est spécifique au problème traité. Traditionnellement la sélection des individus est proportionnelle à leurs force et tous les individus parents sont remplacés par leurs enfants. Les opérateurs de croisement et mutation sont des opérateurs élémentaires : coupure et échange de segments de bits, mutation de bits .... Il est important de préciser que les opérateurs de croisement et de mutation sont strictement indépendants du génotype et donc de la valeur de la fonction objectif.

Pour un codage réel, des adaptations spécifiques doivent être apportées de sorte à pouvoir appliquer un croisement et une mutation sans influencer la valeur de la fonction objectif.

La convergence théorique a été démontrée. Le risque de convergence prématurée, dite aussi dérive génétique, est un inconvénient majeur de la méthode, il peut être engendré par un choix ou un croisement limité ou mal conditionné. Le temps d'exécution est un autre inconvénient de la méthode. Manipuler simultanément une population d'individus et calculer la valeur de la fonction de performance pour chaque individus peut s'avérer pénalisant.

2.6.2 Stratégies Évolutionnaires

Proposées par Rechenberg et Schewefelen 1964, à la différence des algorithmes génétiques, les stratégies évolutionnaires s'appliquent aussi bien pour les problèmes continus, que pour les individus représentés par des vecteurs de réels. Soit une population initiale de taille μ, compte tenu de la fonction de fitness, on tire, de façon déterministe, un certain nombre d'individus de sorte que leurs combinaison discrète engendre des individus qui seront injectés dans la population des individus parents. Une nouvelle population de taille μ est produite par application de la procédure d'élitisme et ainsi de suite jusqu'à satisfaction du critère d'arrêt.

L'avantage de cette méthode est son efficacité, en terme de complexité temporelle. De plus, on a une sélection déterministe et l'exténuation de mauvais individus au cours des itérations. Seuls les bons individus survivront à chaque itération.

2.6.3 Programmation Évolutionniste

La programmation évolutionniste a été introduite par L.J.Fogel and al. en 1975. Ce processus consiste à copier la configuration de la population d'individus courante, dans la nouvelle population, une sélection déterministe. Les configurations résultantes sont ensuite mutées, conduisant, de façon stochastique, à de nouvelles configurations qui vont se lancer dans une compétition de survie dans la prochaine génération.

La mutation est assurée par une des 5 opérations suivantes : changer le symbole d'une sortie, changer un état de transition, ajouter ou supprimer un état et changer l'état initial.

La programmation évolutionniste a été introduite au départ pour simuler l'intelligence, qui est définie simplement par la prédiction d'une séquence de symboles appartenant à un alphabet fini à partir de séquences déjà observées.

2.6.4 Programmation Génétique

La programmation génétique est une approche récente dans la classe des algorithmes évolutionnaires. L'objectif de cette approche est de générer progressivement un programme "optimal" qui solutionne le problème considéré. Un individu est un programme, donc son exécution génère une solution. Les croisements et mutations s'effectuent en terme de programmes eux-mêmes. L'action de recherche est plutôt sur le programme et non pas sur la solution.

2.6.5 Comparaison Entre les Différentes Variantes

Une similitude de structure a été observée pour l'ensemble des méthodes citées précédemment : on définit un codage de données, on construit une population de solutions de départ qu'on

fait évoluer, itérativement, vers une population de solutions, dont les individus améliorent la valeur de la fonction objectif, jusqu'à satisfaction d'un certain critère dit critère d'arrêt.

2.7 Colonies de Fourmis(Ant Colony Optimization)

L'optimisation par colonies de fourmis est une approche inspirée du comportement collectif des fourmis. Ce comportement permet aux fourmis de minimiser la longueur du chemin entre la source de nourriture et la fourmilière. Lorsque la fourmi suit un chemin, elle dépose une trace de phéromone qui est utilisée par les fourmis suivantes. Celles-ci ont d'autant plus tendance à choisir un chemin donné suivant que la concentration en phéromone y est forte. Il en résulte que leur action collective produit un chemin qui peut être le plus court. Dans l'algorithme d'optimisation par colonies de fourmis, la construction d'une nouvelle solution est effectuée par une fourmi.

La mémoire qui guide cette recherche représente la matrice de phéromone. Celle-ci est utilisée par les fourmis pour construire de manière incrémentale les solutions. Chaque solution correspond à un ensemble de composantes. Lors du choix d'une composante de solution, la fourmi est soumise à une tendance à exploiter les informations de la matrice de phéromone, mais aussi à une tendance à utiliser des informations heuristiques [31].

Ces deux tendances sont combinées dans le processus de décision stochastique de la fourmi. Après avoir créé une solution, un dépôt de phéromone dont l'intensité dépend de la qualité de la solution est effectué. Les dépôts successifs de phéromone combinés à l'évaporation vont conduire les fourmis à produire des solutions de meilleure qualité.

Dans cette méta-heuristique, la matrice de phéromone peut être vue comme un modèle probabiliste reflétant les caractéristiques des zones prometteuses de l'espace de recherche. Dans [31], on trouve l'état de l'art sur les variantes et les applications de l'optimisation par colonie de fourmis.

2.8 Méta-Heuristiques d'optimisation par essaims particulaires

PSO pour "Particle Swarm Optimization" est une technique d'optimisation stochastique développée par Eberhart et Kennedy (1995). Elle s'inspire des déplacements collectifs observés chez certains animaux sociaux tels que les poissons et les oiseaux. En effet, ces groupes d'animaux montre des dynamiques de déplacement relativement complexes, alors qu'individuellement, ils n'ont accès qu'à des informations limitées, comme la position et la vitesse de leurs plus proches voisins. Ce type d'approche utilise une population de solutions potentielles connue sous l'appellation d'essaim, les individus sont appelés particules. Elle se base sur la collaboration des particules entre elles en échangeant des informations permettant ainsi l'émergence de comportements complexes. Chaque particule est caractérisée par sa position courante et un vecteur de changement de position (appelé vélocité). Pour influencer leurs évolutions, ces particules sont dotées d'une mémoire structurée au niveau local. Chaque particule n'évolue, à chaque itération, qu'en fonction de ses proches voisins, et non pas selon l'état global de la population à l'itération précédente ....

2.9 Classification des approches

De nombreux critères de classification de Méta-heuristiques peuvent être considérés. Dans le tableau suivant, quelques approches ont été classifiés selon la source d'inspiration, l'usage de mémoire, nature du processus (déterministe ou stochastique) ou bien le nombre de solution manipulées à chaque itération ....

2.10 Conclusion

Toutes approches présentent des points forts (avantages) et au moins un inconvenient. Parfois cet inconvenient peut être palié par l'injection d'une autre aproche ou bien par la combinaison avec une autre approche. L'idée de combinaison d'approches ouvre un grand espace de recherche, et donne naissance à un autre type d'approches dit les approches hybrides qui sont développées dans le chapitre 3.

Critères de classification	**Cas possibles**	**Exemple**
Source d'inspiration	Inspirées de la biologique(Nature)	Colonie d'abeilles
		Méthodes génétiques
	Inspirées de la physique statique	Recuit simulé
	Autres inspirations (intelligence artificielle)	Recherche taboue
Usage de mémoire	A mémoire courte	Recuit simulé
	A mémoire longue	Recherche taboue
	Sans mémoire	
Nombre de solutions manipulées à chaque itération	Une seule solution à chaque itération	Recuit simulé
		Recherche taboue
	Une population de solutions	Colonie de fourmi
	à chaque itération	Algorithme génétique
Usage de l'aléatoire	Stochastique	$k-opt$
		Recuit simulé
		Recherche taboue
	déterministe	$k-opt$
		Recuit simulé
		(usage de la chaine de Markov)
		Recherche taboue
Objectif	Constructives	Tout les algorithmes gloutons
		(le plus proche voisin ...)
	Deux phases	Algorithmes génétiques ...
	Améliorantes	Recuit simulé, Recherche taboue

TAB. 2.1 – Classification des Approches Méta-heuristique.

3

Approches Hybrides et l'approche Génétique-Taboue

3.1 Introduction

Les approches hybrides font l'objet de ce chapitre qui se compose de deux parties. La première partie comporte la présentation des types d'hybridations en général, suivit de l'exposition de quelques approches hybrides célèbres, élaborées pour la résolution de différents problèmes d'optimisation combinatoire.

Quant à la seconde partie, elle comporte la déscription détaillée de la méta-heuristique génétique et de la recherche taboue. Puis, nous présentaons l'approche Génitique-Taboue conçue pour la du la résolution du VRPTW. Nous terminons le chapitre par une conclusion dans laquelle on argumente l'interêt d'implémentation de algorithme Génitique-Taboue.

3.2 Approches Hybrides

Il existe bien d'autres approches que les méthodes exactes et Méta-heuristiques, pour la résolution des problèmes NP-DURS. Parmi celles-ci, on trouve les approches hybrides. On parle d'hybridation dès qu'il y a combinaison d'au moins deux approches dans le but de tirer profit de leurs points forts et de former une approche dont le comportement global est amélioré et donc la pérformance est augmentée. L'hybridation est un processus coopératif où la synergie est fondamentale.

Plusieurs types d'hybridations sont possibles, nous citons :

a) **Hybridation de méthodes exactes-exactes**
Parmi les problèmes traités avec ce genre d'hybridation, on trouve le problème de construction de tournées. La complémentarité de l'approche programmation par contrainte et la programmation linéaire et celle de la programmation par contraintes pour résoudre les sous-problèmes de la génération de colonnes ont permis la résolution de nombreux problèmes d'optimisation combinatoire.

b) **Hybridation de méthodes heuristiques-exactes**
Des techniques d'amélioration itérative et des Méta-heuristiques avec des structures de programmation par contraintes est une technique proposée par De Backer en 2000, le but de cette approche est d'éviter le piègeage de la recherche locale aux optimas locaux.

c) **Hybridation de méthodes heuristiques-heuristiques**
Elle représente la partie la plus large de toutes les hybridations, et particulièrement pour le VRP. Les algorithmes génétiques ou évolutionnaires hybrides de Master et *Bräysy*2007 ou les opérateurs génétiques associés aux méthodes de trajectoires de type recherche locale de de Nagata et *Bräysy*2009 sont parmi les approches de résolution efficace du VRP.

Le choix des approches à combiner est principal pour l'obtention d'une bonne coopération entre les constituants de l'approche hybride. Ainsi il est important de savoir dégager les

points forts et également les points faibles de ces dernières, afin de combler les points faibles des une par les points forts des autres. De nombreuses tentatives d'hybridation se sont avérées efficaces, en pratique, par l'amélioration de la qualité de la solution ou par la diminution du temps d'exécution par rapport aux approches qui la constituent, employées seules.

Une autre méthodologie de recherche, portant le nom d'hyper-heuristique, a été récemment utilisée pour résoudre des problèmes d'optimisation. Ce type d'approche propose de regrouper un ensemble d'heuristiques et/ou de Méta-heuristiques et de mettre en place un mécanisme capable d'identifier et de sélectionner l'heuristique la plus performante au cours du processus d'optimisation parmi l'ensemble des heuristiques considérées. La technique de sélection des heuristiques est une caractéristique très importante pour ce type d'approches.

3.3 Approches Hybrides Célèbres

3.3.1 Approche GRASP

L'approche Greedy Randomized Adaptive Search Procedure est un processus de résolution hybride. Proposée la toute première fois par Feo en 1995 et reprise par Resende en 2000. La méthode GRASP est une méthode itérative, la solution se construit à fur et à mesure d'itérer le processus.

Elle est composée de deux étapes : La première est de construction, dans cette étape on construit un ensemble d'éléments qu'on appelle la liste des candidats restreints soit par le choix de touts les éléments faisables, soit par le choix des meilleurs éléments selon une fonction de coût donné (en pratique on considère généralement la fonction objectif), ou bien on tire aléatoirement parmi les meilleurs éléments un sous-ensemble d'éléments qui constituera la liste des candidats restreints qui est mise à jour itérativement. Ensuite, à chaque itération, on choisit avec une fonction gloutonne ou aléatoirement un élément de la liste des candidats restreints, qui est rajouté à la solution en construction. Ce processus est répété jusqu'à la construction complète de la solution.

La seconde étape est d'amélioration, en effet, on sort de la première étape avec la meilleure

solution à laquelle on applique une méthode de recherche locale afin de déterminer la meilleure solution de son voisinage, ensuite on procède à son remplacement par la meilleure solution de son voisinage.

Le nombre d'itération et la taille de la liste des candidats restreints sont des paramètres principaux de la méthode GRASP. Le dernier paramètre peut être limité selon le nombre de ses éléments ou selon leurs qualité.

a) Algorithme de l'approche GRASP

Algorithme 5 Algorithme GRASP

```
1: Initialisation Inserer : taille ← dimension du probleme
2:                          Critère d'arrêt (nombre d'itération maximal)
3: s* ← ∅
4: Tant que critère d'arrêt non satisfait
5:   faire Construction
6:       s ← ∅
7:       Tant que s n'est pas complète
8:         faire Générer(Liste des condidats réstreints)
9:                  x ← aleatoire(LCR)
10:                 s ← s ⋃ x
11:                 s* ← s
12:        fait
13:        Amélioration
14:        s' ← s
15:        Améloirer(s')
16:        Si(f(s') < f(s*))
17:              s* ← s'
18:        fsi
19:  fait
```

On déduit que la méthode GRASP est constituée principalement d'une procédure gloutonne et d'une procédure de recherche locale.

b) Avantages et Inconvénients de l'Approche GRASP

C'est une méthode simple, peu paramétrée, facile à programmer et elle nécessite peu de temps supplémentaire à une méthode de recherche locale. Le seul inconvénient est l'absence de mémoire dans ce processus ce qui peut mener à l'exploration de la même zone et la reprise des mêmes résultats plus d'une fois.

3.3.2 Méta-Heuristiques d'Optimisation par Estimation de la Distribution

Les algorithmes à estimation de distribution consistent à estimer la distribution du coût des solutions en fonction des différentes composantes de solution ou variables de décision. La distribution doit faire apparaître les zones prometteuses de manière à y concentrer la recherche. La méthode débute avec un échantillonnage aléatoire de solutions. Une partie de ces solutions va permettre de faire une première estimation de la distribution de la fonction de coût. À partir de cette estimation, un nouvel échantillon de solutions est créé de manière probabiliste de façon à concentrer l'échantillon dans les zones prometteuses identifiées par l'estimation de distribution. Ce processus est réitéré afin d'affiner l'estimation de distribution et de trouver de meilleures solutions. Un état de l'art sur les algorithmes à estimation de distribution est réalisé dans [31].

EDA pour "Estimation of distribution algorithm " ou encore " Probabilistic Model Building Genetic Algorithms ". Il s'agit d'approches basées sur des modèles probabilistes (voir [31]) et inspirées des algorithmes génétiques. Ils travaillent sur une population de solutions qu'ils cherchent à améliorer en générant d'autres populations. La construction des nouvelles populations se fait grâce à l'estimation d'une distribution de probabilité (issue de la fonction objectif), associée à chaque individu de la population de l'itération précédente pour décrire la qualité des solutions. La distribution doit faire apparaître les zones prometteuses de manière à y concentrer la recherche.

La méthode commence avec un échantillonnage aléatoire des solutions. Une partie de ces solutions va permettre de faire une première estimation de la distribution de la fonction de coût. A partir de cette estimation, un nouvel échantillon de solutions est créé de manière probabiliste de façon à concentrer l'échantillon dans les zones prometteuses identifiées par

l'estimation de distribution. Ce processus est réitéré afin d'affiner l'estimation de distribution et de trouver de meilleures solutions.

a) Inconvenient d'Optimisation par Estimation de la Distribution

Le principale obstacle, lors d'utilisation d'un algorithme à estimation de distribution, est l'estimation de la distibution de probabilité.

b) Algorithme d'optimisation par Estimation de la distribution

Algorithme 6 Algorithme à estimation de distribution

Initialisation Inserer : $D_0 \leftarrow$ Génératon de M individus aleatoirement
Critère d'arrêt
$i \leftarrow 0$
Tant que critère d'arrêt non satisfait
faire $i \leftarrow i + 1$
    $D_{i-1}^{S_e} \leftarrow$ Sélectionner $N < M$ individus de D_{i-1}
    $p_i(x) = p(\frac{x}{D_{i-1}^{S_e}} \leftarrow$ Estimer
    la probabilité de distribution d'un individu d'être
    parmi les individus sélectionnés
    $D_i \leftarrow$ Echantillonner M individus depuis $p_i(x)$
fait

3.3.3 Algorithme Mémetique

Les algorithmes mémétiques, dit aussi stratégies d'intéligence artificielles, sont des algorithmes évolutionnistes dotés d'un mécanisme d'apprentissage. Le concept de "mème" est introduit par Dawkins. C'est un élément culturel, acquis par apprentissage et pouvant être retransmis de génération en génération.

Les algorithmes mémétiques sont des algorithmes évolutionnistes auxquels ont ajoute le concept de méme, c'est à dire, chaque individu fait évoluer une solution de manière indépendante par recherche locale, puis des opérateurs de croisement, mutation et sélection

sont appliqués. De plus, des choix adaptatifs d'heuristiques sont parfois mis en oeuvre.

Les algorithmes mémétiques se distinguent des algorithmes évolutionnistes par l'application d'une procédure d'amélioration locale, telle que la recherche locale ou recherche avec tabou ou bien la méthode récuit simulé, sur chaque individu de la population avec une certaine probabilité.

Le nombre d'itérations de la procédure d'amélioration dépend de l'itération courante de l'algorithme génétique et le critère d'arrêt de l'algorithme mémétique est généralement le nombre de sans itération son amélioration de la meilleure solution trouvée.

3.4 Hybridation de l'Algorithme Génétique et Recherche Taboue Appliquée au VRPTW

Avant d'entamer la discription du processus d'hybridation "Génétique-Taboue", il est important de rappeler les concepts de base de chacun des algorithmes génétiques et de la méthode de recherche avec taboue.

3.4.1 Algorithmes génétiques

Les algorithmes génétiques font partie de la classe des P-Méta-heuristiques. Les P-méta-heuristiques sont des méthodes itératives, généralement composées de deux étapes : la première étape consiste à créer ou générer une population de solutions de départ où chaque solution est considerée comme un individu. La seconde étape consiste à appliquer des mouvements de croisements et ou de mutations aux individus de la population de solution de départ. Ensuite procéder au remplacement des individus de la population de solutions courantes par les meilleurs individus issus du croisements et ou de mutations. Ce processus est réitéré jusqu'à satisfaction du critère d'arrêt, avec ou sans usage de mémoire.

On en déduit que les algorithmes génétiques sont caractérisés principalement par un codage spécifique de la solution, une population de solutions de départ, une évaluation des solutions selon un critère donné, une sélection, des stratégies de génération (croisement et ou mutation) et enfin une ré-évaluation et actualisation.

1) Représentation des Solutions

Comme toute méthode de résolution, l'algorithme génétique nécessite un codage efficace de la solution. En effet, un codage est dit efficace lorsqu'il permet la représentation de toutes les solutions possibles du problème sans la moindre restriction, garantit la connexité de l'ensemble des solutions et assure la facilité d'application des opérateurs génétiques de mutations et de croisements et enfin n'empèche pas l'évaluation de toutes solutions. De nombreux codages ont été élaborés, nous citons :

✓Codage binaire

C'est un codage simple et facile à manipuler par exemple on utilise une chaine de caractères pour représenter les jour d'accomplissement d'une tâche pendant une durée de planification donnée. Mais en pratique il est peu utilisé, car deux solutions dont les représentations diffèrent de peu doivent donner des solutions proches or ceci n'est pas toujours réalisable, par exemple le problème connu sous le nom de la barièrre de Hamming ou le 7 et 8 sont deux nombre entiers successeurs alors que la représentation 1000 du nombre entier 8 n'est obtenue qu'au bout de quatre mutations successives de la représentation 0111 du nombre entier 7.

0	0	1	0	1	1	1	0	0	1

✓Codage réel ou entier

Il consiste à modéliser la solution sous forme d'une suite de nombres réels ou entiers. Il est un peu plus élaboré que le codage binaire, car il tient compte de la précision et de la compléxité. C'est un codage proche de l'envirennnement des problèmes à résoudre. Par exemple il est utilisé pour représenter le numéro de la salle où un individu doit étre affecté.

7	4	6	1	9	2	0	5	3	18

✓Codage alphabétique ou alphanumérique

C'est un codage largement utilisé en bioinformatique constitué d'une chaine de caractères ou de caractères et de numéros. Le meilleur exmple est la representation de l'ADN.

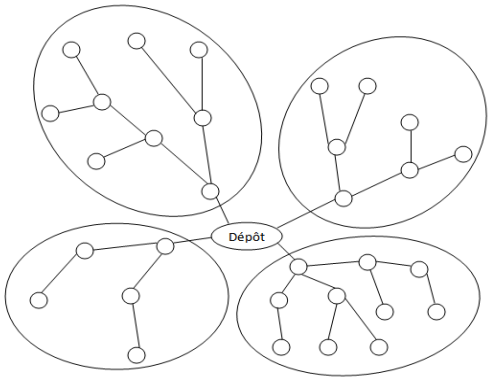

FIG. 3.1 – Codage sous forme arborescente pour le VRP

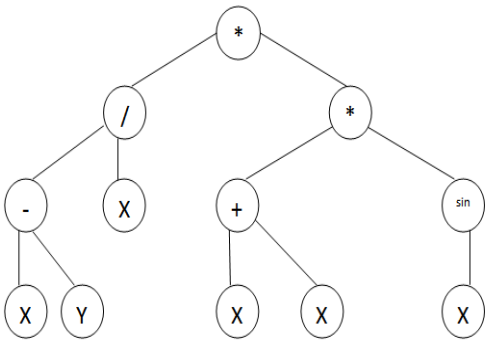

FIG. 3.2 – Codage sous forme arborescente pour la programmation génétique

H	O	N	T	C	A	Y	P	R	K

Q	X	G	M	R	Z	1	5	2	6

✓Représentation sous forme d'une structure arborescente

L'avantage de ce codage est de permettre la représentation de solutions de taille infinie. Généralement, il est adéquat pour la programmation génétique, voir Fig. 3.2. Il peut être également utilisé pour la représentation de tournées comme c'est indiqué sur la Fig. 3.1.

✓Représentation de permutation

Cette représentation convient plus aux les problèmes du VRP que'aux autres problèmes d'OC. Avec ce codage, une solution est une chaine de nombres entiers où chaque nombre représente le numéro du client servi ou à sérvir. Les avantages de ce codage sont le fait de tenir compte de l'ordre de sérvice des clients, la simplicite et la facilité d'application des opérateurs génétiques. La détermination des tournées de la solution se fait à part en tenant compte de la contrainte de capacité et ou de fenètres de temps si le problème en inclut.

7	4	6	1	9	2	10	5	3	8

Des améliorations ont été apportées à ce codage afin d'éviter l'étape de détermination des tournées en gardant la représentation précédente avec une séparation des tournées avec le chifre 0 qui représente le dépôt et comme toute tournée commence et se termine au dépôt alors touets solution commence et se termine par un 0. Ce codage est dit un codage direct, en effet toute information concernant la solution est contenue dans la représentation.

0	5	3	0	7	4	8	0	1	0

L'application des opérateurs génétiques dans ce genre de codage est moins évident que le codage précédent, lors de la recopie des composants des tournées dans la solution enfant. Les contraintes de capacité et de fenètre de temps ne sont pas forcément satisfaites même si elles le sont dans la solution parent.

Il primordial de choisir un codage adéquat au problème à résoudre car ce choix influe directement sur l'efficacité de l'algorithme génétique.

2) Population de Solutions Initiale

Vu la forte influence de la population d'individus initiale sur la convergence et l'efficacité de la méthode, une attention particulière doit être accordée à cette étape d'initialisation. En effet le critère de diversification est naturel pour une population d'individus, raison pour laquelle les algorithmes génétique sont considérés comme des méthodes d'exploration

plus que d'exploitation.

Pour déterminer une population d'individus initiale de bonne qualité, un compromis doit être établi entre la diversification et la complexité spatiale et temporelle engendrée par une forte ou une sur-diversification. Une faible diversification peut mener à une convergence prématurée.

La génération de la population initiale se fait, généralement par l'une des stratégies suivantes :

✓Stratégies de génération aléatoire

C'est une stratégie fréquemment utilisée pour les problèmes à variables continues. Par l'application d'une fonction pseudo-aléatoire ou quasi-aléatoire sur l'ensemble des solutions réalisables d'un problème, on obtient un sous-ensemble de solutions réalisables qui est la population de solutions initiale. Ce processus de génération doit tenir compte non seulement de l'indépendance de deux solutions générées successivement mais aussi de la dispersion et la diversification.
Il est à noter que la population de solutions générée par la fonction quasi-aléatoire est mieux diversifiée que celle générée par la fonction pseudo-aléatoire.

✓Stratégies de génération par diversification séquentielle

C'est une stratégit qui garantie une homogénéité parfaite de répartition de la population de solutions initiale dans l'ensemble de solutions réalisables. Partant d'une sous-population de solutions, initialisée par un tirage aléatoire d'une solution de l'ensemble de solutions réalisables, toute nouvelle solution choisie doit être à une distance définie de toutes les solutions de la sous-population de solution courante. Ce processus doit être répété jusqu'à satisfaction du critère d'arrêt.
Cette stratégie est inappropriée pour les problèmes aux domaines de solutions réalisables vastes.

✓Stratégies de génération par diversification parallèle

Avec la stratégie de génération par diversification parallèle, les solutions sont générées parallélement et de façon indépendante.

✓Stratégies de génération par des heuristiques

Toutes les heuristiques peuvent être utilisées pour initialiser la population de solutions. Le seul inconvénient de l'initialisation d'une population de solutions par une heuristique est la perte de diversification, qui pourra mener à une convergence prématurée, sans pour autant perdre en qualité de la population de solutions.

La génération de la population de solutions par une heuristique est parfois indispensable, car elle fournit des solutions réalisables satisfont les contraintes tantdis que la génération aléatoire peine à fourmer des solutions réalisables.

Quant à la taille de la population, elle est un paramètre de réglage de l'algorithme qui peut varier au cours de l'exécution de l'algorithme. Généralement elle est définie de façon empirique, compte tenu de la dimension de la solution et de l'objectif recherché.

Il est clair que la diversification est assurée par une population de grande taille. Une méthode qui aide à maintenir la diversification et la performance de l'algorithme est le regroupement des solutions en sous-populations de tailles relativement petites.

3) Évalution des solutions de la population solutions de Initiale

À toute solution créée on doit associer une valeur ou une fonction d'évaluation, dite aussi de fitness ou d'aptitude, qui permet l'évaluation de la qualité de la solution.

L'objectif de la fonction d'évaluation est de guider le processus de recherche. La fonction d'évaluation doit étre définie de sorte quelle augmente et diminue au fur et à mesure qu'elle s'approche ou s'éloigne de la "bonne" solution. De ce fait, la fonction objectif convient parfaitement à l'évaluation des solutions dans le cas d'un problème mono-objectif.

Dans le cas de problème multi-objectifs, on considère l'objectif le plus influent ou on définit une nouvelle fonction objectif dans laquelle on combinera les différents objectifs du problème avec ou sans pondération. Parfois la fonction d'évaluation doit intégrer plusieurs objectifs contradictoires alors, soit on défini une fonction d'agrégation ou de pondération selon le niveau d'importance ou de priorité, soit on considère les critères successivement dans un ordre donné, ou bien on utilise la notion de dominance de Pareto qui est largement utilisé dans les algorithmes génétique multi-objectif.

une autre façon d'évaluer les solutions, se fait en incorporant d'autres composantes en relation avec le processus d'évaluation à la fonction objectif.

Les conséquences du choix de la fonction d'évaluation des solutions sont d'une grande importance pour l'ensembles des mécanismes de l'algorithme, donc il est plus que nécessaire de faire un choix adéquat de cette fonction.

4) Sélection

De nombreuses façons de sélctions ont été définies, nous citons :

✓Sélection proportionnelle

a) **Sélection par roulette (Roulette Wheel Selection)**
Introduite par Goldberg en 1989. On associe à chaque individu de la population de solutions une probabilité de sélection qui est relative à la fonction de fitness. Pour un individu i de la population de solutions, on définie f_i la fonction de fiteness de i. Soit P_i la probabilité de sélection de i tel que $P_i = \frac{f_i}{\sum_{j=1}^{n} f_j}$. On représente la probabilité de sélection de chaque individu sur une roulette, tel que plus la probabilité est importante plus l'espace occupé sur la roulette est grand.

Le processus de sélection consiste à faire tourner la roulette face à un pointeur fixe. Á la fin du mouvement de la roulette, l'individu correspondant à la portion pointée sur la roulette esi l'individu sélectionné. Le processus est répété ainsi jusqu'à constitution complète de la population de solutions parents.

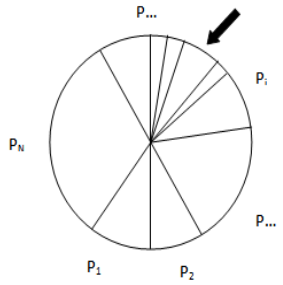

FIG. 3.3 – Sélection par roulette

L'inconvénient de la sélection par roulette est la perte de diversité de la population de solution, en effet l'individu dont la fonction de fitness est la plus élevée (le meilleur individu) sera choisi presque toujours et lorsque la distibution des probabilités de sélection sont quasi équi-probables la sélection par roulette ne favorise pas le meilleur individu.

b) **Sélection par rand (Ranking Selection)**
Proposée par Whitley en 1989, c'est une méthode qui s'effectue en deux étapes : la première est la défintion du rang de chaque individu de la population. Ce rang est obtenu par le rangement, croissant ou décroissant selon l'objectif (maximisation ou minimisation), des individus de la population, suivant leurs valeurs de fitness. En itérant ainsi le processus, pour chacun des individus d'une population de taille N, on attribue le rang N pour le meilleur individu et le rang du pire individu est au moins égal à 1.

La seconde étape consiste à appliquer une selection par roulette basée sur le rang défini en première étape au lieu de la valeur définie en a). Chaque portion de la roulette relative à un individu est proportionnelle au rang de l'individu.

c) **Sélection par échantillonnage universel stochastique**
Proposée par Whitley en 1994. Elle permet la sélection de plus d'un individu simultanément, en effet chaque individu est représenté sur une roulette où la portion de chacun est proportionnelle à la valeur de sa fonction de fitness et la roulette est dotée

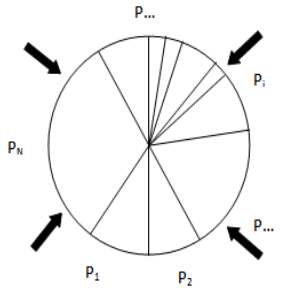

FIG. 3.4 – Sélection par rand

de plus d'un pointeur réparti sur la roulette de façon équidistante. Une rotation de roulette permet la sélection d'individus au nombre de pointeurs de la roulette.

✓Sélection par tournois (Tournament Selection)

Le principe de cette sélection consiste à tirer aléatoirement un certain nombre d'individus de la population de départ pour former un tournoi qu'on note S. Ensuite on sélectionne le meilleur individu, au sens de la fonction de fitness, parmi ceux tirés. La taille du tournoi définit le paramètre de pression de la sélection, en effet plus la taille du tournoi est grande plus la pression est élevée et la probabilité qu'un individu du tournoi soit choisi diminue et inversement.

Comme le codage et la population initiale, la sélection des individus parents influe directement sur l'efficacitée de l'algorithme de résolution, donc le choix des solutions parents doit étre fait avec rigueur afin d'assurer l'efficacitée de l'algorithme.

5) Opérateurs génétiques ou de reproduction

Une fois le codage défini, la population de solution créée et la sélection des individus parents faite, on passe à la création des individus ou solutions enfants. De nombreux opérateurs de reproduction, de croisement et ou mutation, on été développés afin de construire les solutions enfants.

a) Croisement

Le croisement est un opérateur de reproduction utilisant au moins deux individus parents pour former un individu enfant ou plus. Cet opérateur assure la transmission des génes des deux parents de façon plus au moins équilibrée. Au début de la recherche le processus de croisement permet la diversification des solutions, mais au fur et à mesure d'itérer le processus, la recherche tend vers l'intensification du fait que le processus de sélection choisit des individus proches donc les individus descendants ne seront pas loin des individus parents.

Parfois les descendants d'un croisement ont une faible capacité à reproduire, ce croisement est qualifié de létal. Pour éviter une forte proportion de croisements létaux, il faut croiser des individus qui se ressemblent. Il est important de rappeler que le croisement de deux individus identiques produit un individu enfant identique aux parents.

Le taux de croisement représente la proportion d'individus de la population des parents auqels on a appliqué l'opérateur de croisement, donc il dépend essentiellement de la taille de la population des individus parents, du taux de mutation et de la sélection. Généralement le taux de croisement varie entre 45 et 95 pour cent.

Une large gamme de croisements ont été élaborés. On cite quelques exemples de croisemnts : Order Crossover (OX), Linear Order Crossover (LOX), One Point Crossover, N Points Crossover (N-X), Partially Mapped Crossover (PMX), Uniforme Crossover (UX), Position Based Crossover (POX), Order Based Crossover (OBX), Route Crossover (RX) , Cycle Crossover (CX), Trajectory Crossover (TX), Distance Preserving Crossover (DPX), Brady's Crossover, Subsequence Exchange Crossover (SXX), Sequence Based Crossover (SBX), Route Based Crossover (RBX), Merge Crossover, Heuristic Crossover (HX), Best Cost Route Crossover (BCRX), Edge Recombination Crossover (ERX), Recombinaison d'Adjacences, Maximal Preservative Crossover (MPX). Pour plus de détails voir [37].

En pratique, les solutions enfants ne sont pas forcément réalisables. Trois propositions permettent de pallier ce problème : la première est la plus simple et la plus utilisée. Elle consiste à rejeter toute solution irréalisable. L'adaptation de la fonction objectif par le

rajout d'une pénalité afin de réduire le degré d'infaisabilté est une autre façon de régler le problème de faisabilité d'individus descendants non réalisable. La troisième façon est l'application d'une procédure de réparation ou de correction lorsqu'elles sont possibles.

b) Mutation

L'application de l'opérateur de mutation sur un individu de la population le modifie de manière complétement aléatoire. En effet la mutation permet de passer d'une solution à une autre existante dans la population ou tout à fait nouvelle qui sera inserée si elle répond à certains critères de réalisabilités. L'application de cet opérateur à une population d'individus permet la diversification et l'exploration de cette dernière, en empéchant l'algorithme de converger rapidement vers l'optimum local.

Le taux de mutation traduit le nombre d'individus mutés au sein d'une population. Un taux de mutation élevé peut causer une perte d'information et perturber la résolution, donc il est recommandé d'utiliser un taux relativement faible.

Plusieurs mutations ont été élaborées exlusivement pour le codage utilisé pour le problème du VRP.

- ✓ **Insertion ou déplacement**
  L'opérateur d'insertion tire aléatoirement une composante ou un géne d'un individu donné puis, le déplace et l'insère dans une position autre que celle de départ, tirée aussi aléatoirement.

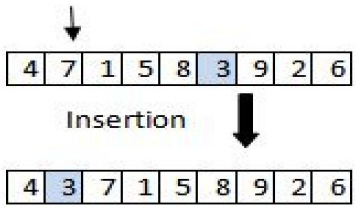

FIG. 3.5 – Opérateur d'insertion

✓ **Échange**

Le processus d'échange s'opère avec deux génes, tirés aléatoirement, d'un même individu, auxquels on applique une pérmutation ou un échange de positions.

Des variantes de l'opérateur d'échange tel que $\lambda - Opt$ et Or-Opt.

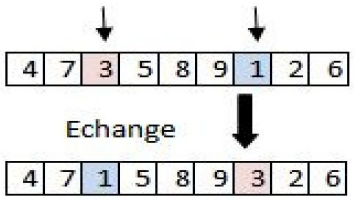

FIG. 3.6 – Opérateur d'échange

✓ **Inversion**

Avec l'opérateur d'inversion, deux positions sont pointées aléatoirement d'un même individu, ensuite on inverse la position des génes compris entre les deux positions pointées initialement de sorte que le premier géne devienne le dernier et inversement.

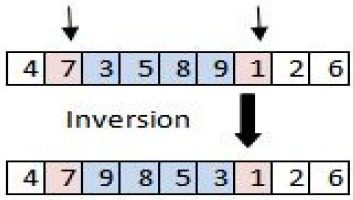

FIG. 3.7 – Opérateur d'invertion

✓ **Mélange**

L'opérateur de mélange fonctionne exactement comme l'opérateur d'inversion à la seule différence que : l'ordre des génes entre les deux positions pointées doit étre aléatoire.

6) Critère d'Arrêt

Le rôle principal du critère d'arrêt, est de mettre fin au processus de recherche de bonne solution, car le cycle de génération ne s'arrête pas tant que ce critère d'arrêt n'est pas satisfait. Il peut étre défini par la fixation préalable du nombre d'itérations du processus de recherche ou du temps maximal d'exécution. En pratique on le définit de façon empirique.

3.4.2 Méthode de Recherche Taboue (Tabu Search)

La Méthode de recherche taboue est initiée, en 1986, par Glover et Haneon indépendamment. Cette méthode de recherche locale, inspirée de l'amélioration itérative, est basée sur le principe suivant : "garder des traces du passé pour mieux s'orienter dans le futur". En effet on tente d'inculquer une portion infiniment petite du bon sens des mortels, afin de pallier le choix hasardeux du voisinage de la méthode du récuit simulé.

Un double défi doit être relevé lors de la mise au point d'une méthode taboue : en premier, on doit s'assurer que le mécanisme d'évaluation des solutions voisines soit efficace et le second diriger intelligemment la recherche en exploitant la mémoire de manière à éviter les optias locaux ou/et le cyclage (bouclage).

Le processus de la méthode de recherche taboue consiste à explorer $N(s)$, le voisinage de la solution s, dans le but d'améliorer la valeur de la fonction objectif. On autorise la dégradation de cette dernière pour échapper aux optimums locaux, mais cela risque d'induire le phénomène de cyclage. Pour y remédier on introduit une liste taboue, dans laquelle on mémorise l'ensemble des dernières solutions, qui est contrôlée par un critère d'aspiration.

La définition de l'ensemble $N(S)$, solution voisine de la solution s, se fait par apport de certaines modifications sur la solution s. Cette modification s'appelle aussi mouvement, donc à chaque application de mouvement sur une solution on définit un nouveau voisinage.

Les méthodes de recherche locale n'ont pas pour objectif l'exploration éxhaustive de l'ensemble de voisinage, ce voisinage ne se fait pas forcément de façon exhaustive, par exemple la méthode du recuit simulé se contente d'un seul élément voisin, la méthode de recherche taboue est censée faire le choix de la solution de $N(s)$ de manière judicieuse. L'accélération

de l'examen du voisinage se fait soit par la réduction ou limtation, au départ, de la taille de l'ensemble des voisins, ou par le tirage aléatoire d'un nombre de solutions voisines de s bien plus petit que $|N(s)|$ ou bien par la répartition de l'ensemble de mouvement en sous-ensembles et la consideration à chaque itération d'un des sous ensembles uniquement.

Avec la méthode de recherche taboue, F.Glover propose le concept de liste de mouvements candidats. On suppose qu'un mouvement de bonne qualité pour une solution reste bon pour l'ensemble des solutions voisines. À une itération donnée, on classe l'ensemble des mouvements par qualité décroissante, et seul les meilleurs mouvements seront considérés pendant quelques itérations ultérieures, ce qui introduit la liste des mouvement condidats. Au bout d'un certain nombre d'itérations le classement des mouvements se dégrade puisque les solutions diffèrent de plus en plus de la solution avec laquelle on a réalisé le classement des mouvements, d'ou la nécessité d'évaluation complète du voisinage, périodiquement, afin de conserver une liste de candidats convenables.

Le processus itératif peut étre également accéléré par la mise au point d'une mémoire afin d'éviter la ré-évaluation de solutions déjà traitées. En effet, on défini $T(s)$ initialement $T(s) = \varnothing$) l'ensemble ou la liste taboue et $N(s)$ l'ensemble des voisins de la solution s, ensuite pour chaque $s' \in N(s)$ on évalue $\nabla f(s) = f(s') - f(s)$:

- ✓ si $\nabla f(s) < 0$ alors $s = s'$ et $T(s) = T(s) \cup s'$(mise à jour de la liste taboue) ;
- ✓ sinon $T_{k+1}(s) = T_k(s) \cup s'$.

La liste taboue, ainsi définie représente l'historique ou la mémoire du processus, qui opère par référence à l'ancienneté, fréquence, qualité et influence. La liste taboue est composée d'un ensemble de solutions traitées qui ne doivent pas étre reprises avant une certaine durée de temps ou d'un certain nombre d'itération appelé aussi durée d'interdiction.

La duréed d'interdiction peut concerner aussi les mouvements, et elle est définie de façon empirique ou aléatoire. La détermination de la durée d'interdiction influe directement sur l'efficacité de l'algorithme. En effet, interdire un grand nombre de mouvements ou de solutions fait augmenter la probabilité de rester prisonnier dans un ensemble de solutions trés restreint, dans ce cas la recherche est dirigée par le peu de mouvements ou solutions

autorisés plutôt que par la foncton objectif. Et interdire un grand nombre de mouvements ou de solutions fait augmenter la probabilité de cycler.

Le critère d'aspiration permet de reprendre un mouvement ou une solution de liste taboue lorsque celle-ci procure une influence sur l'amélioration de la valeur de la fonction objectif de façon extraordinaire.

La durée d'interdiction peut être fixée au préalable de manière empirique ou aléatoire, comme elle peut être dynamique, elle varie selon un certain mécanisme empirique ou aléatoire.

La détermination de la taille de la liste taboue est une étape cruciale du processus de RT, elle doit être choisie judicieusement car son influence est directement liée à l'efficacité de la méthode. En effet, opérer avec une liste de taille réduite augmente le risque de cyclage en revanche cela permet une intensification de recherche car seuls quelques mouvements sont interdis. Et opérer avec une liste élargie ou de grande taille peut interdire un mouvement qui peut mener à une solution optimale par contre, ceci encourage la diversification (développement d'autres mouvements). On en déduit que pour un bon fonctionnement et un rendement efficace, de la méthode RT, un compromis entre diversification et intensification doit être établi, selon l'instance et le type du problème traité.

a) Convergence de la méthode RT Formellement, on ne peut pas parler de convergence de la méthode RT, vu que la solution change presqu'à chaque itération, mais il est intéressant de voir si on passe au moins une fois par la solution optimale globale.

Faigle and al. (1992), ont fait une similitude entre la RT et la RS qui est, rappelons nous, convergeant et ce en considérant les conditions d'interdictions probabilistes et en appliquant quelques adaptations au théorème de convergence de la méthode du RS on en déduit la convergence de la méthode du RT. Néanmoins on peut montrer que la méthode du RT dans sa forme la plus générale s'avère à la fois très efficace en pratique mais incapable de passer par l'optima global pour un certain problème mal conditionné.

b) Avantages et inconvénients de la méthode RT Avec la méthode RT on obtient

une solution, optimale ou du moins quasi optimale suite à une recherche orientée intelligemment. L'interdiction de certains mouvements a permis de limiter la recherche aux zones prometteuses uniquement.

Parmi les inconvénients de cette méthode nous trouvons :

- ✓ La détermination des paramètres liés à la liste taboue qui sont sa taille (courte ou longue mémoire, le critère d'interdiction et sa durée) qui sont déterminés : soit de manière statique fixée au préalable, soit dynamique variant à chaque itération de façon déterministe ou stochastique.

- ✓ L'interdiction de retour à certains solutions peut mener parfois à des absurdités : la déconnexion de la solution courante de la solution optimale faute d'interdiction de passage par une solution déjà visitée.

3.4.3 Approche hybride Génétique-Taboue

Dans cette partie de la thèse, on aborde l'approche hybride dite Génétique-Taboue pour la résolution d'un problème de tournées de véhicules avec fenêtre (VRPTW).

a) Justification du choix

Vu le nombre de travaux réalisés et les résultats obtenus, dans le cadre de résolution de problèmes de tournées de véhicules avec les algorithmes génétiques, une envie de mener une recherche et tantative de résolution d'un problème de tournées de véhicules avec fenêtre de temps nous est venue à l'esprit.

L'application d'une approche purement génétique pour la résolution du VRPTW présente quelques inconvénients. L'inconvénient de l'algorithme génétique réside dans la difficulté de trouver un bon équilibre d'intensification et diversification de la recherche. À cet effet l'appelle de la recherche taboue au moment de séléction des solutions parents à croiser peut apporter un équilibre d'intensification et diversification à la recherche.

Le choix des solutions parents dans un voisinage restreint ou la reprise d'une même solution parent plusieurs fois consécutivement mène à une recherche intensifiée et peu diversifiée, ainsi que l'élimination du pire des parents trés tôt risque de mener à une recherche peu diversifiée et trés tard risque de pièger le processus dans un optimum local. D'ou l'idée de la prise d'un échantillon de couples de solutions parents à croiser en même temps fait que une solution parent peut être la pire dans un couple et non dans un autre couple donc elle sera toujours conserver.

L'utilité de la liste taboue est d'éviter la reprise d'un même couple de solution dans plusieurs échantillons.

b) Algorithmes de l'approche hybride Génétique-Taboue

L'algorithme de Génétique-Tabou conçu pour la résolution du VRPTW est constitué de deux parties : la première est dite d'initialisation, elle consiste à créer la population de solutions de départ et la seconde est dite d'amélioration, elle permet de créer une nouvelle solution à partir des solutions de la première étape dans l'espoir qu'elle soit meilleure.

Algorithme 7 Algorithme Génétique-Tabou conçu

Initialisation
Insertion du Critère d'arrêt
Création de la population des solution de départ
Amélioration
Tant que critère d'arrêt non satisfait
fait **T**irer un ensemble de couples de solutions à croiser
Tant que l'ensemble de couples de solutions à croiser n'est pas vide
fait croiser les solutions parents
séléctionner les deux meilleures solutions
parmi les trois de chaque croisement
ré-insèrer les deux solutions séléctionnées
dans la population de solution de départ
fait
fait

3.5 Conclusion

L'injection d'une approche dans une autre ou la construction d'une approche à partir de deux autres permet de combiner les avantages des deux mais sans pour autant éliminer leurs inconvénients.

La mesure de la performance d'une approche quelconque ne peut être faite qu'aprés son implémentation et analyse des résultats obtenus. Le chapitre suivant est dédié à l'implémentation de l'algorithme Génétique-Taboue conçu pour la résolution d'un exemple qui est décrit également dans le chapitre suivant.

4

Implémentation de l'Algorithme Génétique-Tabou et Analyse des Résultats

4.1 Introduction

Le but de ce chapitre, est d'appliquer l'approche, hybride Génétique-Taboue, définie dans le chapitre 3 à une variante du VRP. On commence par la présentation et la modélisation du problème de tournées de véhicules avec fenêtre de temps en spécifiant l'ensemble de ses paramètres, ensuite on présente l'approche, hybride Génétique-Taboue élaborée pour la résolution du VRPTW, en détaillant les differentes étapes qui la constituent, puis on effectue une brève description de l'environnement de programmation qui est Linux ainsi que la description du langage de programmation utilisé qui est le C.

Ce chapitre est cloturé par une analyse des résultats obtenus. L'analyse est basée sur la comparaison entre les résultats obtenus sur les instances du VRPTW utilisées par Solomon et les résultats obtenus par l'approche Génétique-Tabou élaborée.

4.2 Présentation du VRPTW

Le problème de tournées de véhicules avec fenêtre de temps est parmi les variantes les plus fréquentes du VRP, en effet le VRPTW modélise de nombreux phénomènes de logistiques. En plus des définitions présentées dans le premier chapitre consernant le VRP et ses variantes, particulièrement celle du VRPTW, on précise que le problème de tournées de véhicules qui sera considéré comme exemple d'application de l'approche Génétique-Taboue définie dans le troisième chapitre est un VRPTW à contraintes de temps rigides.

La flotte de véhicules est supposée homogène de taille illimitée (le nombre de véhicules la composant est au moins égale au nombre de clients), la capacité des véhicules est identique pour l'ensemble des véhicules et elle est limitée. Les véhicules démarrent tous d'un seul dépôt et au même instant. La demande et la durée de service des clients sont connues par les distibuteurs avant de quitter le dépôt. L'objectif est de satisfaire la demande de chacun des clients à temps en parcourant une distance minimale.

4.3 Modélisation du VRPTW à contraintes de temps rigides

Le problème du VRPTW à contraintes de temps rigides se modélise exactement comme le VRP classique ou de base présenté dans le chapitre 1 auquel nous rajoutons la contrainte relative à la fenêtre de temps.

Soit $[e_i, l_i]$ la fenêtre de temps du $i^{ème}$ client. Et e_0 l'instant de départ des véhicules du dépôt 0 et l_0 l'instant du retour des véhicules au dépôt 0 aprés service. On défini par d_L^k l'instant de quitter le dernier (last) client servi, qui est L, par le véhicule k.
Pour qu'un client i soit servi par un véhicule k, il faut qu'il vérifie la formule suivante :

$$e_i \leq d_L^k + s_L^k + t_{Li} \leq l_i \qquad (4.1)$$

Nous rappelons que s_i^k est la durée de service du client i par le véhicule k et t_{Li} le temps de déplacement du client L (le dernier client servi par le véhicule k) et le client i.

Nous sous-enttendant par contraintes rigides le rejet de tout client qui ne verifie pas les contraintes de temps, contrairement au relachées qui autorise l'affectation de clients à un véhicule même s'ils ne vérifient pas les contraintes de temps, en rajoutant une fonction de pénalité.

Le codage de solutions choisi est le codage en nombres entiers (voir chapitre 2). Le critère d'arrêt du processus de génération de la population de solutions est la taille de la population de solutions qui est définie au préalable (le nombre de solutions dans la population de slutions de départ). Quand au processus d'amélioration, il est stoppé par le nombre d'échantillons à tirer. Ce nombre aussi est connu préalablement.

4.4 Approche Génétique-Taboue pour résulution du VRPTW à contraintes de temps rigides

L'Algorithme générique de cette approche est donnée dans le chapitre précédant. Dans cette partie, des precisions seront apportées sur chacune des étapes de cet algorithme.

Nous rappelons que cet algorithme est constitué de deux étapes principales :

Étape d'initialisation, dite aussi étape de génération de la population de solutions de départ. Le processus de génération de la population initiale est constitué de deux boucles imbriquées l'une dans l'autre. À chaque itération de la première boucle nous initialisons les paramètres de la solution en cours (nombre total de clients servis, longueur totale parcourue et nombre de véhicules utilisés), ensuite nous initialisons les paramètres des véhicules, puis nous tirons aléatoirement des clients de l'ensemble des clients non encors servis et si la capacité du véhicule au cours du traitement le permet, alors nous affectons ce client à ce véhicule et nous le suprimons de l'ensemble des clients non encors servi, sinon nous tirons un nouveau client. Ce processus est répété jusqu'à ce que la capacité du véhicule soit saturée. Une fois la capacité du véhicule en cours saturée nous initialisons un nouveau véhicule jusqu'à ce que l'ensemble des clients soit servi.

La taille de cette population est déterminée poroportionnellement à l'instance du problème

considéré (plus le problème est de taille importante plus la taille de la population solutions de départ doit être grande).

Le choix du tirage aléatoire de clients est fait pour assurer la diversification de la population de solution de départ de cette initialisation.

Étape d'amélioration, une fois l'étape d'initialisation est achevée, nous entamons l'étape d'amélioration qui est elle même structurée en étapes ordonnées dans le temps.

1) **Sélection**

Avant d'entamer la sélction, nous avons construit un ensemble de couples de solutions initiales à partir de l'ensemble de la population de solutions de départ qui est de taille N. Cet ensemble contient toutes les combinaisons possibles de couples de solutions de la population de départ, il est de taille $(N^2/2) - (N/2)$. La sélection consiste à tirer un échantillon ou une sous population de taille finie de couples de solutions aléatoirement. Un couple de solutions est condidat au croisement que si la différence des valeurs des fonctions objectifs des solutions qui le constituent en valeur absolue soit inférieure à un certain taux. En effet, cette condition du choix ou de sélection assure l'intensification de la recherche. C'est au moment de la sélection que nous faisons appel à la recherche taboue, en effet, lorsque nous tirons un couple de solutions nous l'insérons dans l'ensemble tabou même si le croisement n'est pas effectué.

2) **Croisement**

Le croisement choisi est caractérisé par le grand intérêt qu'il accorde à la distance qui sépare le dernier client inseré dans la solution en formation (enfant) et les clients condidats pour insertion. Ce croisement consiste à construire une solution à partir des solutions du couple sélectionné, en effet, nous débutons par l'initialisation d'un véhicule auquel nous devons affecter des clients jusqu'à ce que sa capacité soit épuisée. L'affectation des clients se fait de la manière suivante : sur les deux clients candidats pour l'affectation (un client de chaque solution du couple) nous choisissons le plus proche du dernier client affecté au véhicule en cours, si les conditions de fenêtre de temps le permettent, sinon on passe au client suivant. Une fois la capacité du véhicule épuisée, on prend un nouvean véhicle et ainsi de suite jusqu'à ce que l'enemble des

clients figure dans la solution en construction.

3) **Évaluation et ré-insertion**

On évalue la solution construite des solutions du couple sélectionné par la comparaison des valeurs des fonctions objectifs, si elle s'avère meilleure que l'une des solutions du couple sélectionné alors elle la remplace dans la population de solutions de départ et si elle est meilleure que les deux toujours en terme de valeur de la fonction objectif alors elle remplace la pire des deux.

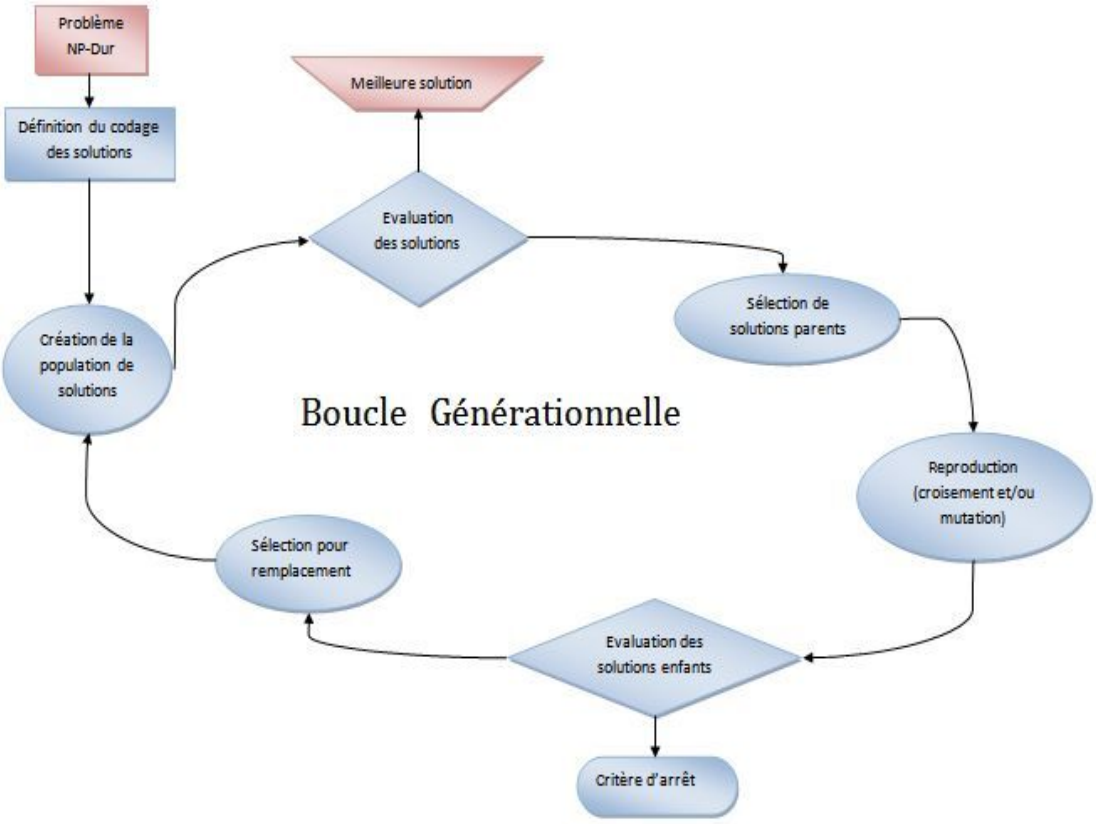

FIG. 4.1 – Boucle générationnelle

Les instances du VRPTW dont nons disposons sont les instances des benchmarks de Solomon télé-chargées sur le site :*http* : *//w.cba.neu.edu/ msolomon/problems.htm.*

Ces instances se limitent à 100 clients, elles se répartissent en six (6) catégories selon le processus suivi pour la détermination des positions des clients ainsi que la demande des clients et les fenêtre de temps. Ils font ressortir plusieurs paramètres qui influent sur le comportement des algorithmes de résolution de problèmes de routage. Les coordonnées géographique des clients et du dépôt sont générées aléatoirement pour les problèmes des catégories $R1$ et $R2$, par contre les problèmes des catégories $C1$ et $C2$ sont générées de façon à assurer le regroupement des clients en sous-ensembles ou en clusters. Les problèmes des catégories $RC1$ et $RC2$ sont structurés de sorte à garantir le positionnement aléatoire et regroupé. Les horizons de planification (fenêtre de temps) des problèmes $C1$, $R1$ et $RC1$ sont courts comparés à ceux des problèmes $C2$, $R2$ et $RC2$. L'étendu des horizons de planification (fenêtre de temps) influent sur le nombre de clients servi par chaque véhicule, en effet, $R1$, $C1$ et $RC1$ n'autorisent que quelque clients par route (environ 5) contrairement aux $R1$, $C1$ et $RC1$, qui peuvent aller au-delà de 30.

La position des clients est désignée par les coordonnées (x, y), déterminées aléatoirement ou bien de telle sorte qu'elles permettent une répartition en groupe (cluster). La distance inter-client est déterminée par le calcul de la distance euclidienne entre les différents clients. Pour le temps de déplacement, nous l'avons déterminé en divisant la distance par la vitesse. Pour tester notre algorithme nous avons considéré six (6) instances (une instance de chaque catégorie).

4.5 Implementation de l'approche hybride Génétique-Taboue

Avant d'entamer l'implémentation de l'approche hybride Génétique-Taboue, il est nécessaire de faire une présentation de l'environnement de l'implémentation.

Le système d'exploitation est un ensemble de programmes chargé de faire l'interface entre l'utilisateur et le matériel. En effet, lorsqu'on tape une commande au niveau d'un logiciel, elle est interprètée par ce dernier et transmise au système d'exploitation qui la traduit en un format compréhensible par la machine ou le matériel. De nombreux systèmes d'exploitation ont été développés, parmi lesquels on cite le WINDOWS, DOS, Mac-OS, UNIX,

Linux ....

Linux est le système qui a été choisi pour l'implémentation de l'approche qui fait l'objet de cette thèse (algotithme de résolution du VRPTW à contraintes de temps rigides). Ce choix se justufie par la liberté d'utilisation de logiciels, la gratuité de son aquisition et sa sécurité. C'est un système solide, il s'adapte aux grandes machines comme aux petites sans le moindre inconvéniant. Il a la particularité d'être un multi-utilisateurs et un multi-taches. En effet il est doté d'un système fondé sur le partage équitable des ressources. Ce système permet la gestion des utilisateurs en leur accordant des droits et espaces d'accés limités et protégés par un mot de passe pour chacun excepté un qui est consideré comme administrateur. L'administrateur a tous les droit sur le système y compris les espace protégés par les mots de passe des utilisateurs. La géstion des taches est basée principalement sur la disponibilité et la priorité. Raisons pour lesquelles il est considéré comme un système d'expoitation efficace pour les réseaux.

Linux peut être aussi efficace comme outil de programmation. Dennis M. Ritchie a initié la conception de la version du langage de programmation C portable sur linux. Ce langage de programmation est structuré, mais il reste trés prés du format machine, d'ailleur il est utilisé en premier pour la programmation du système d'exploitation Linux et UNIX.

Le langage C sous Linux est un langage universel, compact, rapide, portable, indépendant de la machine et il reste toujours moderne, c'est un langage structuré, déclaratif et récursif. La programmation avec le langage C demande les quatre outils suivants : un fichier éditeur de texte pour le cas présent l'édition du texte a été faite sur un fichier dit Emacs, un debbuger et un fichier compilateur, dans ce cas le compilateur est un terminal dans lequel est intégré le debbuger et enfin, un fichier exécutable est créé aprés compilation.

Les tests sur l'algorithme hybride Génétique-Tabou implémenté ont été effectués sur un PC Toshiba de fréquence $1,60$ GHz et 440 Mo de RAM.

4.6 Analyse des résultats

La conception de l'algorithme Génétique-Tabou a nécessité un certain nombre de paramètres. Il est important de définir l'influence de ces paramètres sur l'efficacité de cette approche. Le premier critère à évaluer est le critère d'arrêt relatif à la taille de la population de solutions de départ qui est défini par la limitation de la taille à un nombre fixe et connu qui doit être proportionnel à la taille du problème. Une tentative de définition du critère d'arrêt par la qualité de solution s'avère infructueuse car le risque de génération d'une population de taille trés réduite ou bien de taille trés importante (situations extrèmes) est trés élevé. L'influence de ce critère d'arrêt sur le temps de résolution est important. Quelque soit la variation des autres paramètres le temps de résolution reste inchangé.

Le second paramètre auquel nous nous interessons est la taille de l'échantillon des couples de solutions. L'application de l'algorithme hybride Génétique-Taboue aux six (6) instances du VRPTW choisies, 500 fois en faisant varier la taille de l'echantillon des couples de solutions de 4 à 20 couples et en fixant tous les autres paramètres donne les résultats suivants :

Remarque (5)
Nous notons par (MSPSD) la meilleure solution de la population de solutions de départ, (MSA) la meilleure solution après amélioration et (SOS) la solution obtenue par SOLOMON.

4.6.1 Influence de la taille de l'échantillon sur la qualité des solutions

✓Problème de la Catégorie R1

La variation de la taille de la polpulation de l'échantillon n'a pas d'influence sur la qualité de la solution :

- La courbe des solutions de la population de départ est stationnaire (nous ne voyons aucune tendance). Les valeurs des solutions obtenues fluctuent légèrement autour d'une moyenne de $3505,67$, la valeur maximale est de $3596,96$ et la minimale de $3377,32$.

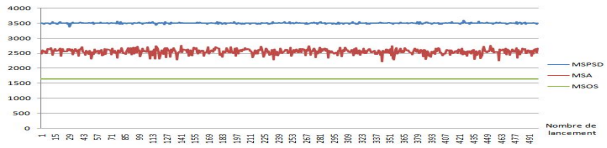

FIG. 4.2 – Influence de la taille de l'échantillon sur la qualité des solutions

– La courbe des solutions aprés amélioration est aussi stationnaire nous ne voyons aucune tendance). Les valeurs des solutions obtenues fluctuent autour d'une moyenne de $2556,76$, la valeur maximale est de $2759,11$ et la minimale est de $2220,72$.

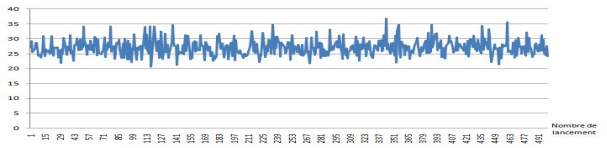

FIG. 4.3 – Évolution du taux d'amélioration selon la taille de l'échantillon sur la qualité des solutions

Le graphe suivant montre l'évolution du taux d'amélioration selon la variation de la taille de l'échantillon. Il laisse apparaitre une stationnarité de la courbe, les taux fluctuent autour d'une moyenne de 27%, le meilleur taux d'amélioration est de 37% en ramenant la meilleure solution de la population de départ de $3518,85$ à $2220,72$.

✓Problème de la Catégorie R2

La variation de la taille de la population de l'échantillon n'a pas d'influence sur la qualité de la solution :

– La courbe des solutions de la population de départ est stationnaire (nous ne voyons aucune tendance). Les valeurs des solutions obtenues fluctuent légèrement autour

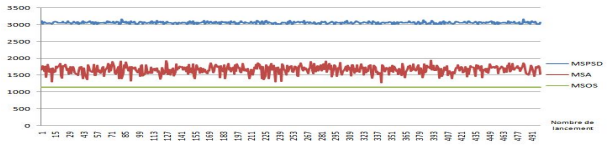

FIG. 4.4 – Influence de la taille de l'échantillon sur la qualité des solutions

d'une moyenne de $3059,05$ la valeur maximale est de $3170,43$ et la minimale de $3009,61$.

– La courbe des solutions aprés amélioration est aussi stationnaire (nous ne voyons aucune tendance). Les valeurs des solutions obtenues fluctuent autour d'une moyenne de $1671,26$ la valeur maximale est de $1949,37$ et la minimale est de $1278,90$.

Le graphe suivant montre l'évolution du taux d'amélioration selon la variation de la taille de l'échantillon. Il laisse apparaitre une stationnarité de la courbe, les taux fluctuent autour d'une moyenne de 45%, le meilleur taux d'amélioration est de 58% en ramenant la meilleure solution de la population de départ de $3049,59$ à $1278,90$.

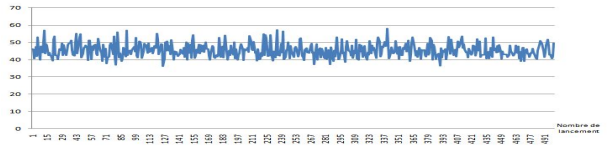

FIG. 4.5 – Évolution du taux d'amélioration selon la taille de l'échantillon sur la qualité des solutions

✓Problème de la Catégorie C1

La variation de la taille de la population de l'échantillon n'a pas d'influence sur la qualité de la solution :

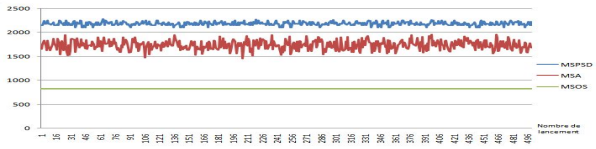

FIG. 4.6 – Influence de la taille de l'échantillon sur la qualité des solutions

- La courbe des solutions de la population de départ est stationnaire (nous voyons aucune tendance). Les valeurs des solutions obtenues fluctuent légèrement autour d'une moyenne de $2183,09$ la valeur maximale est de $2285,16$ et la minimale de $2097,87$.
- La courbe des solutions aprés amélioration est aussi stationnaire (nous voyons aucune tendance). Les valeurs des solutions obtenues fluctuent autour d'une moyenne de $1735,25$ la valeur maximale est de 1956 est la minimale est de $1452,93$.

Le graphe suivant montre l'évolution du taux d'amélioration selon la variation de la taille de l'échantillon. Il laisse apparaitre une stationnarité de la courbe, les taux fluctuent autour d'une moyenne de 20%, le meilleur taux d'amélioration est de 34% en ramennant la meilleure solution de la population de départ de $2215,08$ à $1452,93$.

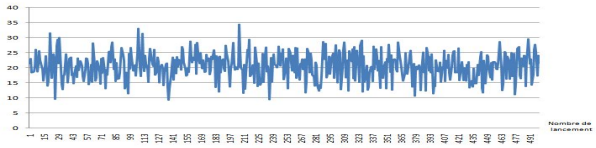

FIG. 4.7 – Évolution du taux d'amélioration selon la taille de l'échantillon sur la qualité des solutions

✓Problème de la Catégorie C2

La variation de la taille de la polpulation de l'échantillon n'a pas d'influence sur la qualité de la solution :

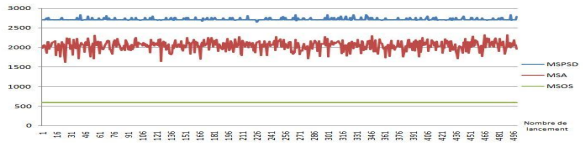

FIG. 4.8 – Influence de la taille de l'échantillon sur la qualité des solutions

- La courbe des solutions de la population de départ est stationnaire (nous voyons aucune tendance). Les valeurs des solutions obtenues fluctuent légèrement autour d'une moyenne de $2712,53$ la valeur maximale est de $2829,59$ et la minimale de $2648,20$.
- La courbe des solutions aprés amélioration est aussi stationnaire (nous voyons aucune tendance). Les valeurs des solutions obtenues fluctuent autour d'une moyenne de $2044,09$ la valeur maximale est de $2316,17$ et la minimale de $1618,95$.

le graphe suivant montre l'évolution du taux d'amélioration selon la variation de la taille de l'échantillon. Il laisse apparaitre une stationnarité de la courbe, les taux fluctuent autour d'une moyenne de 24%, le meilleur taux d'amélioration est de 40% en ramennant la meilleure solution de la population de départ de $2700,93$ à $1618,95$.

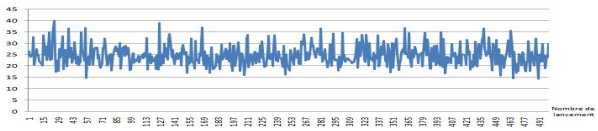

FIG. 4.9 – Évolution du taux d'amélioration selon la taille de l'échantillon sur la qualité des solutions

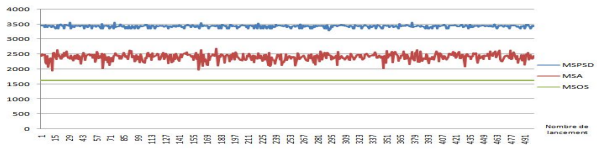

FIG. 4.10 – Influence de la taille de l'échantillon sur la qualité des solutions

✓Problème de la Catégorie RC1

La variation de la taille de la polpulation de l'échantillon n'a pas d'influence sur la qualité de la solution :

- La courbe des solutions de la population de départ est stationnaire (nous voyons aucune tendance). Les valeurs des solutions obtenues fluctuent légèrement autour d'une moyenne de $3431,78$ la valeur maximale est de 3566 et la minimale de $3288,11$.
- La courbe des solutions aprés amélioration est aussi stationnaire (nous voyons aucune tendance). Les valeurs des solutions obtenues fluctuent autour d'une moyenne de $2401,81$ la valeur maximale est de $2691,16$ est la minimale est de $1939,69$.

le graphe suivant montre l'évolution du taux d'amélioration selon la variation de la taille de l'échantillon. Il laisse apparaitre une stationnarité de la courbe, les taux fluctuent autour d'une moyenne de 30%, le meilleur taux d'amélioration est de 44% en ramennant la meilleure solution de la population de départ de $3455,06$ à $1939,69$.

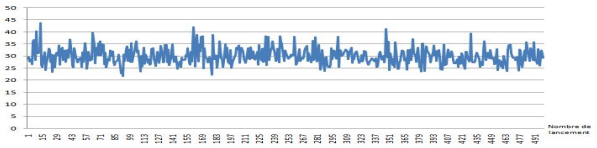

FIG. 4.11 – Évolution du taux d'amélioration selon la taille de l'échantillon sur la qualité des solutions

✓Problème de la Catégorie CR2

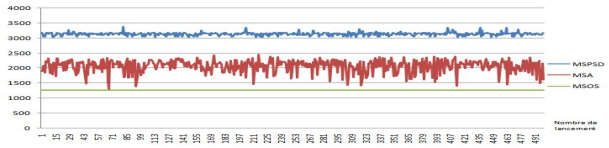

FIG. 4.12 – Influence de la taille de l'échantillon sur la qualité des solutions

La variation de la taille de la polpulation de l'échantillon n'a pas d'influence sur la qualité de la solution :

- La courbe des solutions de la population de départ est stationnaire (nous voyons aucune tendance). Les valeurs des solutions obtenues fluctuent légèrement autour d'une moyenne de $3133,16$ la valeur maximale est de $3379,53$ et la minimale de $3024,88$.
- La courbe des solutions aprés amélioration est aussi stationnaire (nous voyons aucune tendance). Les valeurs des solutions obtenues fluctuent autour d'une moyenne de $2068,62$ la valeur maximale est de $2451,64$ est la minimale est de $1288,10$.

le graphe suivant montre l'évolution du taux d'amélioration selon la variation de la taille de l'échantillon. Il laisse apparaitre une stationnarité de la courbe, les taux fluctuent autour d'une moyenne de 34%, le meilleur taux d'amélioration est de 58% en ramenant la meilleure solution de la population de départ de $3095,63$ à $1261,80$.

4.6.2 Influence du nombre du tirages de l'échantillon sur la qualité des solutions

Le dernier paramètre étudié est le nombre de fois où l'échantillon est tiré. L'application de l'algorithme hybride Génétique-Tabou aux six (6) instances du VRPTW choisies, 500 fois en faisant varier le nombre d'échantillon de couples de solutions de 4 à 20 fois, en fixant tous les autres paramètres donne les résultats suivants :

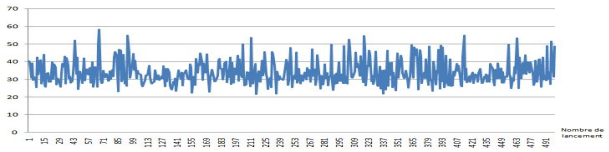

FIG. 4.13 – Évolution du taux d'amélioration selon la taille de l'échantillon sur la qualité des solutions

✓Problème de la Catégorie R1

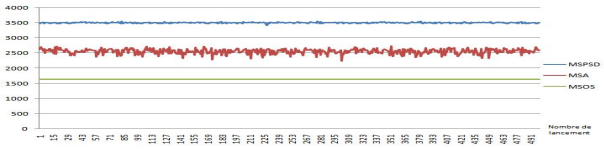

FIG. 4.14 – Influence du nombre du tirages de l'échantillon sur la qualité des solutions

La variation de la taille de la polpulation de l'échantillon n'a pas d'influence sur la qualité de la solution :

- La courbe des solutions de la population de départ est stationnaire (nous voyons aucune tendance). Les valeurs des solutions obtenues fluctuent légèrement autour d'une moyenne de $3503,89$ la valeur maximale est de $3555,24$ et la minimale de $3410,75$.
- La courbe des solutions aprés amélioration est aussi stationnaire (nous voyons aucune tendance). Les valeurs des solutions obtenues fluctuent autour d'une moyenne de $2558,89$ la valeur maximale est de $2740,06$ est la minimale est de $2233,64$.

le graphe suivant montre l'évolution du taux d'amélioration selon la variation de la taille de l'échantillon. Il laisse apparaitre une stationnarité de la courbe, les taux fluctuent autour d'une moyenne de 27%, le meilleur taux d'amélioration est de 37% en ramenant la meilleure solution de la population de départ de $3509,78$ à $2233,64$.

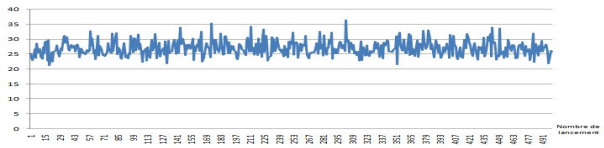

FIG. 4.15 – Évolution du taux d'amélioration selon le nombre de tirages de l'échantillon sur la qualité des solutions

✓Problème de la catégorie R2

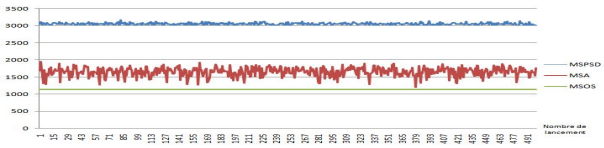

FIG. 4.16 – Influence du nombre de tirages de l'échantillon sur la qualité des solutions

La variation de la taille de la polpulation de l'échantillon n'a pas d'influence sur la qualité de la solution :

- La courbe des solutions de la population de départ est stationnaire (nous ne voyons aucune tendance). Les valeurs des solutions obtenues fluctuent légèrement autour d'une moyenne de $3058,12$ la valeur maximale est de $3170,43$ et la minimale de $3009,61$.
- La courbe des solutions aprés amélioration est aussi stationnaire (nous voyons aucune tendance). Les valeurs des solutions obtenues fluctuent autour d'une moyenne de $1654,93$ la valeur maximale est de $1957,27$ et la minimale de $1178,56$.

le graphe suivant montre l'évolution du taux d'amélioration selon la variation de la taille de l'échantillon. Il laisse apparaitre une stationnarité de la courbe, les taux fluctuent autour d'une moyenne de 46%, le meilleur taux d'amélioration est de 62% en ramenant la meilleure solution de la population de départ de $3093,04$ à $1178,56$.

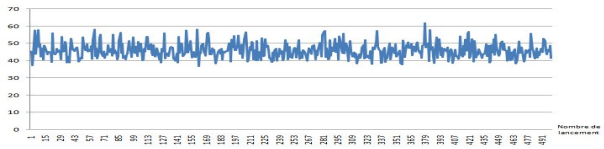

FIG. 4.17 – Évolution du taux d'amélioration selon le nombre de tirages de l'échantillon sur la qualité des solutions

✓Problème de la catégorie C1

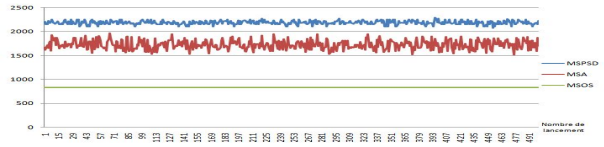

FIG. 4.18 – Influence du nombre de tirage de l'échantillon sur la qualité des solutions

La variation de la taille de la polpulation de l'échantillon n'a pas d'influence sur la qualité de la solution :

- La courbe des solutions de la population de départ est stationnaire (nous ne voyons aucune tendance). Les valeurs des solutions obtenues fluctuent légèrement autour d'une moyenne de $2184,49$ la valeur maximale est de $2291,44$ et la minimale de $2070,20$.
- La courbe des solutions aprés amélioration est aussi stationnaire (nous voyons aucune tendance). Les valeurs des solutions obtenues fluctuent autour d'une moyenne de $1724,87$ la valeur maximale est de $1967,59$ et la minimale de $1511,24$.

le graphe suivant montre l'évolution du taux d'amélioration selon la variation de la taille de l'échantillon. Il laisse apparaitre une stationnarité de la courbe, les taux fluctuent autour d'une moyenne de 21% le meilleur taux d'amélioration est de 32% en ramenant la meilleure solution de la population de départ de $2255,74$ à $1524,70$.

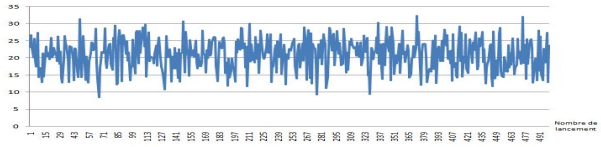

FIG. 4.19 – Évolution du taux d'amélioration selon le nombre de tirages de l'échantillon sur la qualité des solutions

✓Problème de la catégorie C2

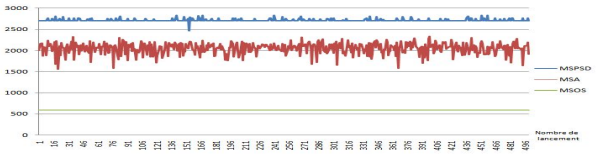

FIG. 4.20 – Influence du nombre de tireges de l'échantillon sur la qualité des solutions

La variation de la taille de la polpulation de l'échantillon n'a pas d'influence sur la qualité de la solution :

- La courbe des solutions de la population de départ est stationnaire (nous ne voyons aucune tendance). Les valeurs des solutions obtenues fluctuent légèrement autour d'une moyenne de $2712,35$ la valeur maximale est de $2828,94$ et la minimale de $2459,94$.
- La courbe des solutions aprés amélioration est aussi stationnaire (nous voyons aucune tendance). Les valeurs des solutions obtenues fluctuent autour d'une moyenne de $2051,50$ la valeur maximale est de $2336,47$ et la minimale de $1557,78$.

le graphe suivant montre l'évolution du taux d'amélioration selon la variation de la taille de l'échantillon. Il laisse apparaitre une stationnarité de la courbe, les taux fluctuent autour d'une moyenne de 24% le meilleur taux d'amélioration est de 42% en ramenant la meilleure solution de la population de départ de $2699,30$ à $1557,78$.

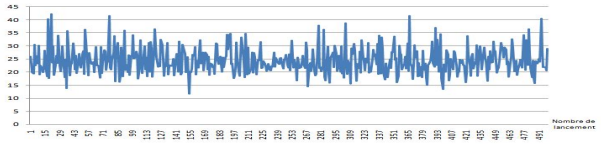

FIG. 4.21 – Évolution du taux d'amélioration selon le nombre de tirages de l'échantillon sur la qualité des solutions

✓Problème de la catégorie RC1

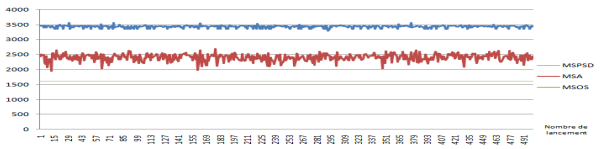

FIG. 4.22 – Influence du nombre de tirages de l'échantillon sur la qualité des solutions

La variation de la taille de la polpulation de l'échantillon n'a pas d'influence sur la qualité de la solution :

- La courbe des solutions de la population de départ est stationnaire (nous ne voyons aucune tendance). Les valeurs des solutions obtenues fluctuent légèrement autour d'une moyenne de $3433,35$ la valeur maximale est de $3573,32$ et la minimale de $3342,76$.
- La courbe des solutions aprés amélioration est aussi stationnaire (nous ne voyons aucune tendance). Les valeurs des solutions obtenues fluctuent autour d'une moyenne de $2402,30$ la valeur maximale est de $2680,33$ est la minimale est de $1713,97$.

le graphe suivant montre l'évolution du taux d'amélioration selon la variation de la taille de l'échantillon. Il laisse apparaitre une stationnarité de la courbe, les taux fluctuent autour d'une moyenne de 30%, le meilleur taux d'amélioration est de 50% en ramennant la meilleure solution de la population de départ de $3455,06$ à $1713,97$.

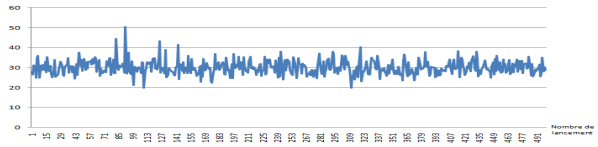

FIG. 4.23 – Évolution du taux d'amélioration selon le nombre de tirages de l'échantillon sur la qualité des solutions

✓Problème de la catégorie RC2

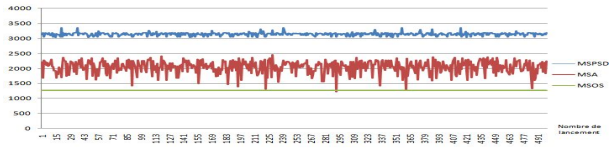

FIG. 4.24 – Influence du nombre de tirages de l'échantillon sur la qualité des solutions

La variation de la taille de la polpulation de l'échantillon n'a pas d'influence sur la qualité de la solution :

- La courbe des solutions de la population de départ est stationnaire (nous ne voyons aucune tendance). Les valeurs des solutions obtenues fluctuent légèrement autour d'une moyenne de $3135,49$ la j,valeur maximale est de $3354,73$ et la minimale de $3024,88$.
- La courbe des solutions aprés amélioration est aussi stationnaire (nous ne voyons aucune tendance). Les valeurs des solutions obtenues fluctuent autour d'une moyenne de $2045,90$ la valeur maximale est de $2449,55$ et la minimale de $1218,68$.

le graphe suivant montre l'évolution du taux d'amélioration selon la variation de la taille de l'échantillon. Il laisse apparaitre une stationnarité de la courbe, les taux fluctuent autour d'une moyenne de 34%, le meilleur taux d'amélioration est de 60% en ramennant la meilleure solution de la population de départ de $3033,68$ à $1218,68$.

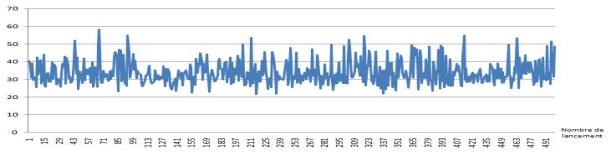

FIG. 4.25 – Évolution du taux d'amélioration selon le nombre de tirages de l'échantillon sur la qualité des solutions

Remarque (6)
Il est important de signaler que les solutions obtenues dans la phase d'initialisation sont trop éloignées des solutions finales pour tous les problèmes ($R1$, $R2$, $C1$, $C2$, $RC1$ et $RC2$) par contre le taux d'amélioration varie d'un problème à un autre, le meilleur taux (62%) à permis de ramener la valeur de la meilleure solution, de la population de solutions, de $3093,04$ à $1178,56$.

4.6.3 Récapitulation

Le tableau suivant représente l'ensemble des résultats obtenus :
Ce tableau laisse apparaitre que les résultats obtenus pour les problèmes de catégorie $R2$ et $RC2$ sont meilleurs que ceux des restes. Rappelons nous que ces problèmes sont caractérisés par la génération aléatoire de leurs paramètres en plus d'un étendu des fenêtres de temps large. Les résultats de $CR2$ sont meilleurs que ceux de référence en terme de longueur totale parcourue par l'ensemble des véhicules et même en terme de nombre de véhicules mobilisés. Par contre les résultats obtenus pour le R2 sont moins bons que ceux de référence vu que la longueur parcourue est supérieure(1178 par rapport à 1143) et le nombre de véhicules est aussi supérieur (9 par rapport à 8).

Les résultats obtenus pour les problèmes caractérisés par un regroupement des clients en sous ensembles ne sont pas satisfaisants. Les longueurs parcourues sont presque doublées ($C1 : 827 \rightarrow 1452$ et $C2 : 589 \rightarrow 1557$), parfois même triplées, et le nombre de véhicules mobilisé a lui aussi augmenté ($C1 : 10 \rightarrow 13$ et $C2 : 3 \rightarrow 13$).
Legende

GT l'approche Génétique-Taboue
KDMSS-N.Kohl, J.Desrosier, O.B.G.Madsen, M.M.Solomon and F. Soumis, "2-path Cust for the Vehicle Routing Problem with Time Windows", ***Transportation Science***,Vol.33(1)101-116(1999).
KLM-B.Kallechauge, J.Larsen and O.B.G.Madsen, "Lagrangean duality and non-differentiable optimization applied on routing with time windows-experimental results", Internal report IMM-REP-2000-8, Department of Mathematical Modeling, Technical l'University of Denmark, Lyngby, Denmark, 2000.
CR-W.Cook and J.L.Rich," A parallel cutting plane algorithm for the vehicle routing problem with time windows", Working Paper,Computational and Applied Mathematics, Rice University,Houston,TX, 1999.
LMS Longueur de la tournée de la meilleure solution.
MS Meilleure solution.

4.7 Évalution de l'approche Génétique-Taboue

Le temps de résolution n'a pas pu être évalué, car les critères d'arrêts utilisés dépendent du temps. La variation du temps de résolution a été infinitisimale, autrement dit le temps de résolution est constant quelque soit la variation des paramètres du problème ou de l'approche.

Les solutions obtenues dans la première phase fluctuent au tour d'une valeur moyenne fixe, aucune tendance n'a été observée, ce qui signifie que les résultats sont stationnaires. Il en est de même pour les solutions obtenues par la deuxième phase (phase d'amélioration). Seulement les valeurs de la première phase sont trop éloignées des solutions de référence par contre le taux d'amélioration est important pour l'ensemble des problèmes même s'il est variable d'un problème à un autre.

4.8 Conclusion

Nous concluons que le langage de programmation choisi (C sous Linux) a permi l'implémentation de l'algorithme Génétique-Tabou facilement. Le logiciel obtenu assure la résolution des problèmes de tournées de véhicules avec fenêtre de temps en fournissant des solutions

admissibles en un temps raisonnable.

Inst. VRPTW	**Cap. véh.**	**App. rés.**	**LMS**	**Nb véh**	**LMS (GT)**	**Nb véh MS (GT)**	**MS (GT)**
R1	200	KDMSS	1637,70	20	2220,73	30	80,78,77,74,70,68,60,58,89,93/11,13,25 ,100/10,17,48/79/12,16,18,9,20,24,91/76 /75,73,1/72,67,35/14,15,19,8,22,26,32, 4/81,71,66,57/69/23,30,34,37,54/82,65, 56/27,28,29,38,40,41,43,46,55/83,50,96/ 84,53/64,85,49,97/21,7,86,6,3/33,36,31, 44,61,51,87,94/63,90/59/42/52/45/62,39, 47/88/5,95/92/2,99/98
R2	1000	KLM	1143,20	8	1178,57	9	66,68,60,70,58,48,77,80,25,89,93,100/64 ,62,61,57,56,55,54,43,74,37,35,32,24,17 ,91,13,4,1/59,67,71,53,51,50,46,34,26, 20,10,96,97/65,63,69,72,73,75,49,6,3/52 ,76,78,79,81,41,40,84,85,86,22,90,94/47 ,44,38,30,23,19,87,18,9,8/39,45,42,36, 33,82,31,29,83,88,21,16,12,11,7,99/28, 27,15,95,14,5,98,2/92
C1	200	KDMSS	827,30	10	1452,93	15	2,4,1,21,47/97,93,91,89,75,69,49/5,6,9, 12,22,34,52/98,96,95,94,92,88,85,80,79, 66/3,7,8,10,11,14,16,23,26/90,87,86,84, 82,77,73,68,99,50/13,15,19,28,36,39,48, 59/81,78,76,74,72,70,64,61/17,18,20,24, 27,30,38,45,51/83,71,62,60,58,100/25,29 ,35,37,46/67,65,63,56,53,44/31,32,40/57 ,55,54,33/41,42,43
C2	700	KLM et CR	589,10	3	1557,79	14	2,4,1,21,47/97,93,91,89,75,69,49/5,6,9, 12,22,34,52/98,96,95,94,92,88,85,80,79, 66/3,7,8,10,11,14,16,23,26/90,87,86,84, 82,77,73,68,99,50/13,15,19,28,36,39,48, 59/81,78,76,74,72,70,64,61/17,18,20,24 ,27,30,38,45,51/83,71,62,60,58,100/25, 29,35,37,4667,65,63,56,53,44/31,32,40/ 57,55,54,33/41,42,43
RC1	200	KDMSS	1619,80	15	1713,97	14	2,4,13,17,24,25,70,80/3,6,10,20,32,35, 48,60,91,93/5,7,8,1,43,54,100/9,18,26, 34,37,55,74,58,77/11,12,15,16,22,41,46, 49/14,19,21,23,30,40,50,56/27,28,29,38, 51,53,68,66,89/31,33,57,73,78,96,97/36, 44,47,61,67,84,79/39,42,45,52,62,71,75, 87,94/59,63,64,76,81,85,86/69,72,82,88, 90,99/65,83,95,98/92
RC2	1000	KLM	1261,80	9	1219,69	8	4,13,17,24,25,48,58,70,80,93,100/3,6,10 ,20,26,32,35,37,43,54,55,60,1,74,77,89, 91/5,7,8,9,18,22,34,46,50,56,66,68,96/ 12,11,15,16,19,21,23,29,30,38,40,41,49 ,97/14,27,28,31,33,44,51,53,57,78,79,81 ,84,85,87,94/36,39,42,2,47,52,61,62,64, 67,71,73,75,76,86,90,99/45,69,82,83,88, 98/59,63,65,72,92,95

TAB. 4.1 – Tableau récapitulatif des résultats obtenus suit à la réitération de l'algorithme GT 1000 fois pour chacune des instances

Conclusion générale

Le travail élaboré dans cette thèse porte essentiellement sur les problèmes des tournées de véhicules et précisément sur les problèmes de tournées de véhicules avec fenêtre de temps, qui permettent la modélisation de nombreux phénomènes de la logistique et de l'économie.

L'objectif visé au départ de cette thèse était la construction d'une approche de résolution du VRPTW aprés une présentation d'un état de l'art du VRP et de ses approches de résolution.

Les deux premiers chapitres ont été consacrés intégralement aux définitions et à la présentation de l'état de l'art du VRPTW et de ses méthodes de résolution. Deux types de méthodes de résolution des problèmes NP-DUR tel que le VRPTW, ont été abordées : les méthodes exactes qui assurent l'optimalité de la solution mais pour des problèmes de petites instances (taille limitée), et les approches heuristiques et méta-heuristiques qui fournissent des solutions appréciables en un temps raisonnable à des problèmes de n'importe quelle instance.

Les approches prosopées dans le troisième chapitre ont la particularité de combiner les approches dans le but d'en construire une qui réunit les avantages des approches qui la constituent et sera plus performante qu'elles. L'approche hybride qui fait l'objet de cette thèse est formée de l'algorithme génétique dans lequel nous injectons une recherche taboue. En effet, dans la première phase qui est la phase d'initialisation, les solutions de départ sont générées aléatoirement, ensuite l'injection d'une recherche taboue se fait dans la phase de sélection de solutions parents, condidats aux croisements dans l'algorithme génétique pour permettre à ce dernier une meilleure diversification car les solutions parents traitées

ne seront plus reprises pour un autre croisement.

Les résultats expérimentaux de cette approche montrent que la première phase de cet algorithme assure une diversification des solutions mais sans pour autant garantir la qualité des solutions. En effet, comparées aux résulats obtenus sur les benchmark de Solomon nous voyons que les solutions sont éloignées de celles des benchmark. Avec la phase d'amélioration nous arrivons à des solutions plus proches de celles des benchmark de Solomon et même plus performantes pour le cas des problèmes dont les paramètres sont générés aléatoirement.

Pour finir nous suggérons tout d'abord, une amélioration de l'approche hybride GT conçue par :

✓ La génération d'une population de solutions de départ par une approche dotée d'un mécanisme de sélection à fin de tenir compte de la distance inter-clients.

✓ L'adaptation du croisement choisi aux problèmes dont les clients sont regroupés en sous ensembles en relâchant les contraintes.

✓ La considération de plus d'un objectif : le premier consiste à minimiser la longueur totale parcourue par l'ensemble des véhicules mobilisés et le second consiste à minimiser le nombre total de véhicule mobilisés.

En suite nous envisageons le developpement d'autres approches hybrides telque les heuristiques et/ou méta-heuristiques et les méthodes exactes de sorte que les critères d'arrêts des approches ne dépendent plus du temps, afin de mesurer le temps de résolution.

Bibliographie

[1] Adrien Goeffon, *Nouvelles heuristiques de voisinage et mémétiques pour le problème Maximum de Parcimonie*. Université d'Angers, novembre 2006.

[2] Alexandre Bouthillier, *Modéliasation UML pour une architecture coopérative appliquée au problème de tournée de véhicules avec fenêtre de temps*. Université de monréal, Faculté des études supérieures, avril 2000.

[3] A.Belahcene*, *C facile*, janvier 2012.

[4] Amir Nakib, *Conception de Méta-heuristiques d'Optimisation pour la Segmentation d'Images. Application à des images biomédicales*. Université de Paris $12 - Val$ de Marnre, décembre 2007.

[5] Arianne Trudeau, *Planification des Tournées de Véhicules pour l'Approvisionnement de dépanneurs*. Université de Québec à Monréal, février 2008.

[6] Boris Bontaux, *Techniques Hybrides de recherche exacte et approchée : application à des problème de transport*. Université d'Avignon et des Pays de Vaucluse, décembre 2008.

[7] Catrine Mancel, *Modélisation et Résolution de Problèmes d'Optimisation Combinatoire Issus d'Application Spatiales*. Institut National des Sciences Appliquées de Toulouse, juin 2004.

[8] Cesar Rego et Catrine Roucairol, *Le Problème de Tournées de Véhicules : étude et résolution approchée*. INRIA, février 1994.

[9] David Meignan, *Une Approche Organisationnelle et Multi-Agentpour la Modélisation et l'Implantation de Méta-Heuristiques Application aux problèmes d'optimisation de réseaux de transports*. Université de Franche-Comté et l'Université de Technologie de Belfort-Montbéliard, décembre 2008.

[10] Denis Lapoire, *Graphes et algorithmes*. ENSEIRB, janvier 2007. [http ://www.enseirb.fr].

[11] David Pisinger Stefan Ropke, *A general heuristic for vehicle routing problems*. University of Copenhagen, DIKU-Department of Computer Science et Universitetsparken 1 February 2005. [E-mail : pisinger, sropke@diku.dkiversité].

[12] David Applegate, Robert Bixby, Vaések Chvéatal and William Cook *FINDING CUTS IN THE TSP*. AT et T.Bell Labs, Rice University, Rutgers University and Bellcore, august 1994.

[13] Emilie Grellier, *Optimisation des tournées de véhicules dans le cadre de la logistique inverse : modélisation et résolution par des méthodes hybrides*. Université de Nantes Faculté des Science et des Techniques, janvier 2008.

[14] Emmanouil E.Zachariadis, Christos D.Tarantilis, Christos T.Kiranoudis, *A Guided Tabu Search for the Vehicle Routing Problem with two-dimensional loading constraints*. National Technical University of Athens, Department of Process Analysis and Plant Design, Department of Chemical Engineering, Athens University of Economics and Business, Department of Management Science and Technology, European Journal of Operational Research 195 (2009) 729–743, Available online, 21 November 2007.

[15] Elsa Bousquet, *Optimisation non linéaire et application au réglage d'un réseau de téléscopes*. Université de Limoges, décembre 2009.

[16] El-Djillali Talbi, *Sélection et réglage de paramètres pour l'optimisation de logiciels d'ordonnancement industriel*. Institut National Polytechnique de Toulouse, novembre 2004.

[17] E.Grellier, P.Dejax, et N.Jussien, *Problème de tournées de collectes et livraisons multi-périodique : résolution grâce au GRASP*. École des Mines de Nantes, IRCCYN, et École des Mines de Nantes, LINA, 2007. [E-mail emilie.grellier,pierre.dejax@emn.fr narendra.jussien@emn.fr].

[18] Eric D.Taillard-Luca M.Gambardella et Michel Gendreau-Jeau-Yves Potvin, *La programmation à mémoire adaptative ou l'évolution des algorithmes évolutifs*. IDSIA et CRT and DIRO, Université de Monréal. [E-mail eric,luca@idsia.ch , michelg@crt.umonreal.ca et potvin@iro.umonreal.ca].

[19] E.G.Talbi*, *Metaheuristics from design to implementation*. University of Lille-CNRS-INERIA.

[20] George Ioannou, Manolis Kritikos and Gregory Prastacos, *A problem generator-solver heuristic for vehicle routing with soft time windows*. Athens University of Economics and Business, Management Sciences Laboratory, Graduate Program in Decision Sciences. Omega 31 (2003) 41 – 53, october 2002.

[21] Gerardo Berbeglia, *Complexity Analyses and Algorithms for Pickup and Delivery Problems*. HEC Monréal Affiliée à l'Université de Monréal, octobre 2009.

[22] Goran Martinovic, Ivan Aleksi, and Alfonzo Baumgartner, *Single-Commodity Vehicle Routing Problem with Pickup and Delivery Service*. University of Osijek, Faculty of Electrical Engineering, Josip Juraj Strossmayer.Correspondence should be addressed

to Goran Martinovic, décembre 2008. [goran.martinovic@etfos.hr].

[23] Guenther Fuellerer, Karl F.Doerner, Richard F.Hartl, Manuel Iori, *Ant Colony Optimization for the Two-dimensional Loading Vehicle Routing Problem*. Université de Vienna, Department of Business Administration et University of Modena and Reggio Emilia,DISMI. [E-mail Guenther.Fuellerer, Karl.Doerner, Richard.Hartl@univie.ac.at, manuel.iori@unimore.it].

[24] Haibing Li and Andrew Lim, *Local search with annealing-like restarts to solve the VRPTW*. National University of Singapore, Department of Computer Science. European Journal of Operational Research 150 (2003) 115–127, mai 2002.

[25] Hassan Saoud, *Étude des Problèmes Unilatéraux :analyse des récession, stabilité de lyapunov et application en électronique et en mécanique*. Université de Limoges, juin 2009.

[26] H.Housroum, T.Hsu, R.Dupas, G.Goncalves, *Une approche génétique "en ligne" pour la gestion de tournées dynamique*. Laboratoire de Génie Informatique et d'Automatique de l'Artois (LGI2A), Université d'Artois.

[27] Helene Toussaint, Christophe Duhamel, Philippe Lacomme, *Une métaheuristique hybride de type GRASP x ELS pour le 3L-CVRP*. Université Clermont-Ferrand II, LIMOS. [E-mail toussain, christophe.duhamel, lacomme@isima.fr].

[28] Isabelle Devarenne, *Etude en Recherche Locale Adaptative pour l'Optimisation Combinatoire*. Université de Franche-Comté et l'Université de Technologie de Belfort-Montbéliard, novembre 2007.

[29] Imen Harbaoui Dridi, *Optimisation Heuristique pour la Résolution du M-PDPTWR*. Ecole Centrale de Lille et l'Université de Tunis El Manar, décembre 2010.

[30] J.N.MacGregor and T.Ormerod, *Human performance on the traveling salesman problem.* Loughborough University of Technology, Loughborough, England. Perception and Psychophysics 1996, 58 (4), 527–539.

[31] J.Dréo A.Pétrowski P.Siarry E.Taillard, *Métaheuristiques pour l'optimisation difficile.* Edition EYRLLES (deuxième tirage 2005).

[32] Jin-Kao Hao, Philippe Galinier, Michel Habib, *Méta-haheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes.* Université d'Angers, LERIA, U.F.R. Sciences, Ecole des Mines d'Alès, LGI2P, URA CNRS, LIRMM et Revue d'Intelligence Artificielle, année 1999. [E-mail Jin-Kao.Hao@univ-angers.fr, galinier@eerie.fr et habib@lirmm.fr].

[33] Jean-François Cordeau, Gilbert Laporte et Stefan Ropke, *Recent Models and Algorithms for One-to-One Pickup and Delivery Problems.* Canada Research Chair in Logistics and Transportation, E-mail cordeau@crt.umontreal.ca et Canada Research Chair in Distribution Management, april 2007. [E-mail gilbert,sropke@crt.umontreal.ca].

[34] Jean-François Cordeau, *A Branch-and-Cut Algorithm for the Dial-a-Ride Problem.* Canada Research Chair in Distribution Management, HEC Montréal, october 2003.

[35] K.C.Tan, T.H.Chew and L.H.Lee, *A hybrid multiobjective evolutionary algorithm for solving truck and trailer vehicle routing problems.* European Journal of Operational Research 172, 855–885, 2006.

[36] Laetitia, *Méta-Heuristique Coopératives : du déterminsiste au stochastique.* Université de Lille 1, septembre 2010.

[37] Mais Hadj-Rachid, Christelle Bloch , Wahiba Ramadane-Cherif, Pascal Chatonnay, *Différents opérateurs évolutionnaires de permutation : sélections, croisements et mutations.* Laboratoire d'Informatique de l'Unversité de Franche-Comte EA 4269

(LIFC), juillet 2010.

[38] Marc Sevaux, *Méta-heuristiques Stratégies pour l'optimisation de la production de biens et de services*. UVHS, juillet 2004.

[39] M.Sevaux and Sorensen, *MA—PM : une nouvelle méta-heuristique hybride*. Université de Valentiennes CNRS E-mail Marc.Sevaux@univ-valentiennes.fr et Université d'Anvers Faculty of Application Economics. [E-mail Kenneth.Sorensen@ua.ab.be].

[40] Michel Nabaa, Besma Zeddini et Pierrick Tranouez, *Approche décentralisée pour résoudre le problème de transport à la demandes*. LITIS, Schedae 2007. [E-mail Michel.Nabaa@gmail, Besma.Zeddini@gmail et Pierrick.Tranouez@gmail].

[41] M.W.P.Savelsbergh and M.Sol, *The General Pickup and Delivery Problem*. School of Industrial and Systems Engineering Georgia Institute of Technology and Eindhoven University of Technology, Department of Mathematics and Computer Sciencersité.

[42] Mark Schoenauer, *Les algorithmes évolutionnaires : état de l'art et enjeux*. Projet Fractales, Inria Rocquencourt (France), Available online at the [URL http ://algo.inria.fr/seminars/]. or Algorithms Seminar 2001–2002, F. Chyzak (ed.), INRIA, (2003), pp. 113–118, octobre 2001.

[43] Maria Soto, André Rossi, Marc Sevaux, *Méta-heuristiques pour l'allocation de mémoire dans les systèmes embarqués*. Université de Bretagne-Sud. Lab-STICC. [E-mail maria.soto, andre.rossi, marc.sevaux@univ-ubs.fr] .

[44] Michel Gendreau, Alain Hertz et Gilbert Laporte, *Tabu Search Heuristic for the Vehicle Routing Problem*. Centre de recherche sur les transports Université de Montréal et École Polytechnique Fédéralde e Lausanne,Stable. [URL : http ://www.jstor.org/stable/2661622] and Source : Management Science, Vol. 40, No. 10 pp 1276-1290, october 1994.

[45] Mahdi Souid, Said Hanafi et Frederic Semet, *Méthodes à deux phases pour le problème de tournées de véhicule avec contraintes d'accessibilité*. Université Valenciennes et du Hainaut Cambresisle Mont Houy, Laboratoire d'Automatique de mécanique et d'informatique Industrielles et humaines, septembre 2010. [E-mailmahdi.souid, said.hanafi, frederic.semet@uni-valenciennes].

[46] Marie-Claude Boldic, Jacqueq Renaudet Fayez F.Boctor, *Heuristique pour le Problème de tournée de véhicules avec flotte limitée et transporteur externe* . Université Laval Québec Canada, Centre de recherche sur les Technologies de l'Organisation Réseau, janvier 2006.

[47] Narendra Jussien et Christelle Guéret, *Utilisation du backtrack intelligent dans un branch-and-bound Application au probl'eme d'Open-Shop*. École des Mines de Nantes, septembre 2010. [E-mail fNarendra.Jussien, Christelle.Gueretg@emn.fr Université].

[48] Nacima Labadie, Jan Melechovsky et Christian Prins, *Recherche Locale Itérée pour le TSP avec profits*. Université de Technologie de Troyes 12, ICD-LOSI. [E-mail nacima.labadie,jan.melechovsky,christian.prins@utt.fr].

[49] Nguyen Thanh Tuan, *Problèmes de tournées de véhicules avec livraisons divisibles*. Institut de la Francophonie pour l'Informatique, Hanoï 2008.

[50] Nora Touati, *Amélioration des performances du schéma de la génération de colonnes : application aux problèmes de tournées de véhicules*. Université de Paris 13 , décembre 2008.

[51] Ouahiba Ramadane Cherif, *Problèmes d'Optimisation en Tournées sur Arcs*. Université de Technologie de toyes, décembre 2002.

[52] Paolo Toth and Daniele Vigo, *Models, relaxations and exact approaches for the capacitated vehicle routing problem*. Universit a di Bologna, Dipartimento di Elettronica, Informatica e Sistemistica, Viale Risorgimento 2. Discrete Applied Mathematics 123

(2002) 487 – 512, october 2000.

[53] Patrick Meyer, *Problème de Tournées de Véhicules avec Fenêtre de temps et application*. Universit de Liège, année 1999.

[54] Quan Lu et Maged Dessouky, *An Exact Algorithm for the Multiple Vehicle Pickup and Delivery Problem*. University of Southern California Los Angeles, Department of Industrial and Systems Engineering, décember 2002. [E-mail qlu, majed@scf.usc.edu].

[55] Pierre Schwartz, *Application d'un algorithme de colonie de fourmis au problème du voyageur de commerce*.juillet 2008. [Developpez.com].

[56] Raksmey PHAN, *Une stratégie hybride pour le problème de tournées de véhicules avec livraisons et collectes*. Laboratoire d'Informatique, de Modélisation et d'optimisation des systèmes, septembre 2009.

[57] Salah Eddine Merzouk, *Problème de Dimensionnement de Lots et de Livraison : Application au Cas d'une Chaine Logistique*. Université de Franche-Comté, decembre 2007.

[58] Susann SCHRENK, *Contributions à la conception de réseau de service en transport*. Université de Grenoble institut polytechnique de Grenoble, septembre 2010.

[59] Safa Khalouli, *Méta-Heuristiques à base de modèles : Application à l'ordonnancement d'atelier Flow-Shop hybride monocritère et mulicritère*. Université de reims Champagne-Ardenne, juillet 2010.

[60] Sylvain Marcel Robert Fournier, *Outils pour des Problèmes Industriels de Tournées de Véhicules avec Transbordement*. Université de Joséph Fourier, octobre 2008.

[61] Soumia Ichoua, *Problèmes de gestion de flotte de véhicules en temps réel*. Université de Monréal faculté des études supérieures, octobre 2001.

[62] Sylvain Girard*, Jacques Renaud and Fayez F.Boctor, *Heuristique Rapides pour le problème de tournées de véhicules*. Faculté des sciences de l'administration Université Laval Centre de recherche sur les technologies de l'organisation réseau, 2006.

[63] Tobias Buer et Giselher Pankratz, *Solving a Bi-Objective Winner Determination Problem Transportation Procurement Auction*. FernUniversitat in Hagen, februrar 2010.

[64] Thierry Garaix, *Étude et résoltution du problème de transport à la demande avec qualité de service*. Politecnico di Torino (italie), E-mail thierry.garaix@polito.it .

[65] Teodor Gabriel Crainic, Frédéric Semet, *Recherche opérationnelle et transport de marchandises*. Université de Montréal,Centre de recherche sur les transports, E-mail theo@crt.umontreal.ca, Université de Valenciennes, E-mail Frederic.Semet@univ-valenciennes.fr mai 2005.

[66] Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau et Christian Prins, *Heuristique pour les Problèmes de Tournées de Véhicules multi-attributs*. CIRRIELT-2011-12 www.cirrelt.ca, mars 2011.

[67] Yannich Kergosien, *Algorithmes de tournées de véhicules pour l'optimisation des flux de produits et de patients dans un complexe hospitalier*. Université de François Rablais de tours, juillet 2010.

[68] Jozefoweiz Nicolas, *Modélisation et résolution approchée de problèmes de tournées multi-objectif*. Université des sciences et technologies de lille, décembre 2004.

NB : L'ensemble de la documentation a été téléchargée sur www.google.com sauf les ouvrages indiqués par (*).

Résumé

Résumé

Le VRPTW (Vehicle Routing Problem with Time Windows) est constitué d'une flotte de véhicules qui doit servir un ensemble de clients dispersés géographiquement, chacun ayant une demande connue et un intervalle de temps, durant lequel il peut être servi, fixé. Dans ce mémoire, nous présentons un état de l'art du VRP et du VRPTW ainsi qu'une gamme d'approches utilisées pour leur résolution. Nous construisons ensuite une approche hybride (Génétique-Taboue) permettant la résolution d'une variante du VRPTW que nous avons implémentée en langage C sous Linux. Les résultats obtenus s'avèrent satisfaisants pour les problèmes dont les paramètres sont générés aléatoirement et moins satisfaisants pour ceux dont les clients sont répartis en groupes. Pour améliorer la qualité des solutions nous suggérons de revoir la phase d'initialisation et d'adapter le croisement choisi aux problèmes dont les clients sont répartis en groupe.

Abstract

The VRPTW (Vehicle Routing Problem with Time Windows) is constituted by a fleet of vehicles which must serve a set of geographically dispersed customer. The demand of each customer and the time window during which each it can be served are fixed. In this memory, we present a state of the art of VRP and VRPTW thus a range of approaches to their resolution. Then, we construct a hybrid approach (Genetic-Tabu) to solve a variant of the VRPTW which we have implemented in C version Linux. The results are satisfactory for problems whose geographical data are randomly generated and less satisfactory for problems those geographical data are clustered. To improve the quality of the solutions we suggest reviewing the initialization phase and adjust the selected crossover to problems whose geographical data are clustered.