République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahleb Blida
Département d'Informatique

Projet fin d'études pour l'obtention du diplôme de master Professionnelle en système informatique et Réseaux.

« SIR »

Théme :

Algorithme inspiré des réactions chimiques amélioré pour résoudre le problème de tournées de véhicules multi-dépôts

***Présenté par*:**

- **Kartous chiraz**
- **Belhadj yamina**

Présentée et soutenue le 19 Juin 2023 devant le jury composé :

Dr. Hireche Célia, Présidente. Université de Blida 1

Dr. Boutoumi Bachira, Examinatrice. Université de Blida 1

Dr. Ferdi Imene, Encadreur. Université Blida 1

Promotion 2022/2023

Remerciements

Tous d'abord, nous tenons en premier lieu à remercier Dieu tout puissant de nous avoir donné la force, la santé et la volonté pour réaliser ce travail.

Nous tenons à remercier vivement notre encadreur ***Dr*** *FerdiImene pour son encadrement et ses précieux conseils tout au long de ce travail. Sa disponibilité, son expertise et sa bienveillance ont grandement facilité notre progression, et nous lui sommes très reconnaissants.*

Nous exprimons notre grand respect également aux membres du jury d'avoir accepté d'examiner et dévaluer ce travail.

Enfin Nous tenons à exprimer notre reconnaissance envers nos proches, nos familles et nos amis, pour leur soutien indéfectible, leurs encouragements et leur compréhension durant cette période intense. Leur présence à nos côtés a été d'un grand réconfort et nous leur en sommes infiniment reconnaissants.

Merci infiniment.

Dédicace

C'est avec profonde gratitude et sincères mots, que nous dédions ce modeste travail de fin d'étude à nos chers parents, qui ont sacrifié leur vie pour notre réussite et nous ont éclairé le chemin par leurs conseils judicieux.

Nous espérons qu'un jour, nous pourrons leurs rendre un peu de ce qu'ils ont fait pour nous, que dieu leur prête bonheur et longue vie. Nous dédiens aussi ce travail à nos frères et sœurs, nos familles, nos amis,

tous nos professeurs qui nous ont enseigné et à tous ceux qui nous sont chers.

Chiraz & Yamina

Résumé

De nos jours, les ingénieurs sont confrontés à une variété de problèmes de complexité grandissante. Ces derniers apparaissent dans différents secteurs tels que les problèmes detransport.Trouver la solution optimale dans un délai raisonnable est devenu une priorité essentielle pour toute entreprise.

Notre travail porte sur le problème de Tournées de Véhicules Multi-Dépôts (Multi-Dépôts Véhicule Routing Problem MDVRP) qui est une extension de VRP classique,Il appartient à la classe des problèmes dire NP-difficiles. L'objectif est de servir un ensemble de clients à partir d'ensemble de dépôts et à travers d'un ensemble de véhicules tout en minimisant un coût total de la livraison.

La motivation de ce travail est d'aborder un problème difficile et dans une version encore peu étudiée dans la littérature, comportant des capacités limitées à la fois pour les dépôts et les véhicules. De plus, des problèmes de taille réaliste sont visés.

Le but de ce mémoire est de développer une méthode efficace et simple pour la résolution de MDVRP. Pour atteindre notre objectif, nous proposons d'améliorer et d'adapter l'algorithme inspiré des réactions chimiques (ChemicalReactionOptimization CRO).

Pour tester les performances de notre méthode proposée, nous l'avons testé et validé sur un ensemble de tests provenantdela littérature. Les études expérimentales faites ont montré l'efficacité de l'approche proposée de trouver des solutions de bonne qualité pour le problème étudiée dans ce travail.

Mots-clés : l'optimisation combinatoire, Problème de tournées de véhicules multi-dépôts(MDVRP), Algorithme inspiré des réactions chimiques (CRO), heuristique « Route first-cluster second ».

Abstract

In the real world, engineers face a variety of problems of increasing complexity. The latter appear in different sectors such as the transport problems. Finding the optimal solution in a reasonable time has become a top priority for any business.

The Multi-Depot Vehicle Routing Problem (MDVRP) which is an extension of classic VRP and belongs to the class NP-difficult problems.The objective is to serve a set of customers from a set of depots by using a set of vehicles; the goal is to satisfy a set of customers by minimizing an overall cost.

The motivation of this work is to approach to a difficult problem and in a version still little studied in the literature, which limited capacities at the same time for the depots and the vehicles are considered. In addition, realistic size problems are targeted.

The purpose of this thesis is to develop an efficient and simple method for solving MDVRP. To achieve our goal, we propose to improve and adapt the algorithm inspired by chemical reactions (Chemical Reaction OptimizationCRO).

To test the performance of our proposed method, we test and validate it against a set of tests in the literature. Experimental studies demonstrate the effectiveness of the proposed method in finding high-quality solutions.

To test the performance of our proposed method, we tested and validated it on a set of tests from the literature. The experimental studies made have shown the effectiveness of the proposed approach to find good quality solutions for the problem studied in this work.

Keywords: combinatorial optimization, Multi-depot vehicle routing problem (MDVRP), Chemical reaction inspired algorithm (CRO), “Route first-cluster second” heuristic.

ملخص

يواجه المهندسون مجموعة متنوعة من المشاكل ذات التعقيد المتزايد تظهر هذه اﻷخيرة في قطاعات مختلفة مثل النقل أصبح إيجاد الحل اﻷمثل في وقت معقول أولوية قصوى ألي عمل تجاري.

يرتكز عملنا على مشكلة توجيه المركبات متعددة اإلستعماالت التي تعد إمتدادالVRP الكالسيكي وهي تنتمي إلى فئة مايسمى بمشاكل NP الصعبة الهدف حيث هو خدمة مجموعة من العمالء من المستودعات وعبر مجموعة من المركبات مع تقليل تكلفة اإلجمالية للتسليم.

الدافع وراء هذا العمل هو معالجة مشكلة صعبة وفي نسخة التزال تدرس في اﻷدبيات مع قدرات محدودة لكل من المستودعات والمركبات. باإلضافة لذلك يتم استهداف مشاكل الحجم الواقعي.

الغرض من هذه اﻷطروحة هو تطوير طريقة فعالة وبسيطة لحل MDVRP لتحقيق هدفنا نقترح تحسين وتكييف الخوارزمية المستوحاة من التفاعالت الكيميائية CRO .

الختبار أداء طريقتنا المقترحة، قمنا باختبار والتحقق من صحتها مقابل مجموعة من االختبارات في اﻷدبيات. توضح الدراسات التجريبية فعالية الطريقة المقترحة إليجاد حلول عالية الجودة للمشكلة المدروسة في هذا العمل.

الكلمات المفتاحية: طرق التحسين التجميعية، مشكلة توجيه المركبات، مشكلة توجيه المركبات متعددة المستودعات، الخوارزمية المستوحاة من ردود التفاعالتالكيميائية،اإلستدالل " الطريق اﻷول-المجموعة الثانية"

Table des matières

Liste des figures

Liste des Tableaux

Liste des algorithmes

Liste des abréviations

RS : Recuit Simulé (RS).

PSO : Optimisation par essaims de particules**.**

AG :algorithme Génétique.

CRO : L'algorithme inspiré des réactions chimiques.

VRP : Le problème de tournées de véhicule.

MDVRP : Problème de tournées de véhicules avec dépôts multiples.

CVRP : Problème de tournée de véhicules avec contrainte de capacité.

VRPTW : Problème de tournée de véhicules avec fenêtre de temps.

SDVRP : Problème de tournées de véhicules Split-Delivery.

PLNE : Programmation Linéaire en nombres entiers.

PL : Programmation Linéaire.

TS : TabuSearch.

KE : Energie Cinétique.

PSO: Particle Swarm Optimization.

GTS: GranularTabu Search.

ACO : Ant Colonie Optimization.

GVNS : voisinage variable.

DAL : OappeléeDisceteAntlionOptimization.

TSP : le probléme du voyageur de commerce.

SA : recuit simulé.

PE : Energie Potentielle.

MHDT: Merge-Head and Drop-Tail.

BKS : Best Known Solution.

Introduction générale

Actuellement, les chercheurs et les ingénieurs sont souvent confrontés à des problèmes socioéconomiques de plus en plus difficiles. Ces problèmes surgissent dans différents domaines, comme le transport et la distribution de marchandises, la production…etc. La résolution des problèmes de logistique de transport sont devenues des éléments essentiels pour assurer le bon fonctionnement des entreprises. Avec la mondialisation croissante, la complexité des chaînes d'approvisionnement et l'augmentation des attentes des clients, la gestion logistique est devenue un facteur clé de compétitivité.Une gestion efficace des coûts de transport peut contribuer à améliorer la rentabilité, à maintenir la compétitivité et à maximiser les bénéfices.

Un problème donné peut être défini par l'ensemble des propriétés que doivent vérifier ses solutions. Il peut s'agir d'un problème de décision ou d'un problème d'optimisation. Un problème de décision peut être réduit à un problème d'existence de solution.Par contre, un problème d'optimisation peut se ramener à un problème d'existence de solution de bonne qualité, Il s'agit de parcourir l'espace de recherche pour extraire la solution optimale parmi un ensemble fini de solutions. La résolution des problèmes d'optimisation nécessite l'utilisation des algorithmiques qui permettent de maximiser ou de minimiser une ou plusieurs fonctions "objectifs", tout en respectant les contraintes imposées par le problème.

Dans ce mémoire, nous nous sommes intéressés d'étudier le Problème de Tournées de Véhicules Multi-Dépôts PTVMD (MDVRP : Multi-Depot Vehicule Routing Problem) qui est une extension du problème classique de tournées de véhicules (VRP). MDVRP consiste à élaborer des tournées de véhicules afin desatisfaire les demandes des clients à partir d'un ensemble de dépôts, pour chacun d'entre eux une flotte de véhicules est affectée. L'objectif est de minimiser le coût total de la livraison.

Le problème étudié dans ce travail appartient aux problèmes dits NP-difficiles. Pour cela, Nous avons basé sur la méta-heuristique plus précisément celles basées population pour résoudre ce problème. La motivation derrière ce choix est que les méta-heuristiques basées population sont généralement plus performantes que les méta-heuristiques basées solution unique.

L'objectif principal à travers ce projet consiste à améliorer et adapter l'algorithme inspiré des réactions chimiques CRO (Chemical Reaction Optimization) pour la résolution du problème MDVRP.

CRO est un algorithme basé population récemment développé pour résoudre desproblèmes d'optimisation dans des domaines discrets et continus. Ce dernier s'inspire des phénomènes de réaction chimique basés sur la répartition de l'énergie entre les molécules.

Des études expérimentales et statistiques ont été faites pour valider l'approche proposée.

Enfin, parmi nos perspective nous envisageons d'amélioré les résultats obtenus en utilisant plusieurs méthode de la recherche local et d'optimiser les opérateurs de CRO.

Structuration du document

Le mémoire présenté est divisé en deux parties chacune contient deux chapitres. La première partie présente les généralités de base et l'état de l'art et la seconde partie contient notre contribution. Les chapitres sont organisés de la manière suivante :

- ✓ Le premier chapitre est consacré à la présentation des notions de base de l'optimisation combinatoire, la théorie de la complexité algorithmique et les méthodes de résolution des problèmes d'optimisation.
- ✓ le deuxième chapitre s'intéresse au problème du tournées de véhicules multi-dépôts, nous commençons par la présentation du problème de tournées de véhicules, et identifier ses variantes, Ensuite, nous présentons le problème de tournées de véhicules multi-dépôts en donnant sa définition mathématique et les différentes méthodes de résolution développées dans la littérature pour le résoudre.
- ✓ Le troisième chapitre, présente en détail notre contribution, il couvre le principe de fonctionnement de l'algorithme inspiré des réactions chimiques amélioré. Nous détaillons la méthode utilisée pour créer la population initiale et les différents opérateurs de CRO proposés pour MDVRP.
- ✓ Le quatrième chapitre est consacré à l'étude expérimentale. Nous commençons par décrire les outils utilisés pour le développement du projet. Ensuite, nous donnons les différents paramètres de CRO fixés pour tester l'algorithme. En fin, les résultats obtenus par notre approche proposée sont également communiqués et discutés.

Finalement, nous terminons avec une conclusion générale sur la contribution proposée et quelque respectives pour nos futures travaux de recherche.

Partie 1

Concepts de base et état de l'art

CHAPITRE 1:

Introduction à l'optimisation combinatoire

1.1 Introduction

De nombreux secteurs de l'industrie (mécanique, chimie, télécommunication transport, etc.) sont concernés par des problèmes complexes de grand dimension et multi-critères (coûts financiers, qualités de services), la recherche d'une solution optimale ou sous optimale dans un temps raisonnable est devenue une priorité importante dans toute entreprise.

L'optimisation combinatoire joue un rôle très importante en recherche opérationnelle, en Mathématiques discrètes et en informatique.Son importance se justifie d'une part par la grande Difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre.Dans ce chapitre, Nous nous focalisons sur les notions de base de l'optimisation, les principales classes de complexité. De plus, la définition des problèmes d'optimisation combinatoire et les différentes méthodes de résolution les plus connues pour les résoudre.

1.2 Notions de base

Selon [Paschos, 2005] « L'optimisation c'est l'art de comprendre un problème réel, de pouvoir le transformer en un modèle mathématique que l'on peut étudier afin d'extraire les propriétés structurelles et de caractériser les solutions du problème. Enfin, l'optimisation c'est l'art d'exploiter cette caractérisation afin de déterminer les algorithmes qui les calculent mais aussi, de mettre en évidence les limites sur l'efficience et l'efficacité de ces algorithmes ».

Un problème d'optimisation se définit comme la recherche du minimum ou du maximum d'une fonction donnée.

Dans ce qui suit, nous proposons quelques définitions tirées de la littérature relative aux problèmes généraux d'optimisation.

- **Définition 1 :** d'après [Papadimitriou et Steiglitz, 1982] : Une instance d'un problème de minimisation (maximisation) est un couple $(X,\ f)$. Où $X \subseteq S$ est un ensemble fini de solutions potentielles admissibles et f est une fonction du coût (fonction objectif) à minimiser (à maximiser), $f : \Re \rightarrow X$. L'objectif est de trouver $s* \in X$ tel que $f(s*) \leq f(s)$ (Au cas de maximisation : $f(s*) \geq f(s)$ pour n'importe quelle solution $s \in X$.

- **Définition 2 :**d'après [Papadimitriou et Steiglitz, 1982] : l'optimisation mono-objectif**,** se base sur la minimisation (ou la maximisation) d'une seule fonction objective où le but est de trouver la meilleure solution appelée solution optimale qui est définie suivant une seule performance du problème étudié.

De manière formelle, à chaque instance d'un tel problème est associé un ensemble W des solutions potentielles respectant certaines contraintes et une fonction d'objectif $W \epsilon Y$ qui associe à chaque solution admissible $s\ de\ W$ une valeur (s). Résoudre l'instance (W) du problème d'optimisation consiste à trouver la solution optimale $s\ *$ de W qui optimise (minimise ou maximise) la valeur de la fonction objective pour le cas de la minimisation : le but est de trouver $s\ *$ de W tel que $(s\ *)\ \epsilon\ (s)$ pour tout élément s de W. Un problème de maximisation peut être défini de manière similaire.

- **Définition 3 :**L'optimum local d'un problème d'optimisation est la solution optimale (maximum ou minimum) dans un ensemble voisin de solutions candidates. on dit que la solution s' (appartenant àS) est un optimum local du voisinage $V(s)$ de la solution s si vérifie la condition suivante :
  - ✓ $f\ (s') \leq f\ (s)\ \forall s \in V(s)$Pour un problème de minimisation.
  - ✓ $f\ (s')\ \geq f\ (s)\ \forall s \in V(s)$ Pour un problème de maximisation.

- **Définition 4:**Une solution optimale (optimum global) : une solution **s*** est considérée comme optimale, s'il n'y a pas d'autre meilleure solution. Autrement dit, l'optimum global est le meilleur optimum local. On dit donc qu'une solution est globalement optimale si :
  - ✓ $f\ (s\ *)\ \geq\ f\ (s)$ dans le cas de problème de maximisation.
  - ✓ $f\ (s\ *)\ \leq\ f\ (s)$ dans le cas de problème de minimisation.

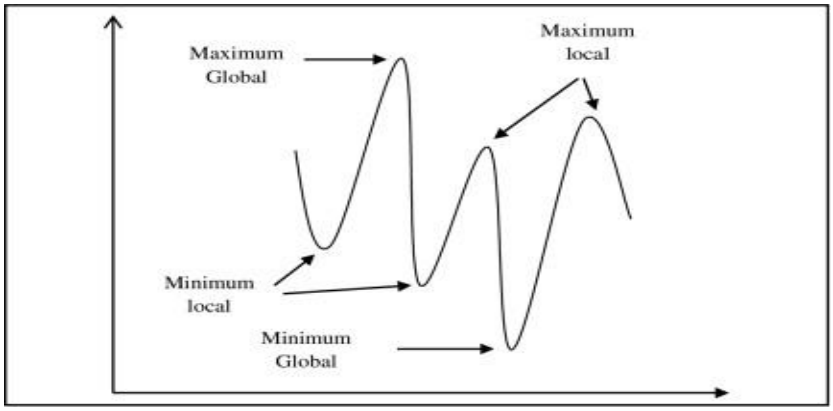

Figure 1.1 : Courbe représentant les optimums locaux et les optimums globaux[Devarenne, 2007].

1.3 Problème d'optimisation combinatoire

L'optimisation combinatoire joue un rôle très important en recherche opérationnelle, en mathématiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation [Papadimitriou et Steiglitz, 1982]et d'autre part par de

nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire[Ribeiro et Maculan, 1994].

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

1.3.1 Quelques problèmes classiques d'optimisation combinatoire

Parmi les problèmes d'optimisation combinatoire les plus connus, On peut citer

- **Problème du sac-à-dos :** Le problème du Sac à Dos aussi noté KSP (KnapsackProblem) étant donné plusieurs objets possédants chacun un poids et une valeur et étant donné un poids maximum pour le sac, quels objets faut-il mettre dans le sac de manière à maximiser la valeur totale sans dépasser le poids maximal autorisé pour le sac[Mounir et Ouldahmed, 2009].
- **Problème d'ordonnancement :**Le problème d'ordonnancement consiste à organiser dans le temps la réalisation d'un ensemble de tâches, compte tenu de contraintes temporelles (délais, contraintes d'enchaînements, ...) et de contraintes portant sur l'utilisation et la disponibilité des ressources requises [Bourazza,2006].

- **Problème du voyageur de commerce :** est un problème classique d'optimisation combinatoire(en anglais Travelling SalesmanProblem : TSP)**.**est un cas particulier du problème de Tournées de Véhicules sans contrainte de capacité et avec un seul véhicule [Rego et al ,1994].on suppose un voyageur de commerce qui doit visiter n villes données, en passant par chaque ville exactement une et une seule fois. Il commence par une ville quelconque et doit terminer en retournant à la ville du départ. Les distances entre les villes sont connues. Le problème consiste à déterminer le chemin permettant de minimiser la distance parcourue.

1.4 Les méthodes de résolution des problèmes d'optimisation

Il existe deux grandes catégories des méthodes de résolution des problèmes d'optimisation combinatoires : les méthodes exactes et les méthodes approchées. Les méthodes exactes se basent sur l'énumération de l'ensemble des solutions potentielles et permettent d'obtenir la solution optimale. Mais la taille du problème influe rationnellement sur l'efficience (temps d'exécution) de cette classe, par contre on a les méthodes approchées, qui permettent d'obtenir une bonne solution, qui n'est pas toujours la solution optimale, mais avec un temps raisonnable.

Le schéma ci-dessous représente cette classification :

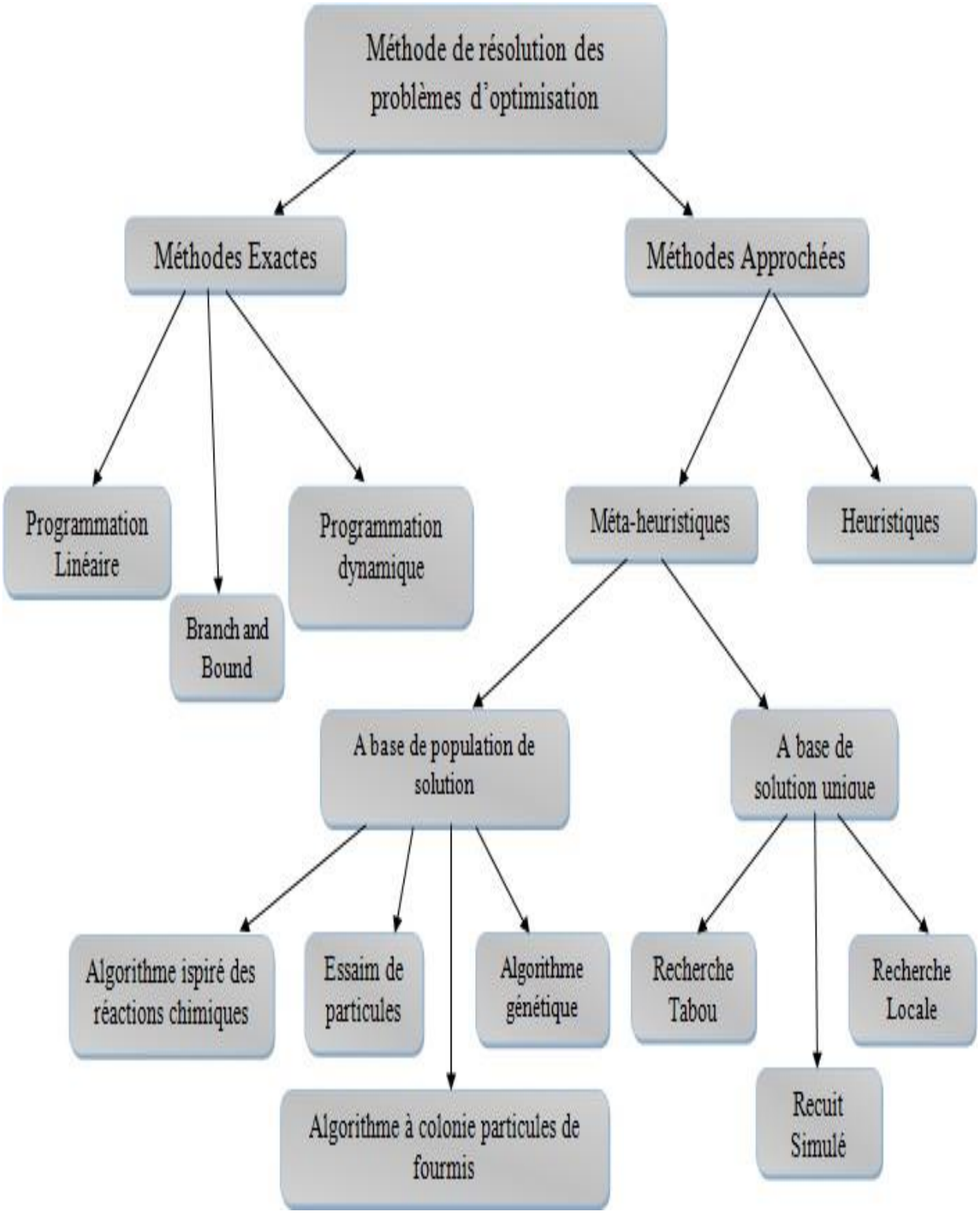

Figure 1.2 : Classification de méthodes de résolution de problèmes d'optimisation [Talbi, 2009]

1.4.1 Les méthodes exactes

Les méthodes exactes génèrent une solution optimale pour une instance donnée d'un problème d'optimisation. Le principal inconvénient de son approche est l'explosion combinatoire : le nombre de combinaisons augmente avec la dimensionnalité du problème. L'efficacité de ces algorithmes n'est prometteuse que pour les petites instances de problèmes.On peut citer parmi les méthodes exactes : la programmation dynamique, la programmation linéaire, l'algorithme Branch and Bound…

1.4.2 Les méthodes approchées

Les méthodes approchées sont souvent utilisées pour résoudre des problèmes plus complexes ou de taille plus grande que ceux traités par les méthodes exactes. Les méthodes approchées ne garantissent pas de solutions optimales, mais plutôt souvent proches de l'optimum. Ces méthodes peuvent être divisées en deux sous familles : les heuristiques et les méta-heuristiques.

1.4.2.1 Les heuristiques

Une heuristique est plutôt une méthode qui cherche (une stratégie) sans garantir le résultat. Destiné à un problème spécifique, le temps de calcul est raisonnable sans garantir la faisabilité ou l'optimalité.Selon [Feigenbaum et Feldman, 1963]:« Une heuristique est une méthode qui aide à découvrir la solution d'un problème en faisant des conjectures plausibles mais faillible de ce qui est la meilleure chose à faire. »

1.4.2.2 Les métaheuristiques

Une métaheuristique est un algorithme d'optimisation visant à résoudre des problèmes d'optimisation difficile (souvent issue des domaines de la recherche opérationnelle, de l'ingénierie ou de l'intelligence artificielle) pour lesquels on ne connaît pas de méthode classique plus efficace.

Les métaheuristiques sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum global. Elles se comportent comme des algorithmes de recherche, tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution (d'une manière proche des algorithmes d'approximation).

Il existe un grand nombre de métaheuristiques différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale. Ces méthodes utilisent cependantun haut niveau d'abstraction, leur permettant d'être adaptées à une large gamme de problèmes différents.

En 1996, I.H. Osman et G. Laporte définissaient la métaheuristique comme « un processus itératif qui subordonne et qui guide une heuristique, en combinant intelligemment plusieurs concepts pour explorer et exploiter tout l'espace de recherche. Des stratégies d'apprentissage sont utilisées pour structurer l'information afin de trouver efficacement des solutions optimales, ou presque-optimales »**.**

Cette branche est composée de deux grandes classes :

1.4.2.2.1 Méta-heuristiques à solution unique

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

A) Le recuit simulé

La méthode du Recuit Simulé (RS) est une technique de recherche locale inspirée du processus utilisé en métallurgie. Ce processus alterne des cycles de refroidissement lents et de réchauffage ou de recuit qui tendent à minimiser l'énergie du matériel. Le recuit simulé s'appuie sur l'algorithme de Metropolis-Hastings, qui permet de décrire l'évolution d'un système thermodynamique. Par analogie avec le processus physique, la fonction à minimiser deviendra l'énergie E du système. On introduit également un paramètre fictif, la température T du système. L'algorithme du recuit simulé part d'une solution donnée, et la modifie itérativement jusqu'au refroidissement du Système. Les solutions trouvées peuvent améliorer le critère que l'on cherche à optimiser, on dit alors qu'on a fait baisser l'énergie du système, comme elles peuvent le dégrader. Si on accepte une solution qui améliore le critère, on tend ainsi à chercher l'optimum dans le voisinage de la solution de départ. Contrairement autres méthodes de recherche locale, le recuit simulé peut accepter des solutions dont la qualité est moins bonne en fonction de la dégradation de la solution considérée [Aarts et Korst, 1989].

$x :=$ *solution aléatoire*
$f(x) =$ *valeur de la fonction*
F_{min} := f(x)

X_{min}:= x
$T :=$ *initialisertempérature* (*assez élevée*)
REPETER
générer un voisin$s(x) \in$ voisinage(x), appliquer la règle de Metropolis
$SIf(x) < F_{min}$

F_{min} := f(x)
$X_{min} :=$ x

FIN DE SI
JUSQU'A équilibre thermodynamique atteint
$T :=$ décroître température
JUSQU'A conditions d'arrêt satisfaites

Algorithme1.1 : Pseudocode de la méthode du recuit simulé [web2].

B) La méthodeTabou:

La recherche Tabou est une méthode de recherche dont les principes ont été proposés pour la première fois par Fred Glover [Glover, 1986] et elle est devenue très classique en optimisation combinatoire. Elle se distingue des méthodes de recherche locale simples par le recours à un historique des solutions visitées, de façon à rendre la recherche un peu moins« aveugle ».pour éviter de retomber périodiquement dans un minimum local, certaines solutions sont bannies, elles sont rendues taboues.

A l'inverse du recuit simulé qui génère de manière aléatoire une seule solution voisine $N(s) \in s'$ à chaque itération, Tabou examine un échantillonnage de solutions de $N(s)$ et retient la meilleure s' même si $f(s') > f(s)$. La recherche Tabou ne s'arrête donc pas au premier optimum trouvé.

Le danger serait alors de revenir à s immédiatement, puisque s est meilleure que s'. Pour éviter de tourner ainsi, on crée une liste T qui mémorise les dernières solutions visitées et qui interdit tout déplacement vers une solution de cette liste. Cette liste T est appelée liste Tabou [Baptiste, 2006].

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

Algorithme 1.2 : L'algorithme de la recherche tabou. [Hafez, 1999]

1.4.2.2.2 Les Méta-heuristiques à base de population de solutions

Les méta-heuristiques à base de population de solutions débutent la recherche avec une panoplie de solutions. Elles s'appliquent sur un ensemble de solutions afin d'en extraire la meilleure (l'optimum global) qui représentera la solution du problème traité. L'idée d'utiliser un ensemble de solutions au lieu d'une seule solution renforce la diversité de la recherche et augmente la possibilité d'émergence de solutions de bonne qualité. Une grande variété de méthodes basées sur une population de solutions a été proposée dans la littérature. [Gherbouj, 2013].

A) Optimisation par essaims de particules

L'optimisation des essaims de particules (PSO) a été inventée par Russel Eberhart et James Kennedy [Kennedy et Russell, 1995],pour résoudre des problèmes d'optimisation complexes.L'optimisation des essaims de particules (PSO) s'inspire de la nature. Les troupeaux d'oiseaux et de bancs de poissons ajustent leurs mouvements physiques pour éviter les prédateurs, chercher de la nourriture ou des partenaires, ou optimiser les paramètres environnementaux tels que la température.

L'objectif de PSO est de trouver un optimum global sur un espace de recherche. Un essaim dans l'algorithme est représenté par un ensemble de particules [x1, x2, x3….] l'objectif est de minimiser ou de maximiser la fonction objectif f(x).

L'avantage majeur de PSO est la facilité de l'implémentation, un autre point fort de cette méthode est que chaque particule est soumise à des règles de déplacement très simple, qui mènent cependant l'essaim à converger rapidement vers un optimum[web4].

B) L'algorithme Génétique

L'algorithme génétique (AG) est une méthode d'optimisation dans laquelle la population de solutions potentielles, appelées individus, est progressivement mise à jour grâce à des mécanismes de sélection et à des processus génétiques : croisement et mutation. La sélection naturelle de l'évolution, favorise les individus les mieux adaptés dans une population. La sélection est suivie d'une hybridation et d'une mutation au niveau individuel, consistant en un ensemble de gènes. Ainsi, deux individus "parents" hybrides transmettent une partie de leur patrimoine génétique à leur descendance. Chaque enfant est plus ou moins adapté à l'environnement. S'il s'adapte bien, il a de meilleures chances de se reproduire à la prochaine génération. Au fil des générations, les individus les plus en forme sont sélectionnés et une augmentation du nombre d'individus en bonne forme entraîne l'évolution de l'ensemble de la population [Benkadour et Aribi, 2013].

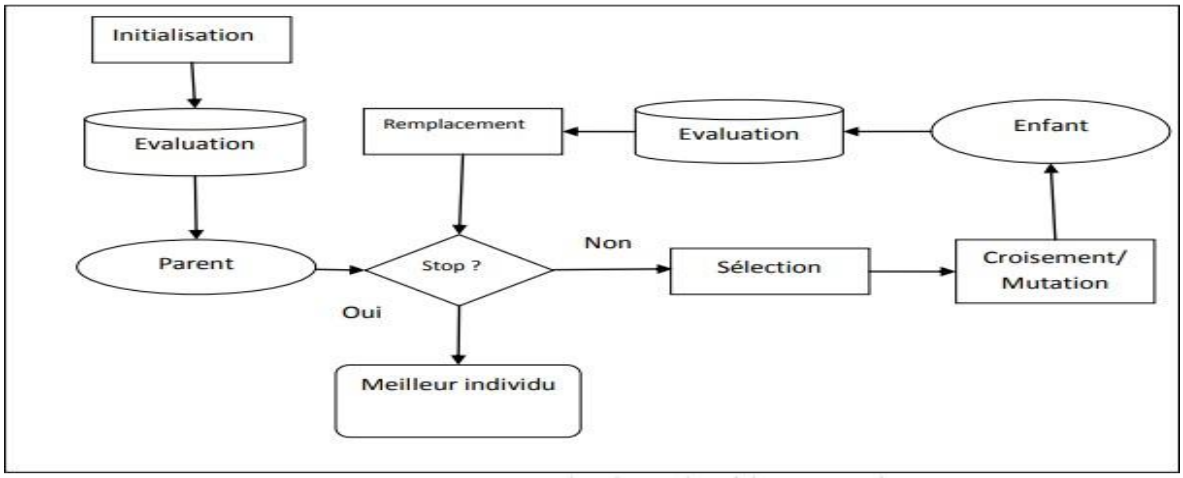

Figure1.3 : Démarche d'un algorithme génétique. [Ghali, 2005]

1.5 L'algorithme inspiré des réactions chimiques CRO

ChemicalReactionOptimization (CRO) est un algorithme méta-heuristique basé population, récemment développé pour résoudre des problèmes d'optimisation dans des domaines discrets et continus. Ce dernier s'inspire des phénomènes de réaction chimique basés sur la répartition de l'énergie entre les molécules. Le CRO est utilisé pour résoudre de nombreux problèmes optimisation. On peut citer : le travail de [Truong et al, 2013] qui applique la CRO pour résoudre le problème du sac à dos binaire, [Sun et al., 2011] ont développé une approche hybride basée sur le CRO et l'algorithme de recherche locale pour résoudre le problème du voyageur de commerce à objectif unique, Par ailleurs, [Xu et al., 2013] ont développé un CRO pourla planification du graphe acyclique orienté sur les systèmes informatique hétérogènes.

A traves les différents travaux dans l'axe de CRO, on constante que cet algorithme a montré une grande efficacité dans nombreux problèmes d'optimisations. Nous allons détailler les principes de cet algorithme et sa source d'inspiration par la suite.

1.5.1 Inspiration

Les principes des réactions chimiques sont régis par deux lois de la thermodynamique. La première loi de conservation de l'énergie stipule que l'énergie n'est ni créée ni détruite. L'énergie est convertie d'une forme à une autre et transférée d'un matériau à un autre. Les réactions chimiques peuvent être endothermiques ou exothermiques. Les réactions endothermiques nécessitent la chaleur de l'environnement pour initier le processus de réaction. D'autre part, les réactions qui libèrent de la chaleur dans l'environnement sont appelées réactions exothermiques. La deuxième loi stipule que l'entropie d'un système tend à augmenter, et l'entropie est une mesure du degré de désordre. L'énergie potentielle est l'énergie stockée dans une molécule. Lorsque cette énergie est transformée en d'autres formes, le système devient plus désordonné.

[Lam et Li, 2010] se sont inspirées de ce processus thermodynamique pour crée un algorithme bio-inspirés qui se base sur la collision entre les molécules. Chaque molécule M possède plusieurs attributs essentiels notamment sa structure moléculaire(ω) son energie potentielle (PE) et son énergie cinétique (KE). De plus, il existe de nombreuses variantes différentes de CRO en fonction des opérateurs de l'algorithme et de la manière dont ils sont utilisés pour le construire. La plupart des variantes CRO publiées incluent le nombre de collision que la molécule subite NumHit. La meilleure valeur que M a eu durant son existence $MinPE$ et le nombre de collision associé à $MinPE$ qui est $MinHitPE$ qui représente la valeur de la fonction objectif de ω tandis que KE quantifie la tolérance du système d'accepter une mauvaise solution[Sun et al, 2011].

1.5.2 Schéma général de CRO

CRO est un algorithme similaire à d'autres algorithmes évolutifs et se compose de trois étapes principales : initialisation, itération et étape finale. Lors de l'initialisation, le CRO définit différents éléments (paramètres du numérateur et de l'algorithme) nécessaires à son fonctionnement. L'algorithme itère ensuite dans l'espace des solutions. À la dernière étape,

l'algorithme se termine et produit la solution optimale trouvée. Le schéma général de CRO est illustré dans la figure 1.5.

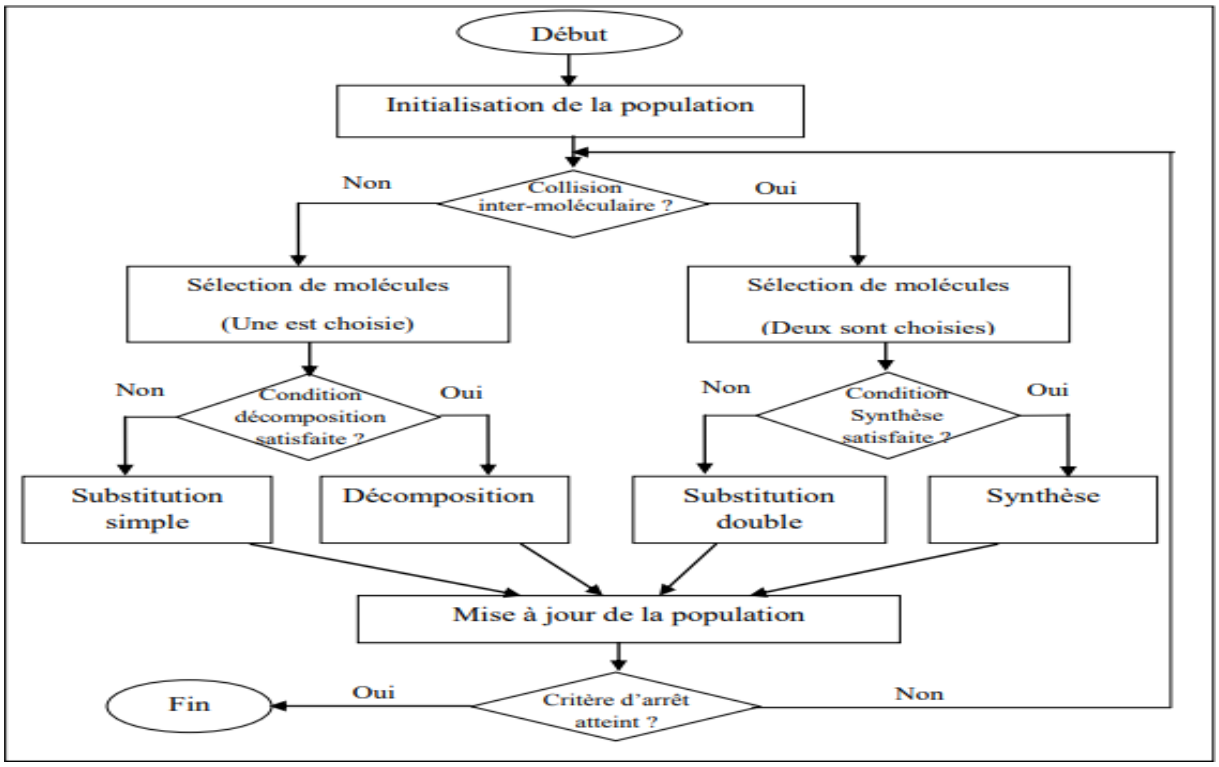

Figure 1.4 : Schéma général de CRO. [Lam et Li, 2010]

1.5.2.1 L'initialisation

CRO commence à générer une population de molécules PopSize. Chaque molécule est composée des champs susmentionnés. De plus, l'algorithme initialise les paramètres aléatoirment de l'algorithme qui sont $KELossRate, MoleColl, Buffer, InitialKE, \alpha et \beta$. 000

1.5.2.2 Les itérations

Les molécules peuvent entrer en collision avec les parois du conteneur ou entrer en collision avec d'autres molécules. Ceci est déterminé en générant un nombre aléatoire r $\in [0,1]$. Si $r > MoleColl,$ il n'y a qu'une seule molécule dans le système et des collisions unimoléculaires se produisent. Sinon, des collisions intermoléculaires se produiront.

Pour que la collision unimoléculaire se produise, la molécule M doit pouvoir subir une décomposition ou une substitution simple. Si la condition de décomposition est satisfaite $(NumHit - MinHit > \alpha)$, alors Msera remplacé par les deux nouvelles molécules M_0et M_1si la condition de conservation de l'énergie de décomposition est satisfaite (formule 1.3). Sinon, nous obtenus une substitution simple et le changement de la molécule M sera accepté sauf si la condition (1.2) est vérifiée. De même, pour une collision intermoléculaire, on sélectionne au hasard deux molécules$M1$ et $M2$ ces molécules subiront une synthèse ou une substitution double. Dans le cas où la condition de la synthèse (KE≤ β) est vérifiée alors une synthèse sera

appliqué et les molécules M_1et M_2 seront remplacées automatiquement par la nouvelle molécule M' si la condition de conservation de l'énergie (1.8) est satisfaite. Sinon, la substitution double aura lieu, le changement engendré par l'opérateur sur les deux molécules seront acceptés si la condition de conservation d'énergie (1.6) est satisfaite. Après la fin d'une réaction élémentaire. L'algorithme vérifié si la condition de conservation de l'énergie est respectée. Dans le cas contraire le changement est annulé. Ensuite, l'algorithme vérifie si la nouvelle solution a une valeur de fonction objectif meilleure que l'ancienne. Si c'est le cas, alors elle sera enregistrée, si aucun critère d'arrêt n'est atteint une nouvelle itération sera lancée.

A) La substitution simple modélise la situation quand une molécule percute la paroi du conteneur et reste en une seule unité. Dans cette collision seulement la molécule existante ω était perturbée en ω' [Lam etLi, 2012]. Soit ω' une solution de l'entourage deω, et $PE(\omega') = f(\omega')$.L'énergie cinétique KE ω' est calculée selon la formule (1.1) :

$$KE\omega' = (PE(\omega') - PE(\omega') + KE\omega') \times \alpha. \qquad (1.1)$$

De plus, l'énergie restante $(PE(\omega') - PE(\omega') + KE(\omega') \times (1 - \alpha))$, est transférée au buffer. Soit KELossRate un paramètre de CRO où : 0≤ KELossRate ≤1 et α est un nombre aléatoire dans $(KELossRate, 1)$. La molécule transformée sera accepté si la condition de conservation d'énergie de la substitution simple (Dans le cas de minimisation) est satisfaite comme exprimé dans la formule (1.2) :

$$(\omega) + KE\omega' \geq PE(\omega') \qquad (1.2)$$

Cette condition accepte toujours les bonnes solutions $(\omega') > PE(\omega)$, mais dans le cas contraire $[(\omega) < PE(\omega')]$ elle peut accepter une mauvaise solution. Lorsqu'une molécule subit plus de réaction élémentaire elle aura plus de KE transféré dans le buffer. Par conséquence, la probabilité d'avoir une solution pire est plus faible lors d'un changement ultérieur.

B) La décomposition l'opérateur de la décomposition se réfère à la situation où une molécule frappe la paroi de conteneur et se décompose en plusieurs parties (généralement deux parties). Supposons que la molécule $M\omega$ produit $M\omega 1'$ et $M\omega 2'$. L'idée de la décomposition est de permettre au système d'explorer d'autres régions de l'espace de recherche après une recherche locale suffisante. Théoriquement, il est même possible de générer des nouvelles solutions indépendantes de celles existantes. Pour accepter les nouvelles molécules, la formule (1.3) doit être satisfaite.

$$(\omega) + KE\omega + \delta 1 \delta 2\, buffer \geq PE(\omega 1') + PE(\omega 2') \quad (1.3)$$

Où $\delta 1$ $et \delta 2$ sont deux nombres indépendants appartient à l'intervalle [0,1]. Pour supporter le changement moléculaire requis pour la réaction une partie de l'énergie du buffer est nécessaire comme suit :

$$Buffer' = (1 - \delta 1 \delta 2)\, buffer$$

Si la formule (1.3) est vérifiée alors Mω est remplacé par les deux nouvelles molécules générées et leurs énergies cinétiques partagent aléatoirement l'énergie restante. [Lam et Li, 2010].

$$Dec = (PE\omega + KE\omega \times \delta 1 \times \delta 2 \times buffer) - (PE\omega 1' - PE\omega 2') \quad (1.5)$$

$$\text{Et} \begin{cases} KE\omega 1' = Edec \times \delta 3. \\ KE\omega 2' = Edec \times (1 - \delta 3). \end{cases}$$

Où δ3 est généré aléatoirement et qui appartiens à $[0, 1]$.

C) La substitution double cet opérateur se produit lorsque plusieurs molécules (généralement deux Mω1 et Mω2) entrent en collision puis se séparent. Le double remplacement est très similaire au remplacement simple en ce sens qu'il divise les molécules existantes $\omega 1$ et $\omega 2$ en$\omega 1`$ et $\omega 2`$ mais ne consomme pas d'énergie tampon. La condition de conservation de l'énergie peut être définie comme suit :

$$PE\omega 1 + KE\omega 1 + PE\omega 2 + KE\omega 2 \geq PE\omega 1' + PE\omega 2 \quad (1.6)$$

Pour que les changements apportés par cette réaction chimique seront acceptés. La condition 1.6 doit être satisfaite [Lam et Li, 2010]. Les énergies cinétiques des molécules transformées partagent l'énergie restante comme suit :

$$Inter = (PE\omega 1 + KE\omega 1 + PE\omega 2 + KE\omega 2) - (PE\omega 1' + PE\omega 2')$$

$$\text{Et} \begin{cases} KE\omega 1' = Einter \times \delta 4. \\ KE\omega 2' = Einter \times (1 - \delta 4). \end{cases} \quad (1.7)$$

Où $\delta 4$ est un nombre aléatoire généré dans [0,1] et *E*inter est l'énergie de buffer.

D) La synthèse fait référence au fait que certaines molécules (généralement deux $M\omega 1 et M\omega 2$) entrent en collision les unes avec les autres pour donner une nouvelle molécule $M\omega`$. Cette molécule doit satisfaire les conditions de conservation d'énergie pour la synthèse [Lam et Li, 2010] :

$$PE\omega 1 + PE\omega 2 + KE\omega 1 + KE\omega 2 \geq PE\omega' \quad (1.8)$$

Si la condition (1.8) est vérifiée, le résultat$KE\omega'$ prend tout le reste de l'énergie.

$$KE\omega' = (PE\omega 1 + PE\omega 2 + KE\omega 1 + KE\omega 2) - (PE\omega') \quad (1.9)$$

1.5.2.3 L’étape finale

Si les critères de sortie sont remplis, l'algorithme continue à partir de la dernière étape. Les critères d'arrêt sont définis en fonction des besoins et des préférences de l'utilisateur. Ceux-ci incluent le temps CPU maximal utilisé, l'obtention de valeurs de fonction objectives inférieures à un seuil défini, le nombre maximal d'itérations effectuées sans amélioration, etc. À ce stade, le processus CRO se termine par l'affichage de la meilleure solution trouvée pour cette valeur de fonction objectif.

1.6 Conclusion

Dans ce chapitre, nous présentons notre contexte de travail. Nous introduisons d'abord quelques concepts de base de l'optimisation, Puis, nous avons présenté brièvement les différentes méthodes utilisées pour la résolution des différents problèmes d’optimisation. Nous citons quelques exemples des méthodes exactes, des heuristiques et des méta-heuristiques. Après, nous avons détaillé l’algorithme CRO utilisé dans notre travail, sa source d’inspiration, les différents opérateurs et le principe de fonctionnement.

Le chapitre suivant est consacré au problème de tournées de véhicules multi-dépôts (MDVRP), et les différents travaux développés pour résoudre ce problème.

CHAPITRE 2:

Le Problème de Tournées de Véhicules Multi-dépôts

2.1 Introduction

Une des grandes préoccupations des entreprises consiste à proposer un meilleur service client et à assurer la fluidité des flux de marchandises. L'élément fondamental de tout système logistique est la gestion et la planification des réseaux de distribution des flottes de véhicules. De nos jours, L'optimisation des tournées de véhicules et localisation des dépôts est une préoccupation majeure pour les entreprises qui doivent assurer la distribution des demandes des clients.

La logistique des transports est l'une des branches les plus étudiées de la recherche opérationnelle. Le problème de tournées de véhicule (VRP) est l'un des problèmes importants dans nombreux domaines tels que l'économie, l'industrie et la production. En particulier dans le problème de routage de véhicules multi-dépôts (MDVRP), qui est un cas particulier du problème de routage de véhicules classique (VRP). En MDVRP, les véhicules servent un ensemble de clients selon des contraintes précises au moindre coût possible

Dans ce chapitre, nous introduisons la définition de problème de tournées de véhicules (VRP) classique et ses variantes. Nous nous concentrons ensuite sur le problème de tournées de véhicules multi-dépôts (MDVRP), sa définition mathématique, et les différentes méthodes développées dans la littérature pour le résoudre.

2.2 Le problème de tournées de véhicules (VRP)

Le problème de tournées de véhicules (véhicule routing problème VRP) est une extension du problème du voyageur du commerce [Dhaenens, Espinouse.2002]. Il a été introduit pour la première fois par Dantzig en 1954 sous le nom de (Truck Dispatching Problème) [Dantzig, Fulkerson ,1954]. Une flotte de véhicules de capacité limitée, chaque véhicule doit satisfaire les demandes d'un ensemble de clients. L'ensemble des clients visités par un véhicule désigne une tournée. Chaque client doit être servie une et une seule fois et chaque tournée commence et se termine au dépôt**.** [Montoya, 2016]

L'objectif du problème de tournées de véhicules est de minimiser le coût total, c'est-à-dire la somme des distances ou des temps de parcours des tournées, tout en respectant la contrainte de capacité des véhicules. Chaque tournée doit respecter les contraintes suivantes :

- Un client ne peut être servi que par un et un seul véhicule.
- Chaque véhicule effectue une seule tournée.
- Tous les clients doivent être servir.

La figure 2.1 représente un exemple de problème de vrp avec 20 clients, résolu avec 4 véhicules

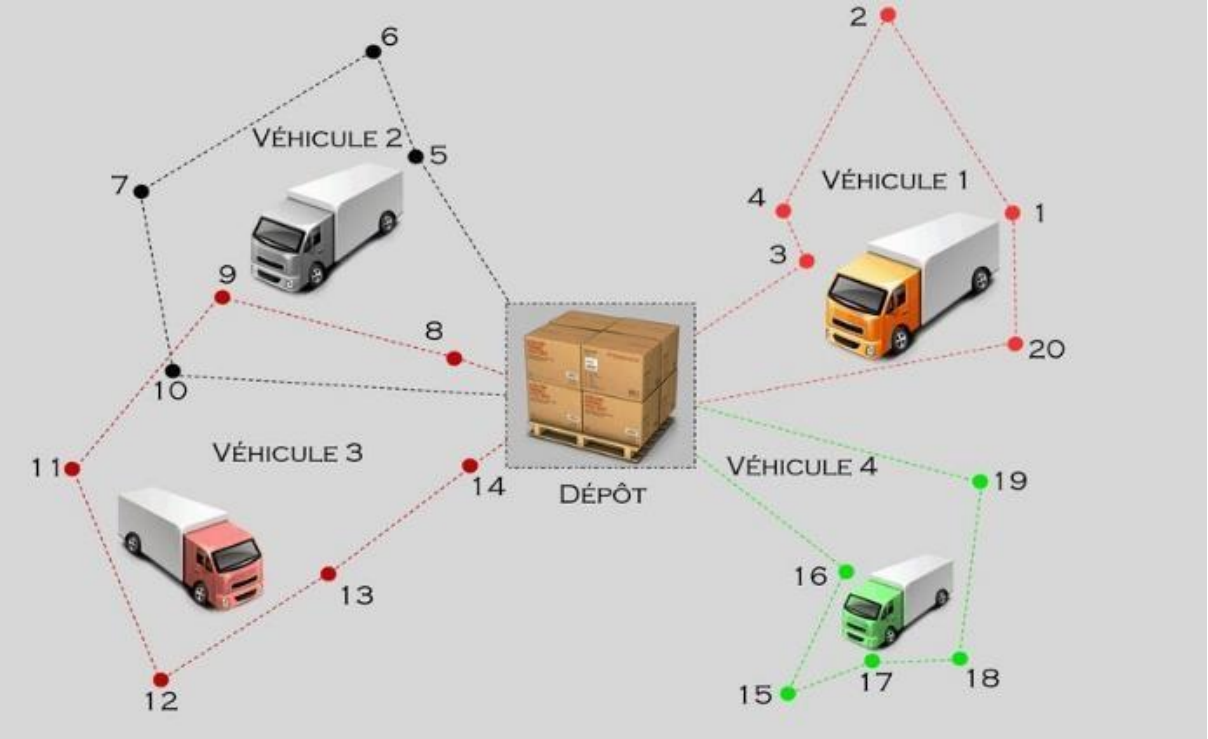

Figure2.1 : Exemple illustratif du problème de tournée de véhicules**.** [web3]

2.2.1 Les Variantes du VRP

Il existe de nombreux type de variantes du VRP selon des ajouts de contraintes au modèle de base, on peut citer :

- **Problème de tournée de véhicules avec contrainte de capacité (CVRP) :**
  Le CapacitatedVehicleRoutingProblem (CVRP) est l'une des variantes ou' une contrainte de capacité est posé sur les véhicules [Cordeau et al., 2005].
- **Problème de tournée de véhicules avec fenêtre de temps (VRPTW) :**
  Cette alternative associe une fenêtre temporelle à chaque client, et ce nouveau paramètre inclut l'intervalle de temps sur lequel les demandes des clients doivent être satisfaites. [Thangiah ,1995]
- **Problème de tournées de véhicules Split-Delivery (SDVRP) :**
  Le SDVRP exclut la nécessité de ne visiter chaque client qu'une seule fois. De cette nouvelle flexibilité résulte la possibilité de diviser la demande de service d'un seul client sur plusieurs tournées, cette liberté modifie néanmoins l'appréhension de la contrainte de capacité et alloue donc aux clients de formuler des demandes de quantité supérieur à la capacité des véhicules. **[**Archetti et Speranza ,2002]
- **Problème de tournées de véhicules avec dépôts multiples (MDVRP)** :
  Ce problème considère plusieurs dépôts au lieu d'un, dans chacun d'entre eux est affectée une partie de la flotte de véhicules. Chaque client est caractérisé par un sous-ensemble de dépôts, il ne doit être visité qu'une fois à partir de l'un d'eux. Les véhicules doivent

commencer et finir leur tournée au dépôt où ils sont affectés et ne peuvent pas visiter les autres dépôts**.** [Tricoire et Romauch ,2010].

Il existe de nombreux articles dans la littérature qui étudient des problèmes combinant plusieurs variantes du problème de tournées de véhicules, dans notre mémoire, nous nous sommes intéressés d'étudier le problème de tournées de véhicules multi-dépôts (MDVRP).

2.3 Le problème de tournées de véhicules multi-dépôts (MDVRP)

Le MDVRP est un problème célèbre formulé en 1959 par Dartzig et Ramser [Dantzig et Ramser, 1959]**,** L'objectif est de servir un ensemble de clients tout en minimisant la distance totale de déplacement tout en respectant la contrainte de capacité sur les véhicules et les dépôts. La figure 2.2 montre un exemple illustratif du problème MDVRP :

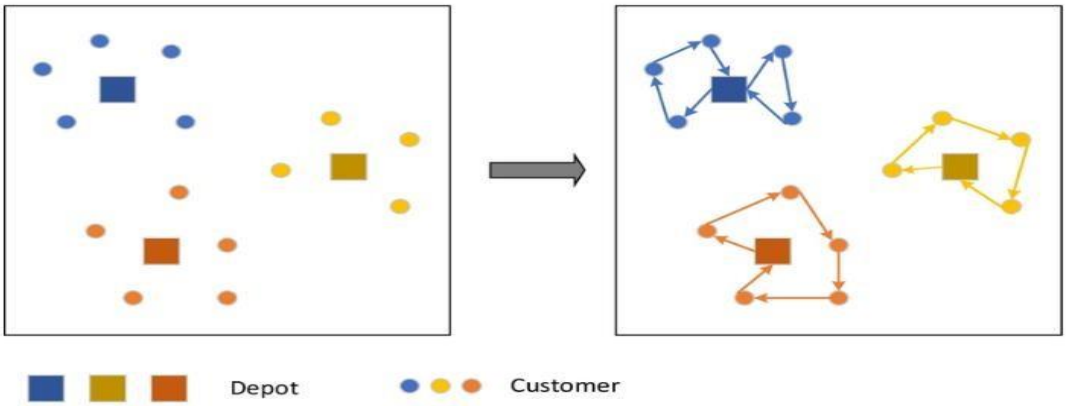

Figure 2.2 : un exemple illustratif du problème MDVRP [web4]

2.4 Formulation mathématique du MDVRP

L'objectif du MDVRP est de servir tous les clients tout en minimisant la distance totale de déplacement.Oncherche à insérer chaque client sur une tournées pour un et un seul véhicule. Tout en respectant le fait que chaque tournée doit débuter et se terminer au même dépôt et chacune tournées qui composent ces relations doit respecter les contraintes de capacités (dépôt et véhicule) .La formulation mathématique est lasuivante : [Mirabi et al., 2010].

	Modèle Mathématique du MDVRP	***Description***
Paramètres	Q_k	Capacité du véhicule sur l'itinéraire k
	V_i	capacity maximal au dépôt i
	C_{ij}	Distance entre le point $ietj, i, j \in I \cup J$
	D_J	Demande du client j
Ensembles	J	Ensemble de tous les clients
	K	Ensemble de tous les véhicules
	N	Nombre des clients
	I	Ensemble de tous les dépôts
Variable de décision	X_{ijk}	1, si le point i possède immédiatement le point j sur la route$k$ $(i, j \in I \cup J)$; sinon0
	Z_{ij}	1, si le client j est affecté au dépôt i ; 0, sinon
	U_{lk}	Variable auxiliaire pour lescontraintes d'élimination des sous-tours dans l'itinéraire
Les Contraintes	$\text{Min z} = \sum_{i \in J \cup j} \sum_{j \in I \cup J} \sum_{k \in K} C_{ij} X_{ijk}$	La fonction objectif minimise la distance totale de la livraison
	$\sum_{k \in K} \sum_{i \in I \cup J} X_{ijk} = 1, \text{j} \in \text{J}$	exige que chaque client soit affecté à une seule route
	$U_{ik} - U_{jk} + N x_{ijk} \leq N - 1, \text{l}, j \in J, k \in \text{K}$	est le nouveau ensemble de contraintes d'élimination de sous-tours (tournées incomplet)
	$\sum_{j \in I \cup \text{j}} X_{ijk} - \sum_{j \in I \cup \text{j}} X_{ijk} = 0, k \in K, i \in I \cup J$	La contrainte de conservation de capacité
	$\sum_{i \in I} \sum_{j \in J} X_{ijk} \leq 1, k \in K$	Assurer que chaque itinéraire peut être servi au plus une fois
	$\sum_{\text{j} \in \text{J}} \text{d}_\text{i} \text{Z}_\text{ij} \leq V_i, \text{i} \in \text{I}$	Les contraintes de capacité pour les dépôts
	$-Z_{ij} + \sum_{\text{u} \in \text{I} \cup \text{J}} (X_{iuk} + X_{ujk}) \leq 1, \text{i} \in \text{I}, \text{j} \in \text{J}, \text{k} \in \text{K}$	précise qu'un client peut être affecté à un dépôt uniquement s'il existe un itinéraire à partir de ce dépôt passant par ce client
	$\text{x}_{\text{ijk}} \in \{0,1\}, \text{i} \in \text{I}, \text{j} \in \text{J}, \text{k} \in \text{K}$	les exigences binaires sur les variables de décision
	$z_{ij} \in \{0,1\}, \text{i} \in I, \in \text{J}$	les exigences binaires sur les variables de décision
	$\text{U}_{\text{ik}} \geq 0, l \in J, k \in K$	Les U_{ik} les variables auxiliaires prenant des valeurs positives

Tableau 2.1: Formulation mathématique du MDVRP. [Mirabi et al., 2010].

2.5 Les méthodes de résolution de MDVRP

Le problème de tournées de véhicules multi-dépôts a attiré l'attention de beaucoup de chercheures dans la littérature. Plusieurs études sur le MDVRP ont développé des méthodes pour résoudre ce problème, ces méthodes sont généralement divisées en deux parties : les méthodes exactes et les méthodes approchées Celles-ci se divisent en deux sous-catégories, les heuristiques et les métaheuristiques. Dans la littérature,les approches exactes pour le MDVRP sont rares. En fait, la plupart des auteurs sont concentrés sur le développement de méthodes approchées pour trouver des solutions de bonne qualité dans un temps raisonnable. Dans ce qui suite, nous donnerons quelques exemples pour chacune de ces méthodes.

2.5.1 Les méthodes exactes

Certaines solutions exactes à ce problème peuvent être répertoriées dans la littérature.

- [Laporte et Arpin, 1984] ont développé des algorithmes de branchement et de bornage pour résoudre la version symétrique et asymétrique du MDVRP.
- [Bettinelli et al ,2011] ont proposé un algorithme de branchement, de coupe et de tarification pour le problème de routage de véhicules hétérogènes à dépôts multiples avec fenêtres temporelles, en considérant différentes combinaisons de stratégies de coupe et de tarification. Les résultats de trois ensembles différents de tests sur trois ensembles de données d'instances ont été explorés.
- [Contardo et Martinelli ,2014] ont présenté une nouvelle méthode exacte pour le MDVRP basée sur la fixation de variables, la génération de colonnes et de coupes et l'énumération de colonnes. La méthode proposée a été capable de prouver l'optimalité pour la première fois pour certaines instances de référence.

2.5.2 Les méthodes approchées

Vu les limites des méthodes exactes de trouver la solution optimale pour les grandes instances de MDVRP dans un délai raisonnable, la plupart des auteurs se sont concentrés sur le développement de méthodes approchés (heuristiques et métaheuristiques). On donne par la suite quelques travaux basés sur les méthodes approchées.

2.5.2.1 Les heuristiques

Dans cette section nous présentons les travaux existants pour résoudre MDVRP par divers méthodes heuristiques.

- Une approche heuristique basée sur la méthode d'économie de Clarke et Wright [Clarke et Wright, 1964] a été développée, et ses performances ont été évaluées en comparant avec la borne inférieure.
- [Gillett et Johnson, 1976] ont utilisé une procédure de clustring (heuristique de découpage) pour chaque dépôt, l'idée est de regrouper les clients de façon à former une tournée autour de chaque dépôt.

- [Renaud et al., 1996]ont adopté une méthode heuristique pour traiter le MDVRP. La méthode a construit une solution initiale réalisable, suivie d'un processus d'amélioration utilisant la recherche tabou (TS).
- [Salhi et Sari ,1997] ont proposé une méthode heuristique à trois niveaux pour résoudre le MDVRP. Le premier niveau était la construction d'une solution initiale réalisable. Le deuxième et troisième niveau consistent à améliorer les tournées de chaque dépôt (intra-dépôt), et les tournées de tous les dépôts (inter-dépôts) respectivement.

2.5.2.2 Les méta-heuristiques

Dans cette section nous citons quelques méthodes méta-heuristiques qui ont été proposé pour résoudre le MDVRP

- En 1996, Renaud, Laporte, et Boctor[Renaud et al., 1996] ont proposé une nouvelle métaheuristique pour résoudre MDVRP, utilisant une recherche tabou avec des contraintes de route et de capacité.
- **[**Sambunthan et Kachitvichayanukul, 2010], le PSO amélioré est utilisé pour résoudre MDVRP avec mutation et inertie améliorée.
- [Escobar, 2013] a résolu MDVRP en utilisant l'algorithme GranularTabuSearch (GTS), qui est basé sur l'idée bien connue de recherche granulaire introduite par Toth et Vigo en 2003 [Toth et Vigo, 2003].
- [Li et al., 2015]ont résolu en utilisant une approche méta-heuristique basé sur une recherche locale itérée. Il a été observé à partir des résultats de calcul testés que l'approche proposé surpasse les méthodes précédentes et elle est meilleure que l'utilisation de la recherche de grands voisinages, l'optimisation des essaims de particules et l'optimisation des colonies de fourmis.
- [Shuihua et al., 2016] ont proposé un nouvel algorithme génétique adaptatif pour résoudre MDVRP. Cette technique transforme les valeurs de fitness brutes en nouvelles valeurs de fitness adaptées à la sélection. La stratégie de taux adaptatif modifie les probabilités de croisement et de mutation en fonction de la valeur de fitness. Les auteurs ont utilisé 33 tests de référence MDVRP, le nombre de clients varie entre 48 et le nombre de dépôt variant de 2 à 9.
- Les recherches de Kaabachi,Jriji et Krichaen[Kaabachi et al., 2017] ont amélioré l'ACO en ajoutant une recherche locale pour résoudre le MDVRP
- [Bezerra et al., 2018] ont résolu le MDVRP grâce à des variantes de l'algorithme de recherche de voisinage variable qui est une recherche générale de voisinage variable (GVNS) et ont mesuré les performances de leur méthode de résolution uniquement sur les 10 instances de test sans contrainte de durée de tournées.

2.6 Discussion et comparaison

D'après l'étude de ces méthodes, en raison de la complexité du problème, la résolution du MDVRP jusqu'à l'optimalité prend énormément de temps.

Il existe deux points communs entre ces méthodes proposées. Premièrement, le MDVRP a été décomposé en des sous problèmes moins complexe, puis les sous-problèmes ont été résolus de manière séquentielle et itérative. Deuxièmement, les méthodes heuristiques souvent utilisent ces deux mécanismes : la construction des solutions initiales possibles et l'amélioration des solutions existantes pour obtenir de meilleurs résultats.

Le tableau suivant synthétise quelques méthodes proposées dans la littérature:

Les méthodesexactes			
Auteurs	**Type d'algorithme**	**Tailled'instancesrésolues**	
		C:nombre de client	**D: nombre de dépôts.**
L'aporte, Nobert et Aspin (1984).	Algorithme algorithmes de branchement et de bornage	C=40	d≤8
Andrea Bettinelli, Alberto Ceselli, Giovanni Righini (2011)	branch-and-cut-and-price algorithme	c≤72	d = 6
Contrado et Martinelli (2014)	La fixation de variables, La génération de colonnes et de coupes et l'énumération de colonnes	50≤c≤249	2≤d≤6

Tableau2.2 : Tableau synthétise des méthodes exactes pour MDVRP

Les heuristiques			
Auteurs	**Type d'algorithme**	**Taille d'instances résolues**	
		C:nombr de clients	**D:nombre de dépôts**
Clarke et Wright, 1964	Clarke and Wright algorithm.	c =100	d=1
Gillett et Jonhson (1976)	Création de regroupements de clients autour des dépôts. Résolution de PTV	50≤c≤249	2≤d≤5
Salhi et Sari (1977)	Une extension du mix flotte au MDVRP, plusieurs types d'échanges (inter-dépôts, intra-dépôts).	C= 360	2≤d≤9

Tableau2.3 : Tableau synthétise des méthodes heuristiques pour MDVRP.

Les méta-heuristiques			
Auteurs	**Type d'algorithme**	**Taille d'instancesrésolues**	
		C:nombre de clients	**D: nombre de dépôts.**
Renaud, Laporte et Boctor (1996).	Recherche avec tabou avec intensification et diversification.	50≤c≤360	2≤d≤9
Sumbunthan et Kachitvichayanukul (2010)	Optimisation d'essaim de particules avec plusieurs structures d'apprentissage social	50≤c≤249	2≤d≤5
Escobar (2013).	GranularTabuSearch	50≤c≤360	2≤d≤9
Ji.L et al (2015).	La recherche locale itérée.	50≤c≤249	2≤d≤5
Kaabachi, Jriji et Krichan (2017)	Optimisation amélioré de colonie de fourmis (IACO) avec recherche locale	48≤c≤360	2≤d≤9
Bezerra et etAlinaghian et Shokouhi (2018)	Une recherche recherche générale de voisinage variable (GVNS)	50≤c≤360	2≤d≤9

Tableau 2.4 : Tableau synthétise des méthodes méta-heuristiques pour MDVRP.

L'analyse que nous avons fait permet de voir jusqu'à quel point le MDVRP est un problème complexe. Nous avons discuté quelques travaux existés de la littérature pour ce problème en donnant le nombre de clients et de dépôts traités pour chaque travail.

On peut déduire qu'il n'existe pas une approche meilleure que l'autre pour résoudre le problème, et que chaque méthode a des avantages et des limites.

2.7 Conclusion

Comme nous l'avons vu dans ce chapitre, MDVRP n'est pas une tâche facile, c'est un problème NP-difficile. La prise en compte de plusieurs dépôts rend le problème plus difficile que le VRP classique. Aussi, il est intéressant de combiner les contraintes de capacité des véhicules et desdépôts pour se rapprocher le plus possible du problème réel.

Ce chapitre donne la définition de problème de tournées de véhicules classique (VRP), ces variantes. Puis, il détail de problème de tournées de véhicules multi-dépôt, sa définition formelle et sa formulation mathématique. Ensuite, des différentes approches développées dans la littérature pour résoudre MDVRP ont présenté. A la fin, nous avons fait une comparaison de ces méthodes, on a discuté le nombre des instances traités par chaque méthodes et les avantages et les imites des méthodes proposée.

Dans le chapitre suivant, notre contribution sera présentée pour résoudre le problème MDVRP.

Partie 2

Contribution etimplémentation

CHAPITRE 3:

Conception et Contribution

3.1 Introduction

La nature, en elle-même, est un système complexe et les êtres humains et leurs activités associées sont des parties de la nature. Ce système complexe fonctionne éternellement sans aucun problème parce qu'il existe des lois qui régissent les opérations de tous les composants qu'il contient. Les modes de fonctionnement de la nature sont d'excellents principes pour faire fonctionner des systèmes complexes et résoudre des problèmes. En mathématiques et en informatique. [Rogers ,1987]

Parmi les algorithmes inspirés par la natureona l'algorithme desréactions chimique (CRO) proposé par [Lam et Li, 2010] **,** est une méta-heuristique mono-objectif inspirée d'un processus naturel de transformation des substances de l'instable au stable.Une réaction chimique commence avec quelques molécules instables ayant une énergie excessive. Les molécules interagissent les unes avec les autres, à la fin, elles sont converties en molécules ayant un minimum d'énergie pour assurer leur existence**.** [Lam et Li, 2012]

Dans ce chapitre, on va présenter notre approche proposée, un algorithme inspiré des réactions chimiques amélioré (CROA-MDVRP). On décrit en premier lieu son processus et ses principales caractéristiques.Ensuite, on présente son application pour le MDVRP et l'adaptation de ses opérateurs pour ce problème.

3.2 Motivation

L'algorithme CRO, qui s'inspire des réactions chimiques, est une métaheuristique captivante qui fonctionne sur un système basé sur la population. IL s'inspire de la façon dont les réactions chimiques se produisent et utilise la répartition de l'énergie entre les molécules comme base.

Malgré qu'elle soit récente, elle représente un axe actif dans la recherche. Il est évident que CRO peut s'adapter à de nombreuses applications liées à des problèmes d'optimisation discrets ou continus dans plusieurs domaines.

Les principaux avantages de cet d'algorithme sont : [Lam et Li, 2012]

- CRO permet de déployer différents opérateurs pour répondre à différents problèmes
- La taille variable de la population permet au système de s'adapter automatiquement aux problèmes.
- La conversion de l'énergie et le transfert d'énergie dans différentes entités et sous différentes formes rendent CRO unique parmi les métaheuristiques.
- CRO peut être facilement programmée dans un langage de programmation orienté objet, où une classe définit une molécule et des méthodes définissent les types de réactions élémentaires.
- Il combine les avantages du recuit simulé (SA) en permettant l'acceptation de mauvaises solutions, ce qui peut guider la recherche de la meilleure solution (optimum global)

[Kirkpatrick et al., 1983] et les avantages de l'algorithme génétique (GA), c'est-à-dire permet une bonne combinaison entre l'exploitation de solutions et l'exploration de l'espace de recherche par le biais des opérateurs de croisement et de mutation [Goldberg, 1989].

L'originalité de notre travail réside dans l'amélioration de l'algorithme CRO pour résoudre MDVRP. Nous avons utilisé une heuristique constructive **« Route first, Cluster second »**[Laporte, 2009],[Laporte et al., 2001] inspirée de l'heuristique de l'ordre de densité proposée par [Layeb et al., 2013] pour résoudre le problème de VRP, et nous avons adapté les différents opérateurs de CRO pour notre problème.

3.3 Présentation de CROA pour MDVRP

Comme il est montré dans le chapitre précédent, Le problème MDVRP est l'un des problèmes d'optimisation dans le domaine des transports, tel que ce problème est classé parmi les problèmes d'optimisation mono-objectif avec contraintes, de Forme Linéaire et les Variables de type discrètes déterministe.

Dans CRO, chaque molécule M a des attributs principaux : une structure moléculaire ω, une énergie potentielle PE, une énergie cinétiqueKE. Par projection**,**ω représente une solution d'un problème donné**,** PE définit la valeur de la fonction objectif deω. Elle est calculée comme la somme des distances parcourus entre clients-clients et clients-dépôts afin de servir tous les clients. Tandis que KE quantifie la tolérance du système d'accepter une mauvaise solution[Sun et al., 2011].

Dans cette section, nous allons présenter notre approche proposée. Premièrement, nous décrirons le schéma général et le processus de CRO, Ensuite, nous présentons notre solution proposée et les différents opérateurs chimiques proposés

Pour résoudre MDVRP, nous avons amélioré et adapté l'algorithme inspiré des réactions chimiques, Le processus CROA-MDVRP proposé se compose de trois phases principales: initialisation de la population (molécules), les itérations et la phase finale (voir chapitre 1).

Le schéma général de CROA-MDVRP proposé et présenté dans la figure suivante :

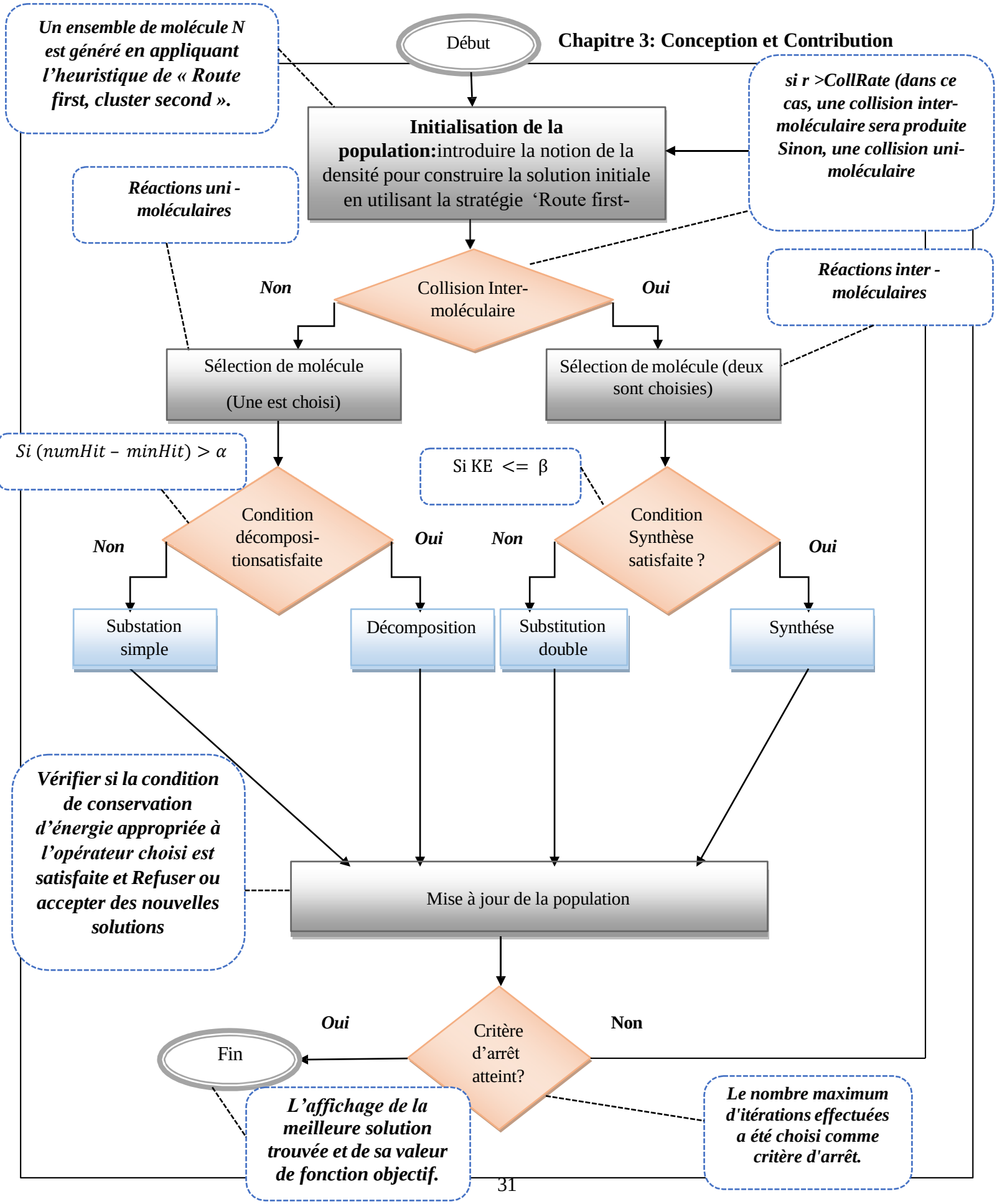

Figure3.1 : Schéma général de l'algorithme CROA-MDVRP

3.3.1 Initialisation de la population

L'algorithme proposé, commence par l'initialisation de la population en appliquant l'heuristique de « Route first, Cluster second ». De plus, Les paramètres de l'algorithme CROA-MDVRP, à savoir $PopSize, KELossRate, MoleColl, buffer, InitialKE, \alpha et \beta$. sont initialisés dans cette phase

3.3.1.1 La méthode « route first-cluster second »

L'objectif principal de cette phase est de construire une solution initiale faisable pour le problème MDVRP en se basant la matrice de la densité pour ordonner les clients. Pour cela, nous avons un ensemble de dépôts ou chacun ayant une capacité limitée $Capdep$ et$n$ clients, chacun a sa propre demande fixe di, et un ensemble de véhicules identiques avec une capacité limitée.

La solution est construite en appliquant la méthode « Route first-cluster second ». Premièrement, cette heuristique construit une tournée géante couvrant tous les clients « route first ». Dans la deuxième phase, cette tournée géante est divisée en des tournées de véhicules faisables dont la demande totale de chaque tournée ne dépasse pas la capacité du véhicule. Après avoir créé les tournées, on les affecte aux différents dépôts tout en respectant la capacité de chaque dépôt (la somme totale des demandes des tournées ne dépasse pas la capacité du dépôt)

Pour créer la tournée géante, on suppose que nous avons un dépôt avec une capacité limitée et les coordonnées $(x,\ y)$ représentant le centre de gravité de tous les dépôts, n clients, et un ensemble de véhicules avec une capacité identique.

L'algorithme commence par calculer la matrice densité entre deux clients i et j. La valeur de densité est définie comme la différence entre la capacité de dépôt et la demande des clients divisée par la valeur de distance entre deux ensembles de clients. La formule de densité est donnée par l'équation suivante : [Layeb et al., 2013]

$$Density\ (i,j) = \frac{|capdep - (di + dj)|^k}{(\text{dis(i, j)})^p\ * Density(i, \text{dépôt}) * Density(j, \text{dépôt}))} \qquad (3.1)$$

Et la densité entre le dépôt et chaque client est décrite comme :

$$Density = \frac{|Capdep - di|^k}{dist(i, \text{dépôt})^p} \qquad (3.2)$$

- $i = 1, \dots\dots\dots\dots, n;\ j = 1, \dots\dots\dots\dots, n;\ k, p \in \{1,2,3,4\}$.
- $dis(i,j)$ est la distance entre le client i et le client j.
- $density(i, depot)$est la valeur de la densité entre le client i et ledépôt.
- k et p sont des paramètres entiers qui affectent la qualité de la solution.
- diest la demande du client.

La densité est choisie comme critère d'ordonnancement des clients car elle est plus efficace que l'utilisation seule de la distance, car elle contient des informations sur la capacité dépôt, les demandes des clients, les valeurs de la distance.

La deuxième étape consiste à créer une tournée géante basée sur la matrice de densité. Le premier client inséré dans la tournée est le clienti', qui a la valeur de densité la plus élevée par rapport au dépôt $Max\,((Density(i', Dépôt)))$Le deuxième client $(inext)$ ajouté à la tournée est le client qui a la valeur la plus élevée de la densité après le premier $client\ (i')$. Nous continuons d'appliquer la même procédure jusqu'à ce que la tournée géante comprenne tous les clients.

Enfin, nous appliquons la deuxième phase de la stratégie « Route first-cluster second » qui est le "clustring" pour former des tournées de véhicules faisables. Le découpage de la tournée géante est basé sur l'ordre d'apparition des clients dans cette tournée et la capacité du véhicule.

Le processus de division prend à chaque fois un véhicule et commence à ajouter le client actuel dans la tournée si sa demande est satisfaite par ce véhicule ; sinon le client suivant est sélectionné. Ensuite, le second véhicule commence à ajouter le premier client non servi dans la tournée ; le processus est répété jusqu'à ce que tous les clients soient insérés dans des tournées faisables. Le résultat de cette étape est un ensemble de clusters ou groupes, où chaque groupe représente une tournée de véhicule qui contient un certain nombre de clients.

Après avoir créé les tournées de véhicules, on va les affecter aux différents dépôts afin d'obtenir une solution faisable au problème.

Notre objectif est de servir tous les clients au moindre coût total possible, en respectant les contraintes de capacité (véhicules et dépôts). Pour ce faire, nous devons calculer le coût total à l'aide de la fonction objectif comme suit:

- On Calcule la distance entre les clients, la distance entre les dépôts et les clients.
- Le coût de chaque tournée est calculé comme suite : la distance entre chaque client, la distance entre le dépôt et le premier client, la distance entre le dernier client et le dépôt.pour obtenir le coût d'une seule tournée, répétons le processus pour chaque tournée du dépôt
- Le coût total sera la somme des couts de touts les dépôts ouverts.

Pour créer la population initiale et générer des solutions différentes, on affecte les tournées crées par l'heuristique aux différents dépôts de manière aléatoire.

Après cette étape nous avons une population de solution ou chaque solution représente une molécule

3.3.2 La représentation de la solution

Une solution au problème MDVRP consiste à décider quels dépôts à ouvrir, les clients et les véhicules attribués à chaque dépôt et les routes à construire pour satisfaire la demande des clients avec un coût global minimum tout en respectant les contraintes de capacités.

En métaheuristique, le codage de la solution joue un rôle important dans son efficacité. Dans notre approche nous proposons de représenter chaque solution par une molécule M où sa structure moléculaire ωest représentée par une matrice à deux dimensions, La première colonne représente les dépôts et la deuxième colonne contient les tournées selon laquelle les clients doivent être visités par chaque véhicule, les tournées de chaque ligne sont placées dans un « $Vector$ », chaque ligne représente les tournées de véhicules assignées au dépôt correspondant. Les tournées sont séparées en utilisant le délimiteur« 0 ».

Cette représentation permet une évaluation rapide des solutions. De plus, il facilite la manipulation des différents opérateurs utilisés en CRO.

La figure 3.2 et 3.3 montre un exemple de solution pour MDVRP avec 50 clients, 4 véhicules avec une capacité égale à 80 et 4 dépôts où chaque dépôt à une capacité égale à 320 (égale a la capacité d'un véhicule multipliée par le nombre de véhicules affectés pour chaque dépôt) et la représentation matricielle correspondante.Cette solution utilise seulement trois dépôts. Le dernier dépôt reste fermé car les trois dépôts satisfaire tous les demandes des clients. Donc, on n'a pas besoind'ouvrir le dernier dépôt.

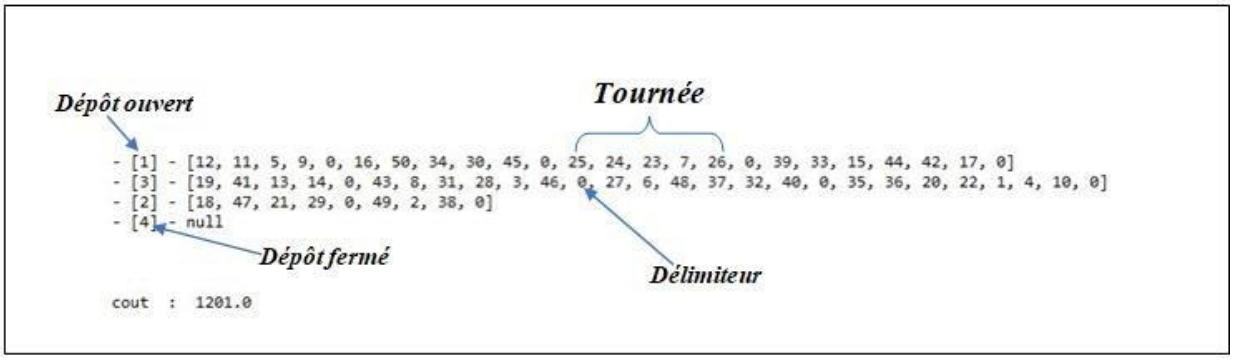

Figure 3.2 : La représentation matricielle de la solution.

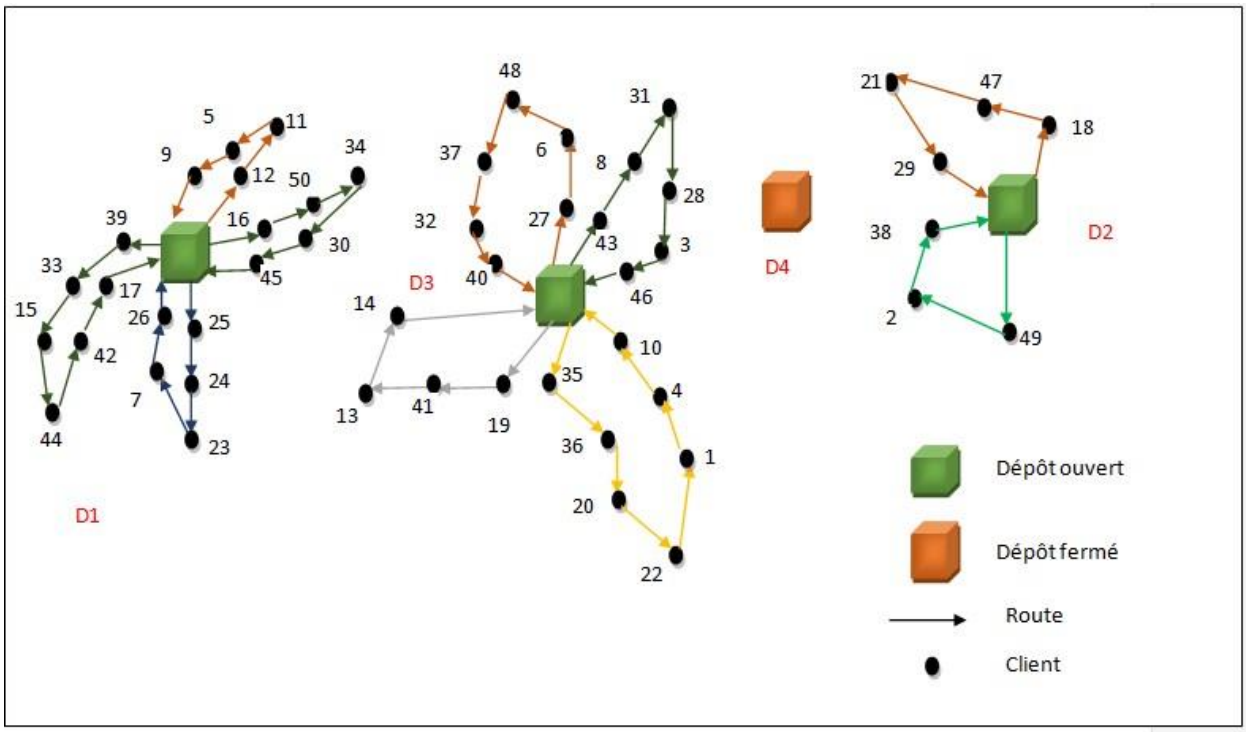

Figure 3.3 : La représentation de la solution

3.3.3 Les itérations

Après initialisation des molécules et les différents paramètres de l'algorithme, nous procédons à La deuxième étape de l'algorithme. À chaque itération, un événement se produit et les molécules entrent en collision les unes avec les autres ou avec les parois du conteneur. Cette décision est prise en générant un nombre aléatoire$r$ dans $[0, 1]$. L'algorithme choisit Si$r < CollRate$ (CollRate est un paramètre de CRO), une molécule$M$ est choisi aléatoirement et une collision d'une seule molécule se produira à ce moment. Sinon, les collisions intermoléculaires sont choisies, donc deux molécules $M1$ $etM2$ sont choisies au hasard.

Dans le cas d'une collision uni-moléculaire, CRO choisira d'appliquer la décomposition si sa condition est vérifié ($NumHit - MinHit > \alpha$). Sinon la substitution simple sera choisie.Cependant, lorsque des collisions inter-moléculaires se produisent, l'algorithme choisit entre la substitution double et la synthèse. Cette dernière s'applique si la condition de$synthèse$ ($KE \leq \beta$) est satisfaite. Sinon, deux molécules sont sélectionnées aléatoirement pour effectuer la substitution double.

3.3.3.1 Les opérateurs chimiques

Dans cette section, nous présentons les différents opérateurs chimiques proposés. Ces operateurs sont adaptés au problème de MDVRP.

A) La substitution simple

Cet opérateur modélise lasituationoù une molécule percute la paroi du conteneur et reste en une seule unité. Dans cette situation, seule la molécule existant ω était perturbée en ω' , **[** Lam et Li, 2012].Tout d'abord, nous avons transformé toutes les tournée de la solution en une seule tournée géante (dans un vecteur), nous avons choisi aléatoirement deux positions différents(P, P') dans ω et échangé les deux clients consécutives ayant la position $p \ et \ p+1$ par les deux clients consécutives ayant la position $p' et \ p'+1$. Tout en respectant les contraintes de capacité, cela signifie que la somme des demandes des clients regroupées en une tournée ne doit pas dépasser la capacité du véhiculesachant que la demande de chaque client ajoutée à chaque tournée est inférieure à la capacité des véhicules (Si la somme dépasse la capacité du véhicule, on choisit deux autres positions) , et tous les véhicules affectés à chaque dépôt ne doivent pas dépasse la capacité de ce dépôt. Comme montre la figure 3.4

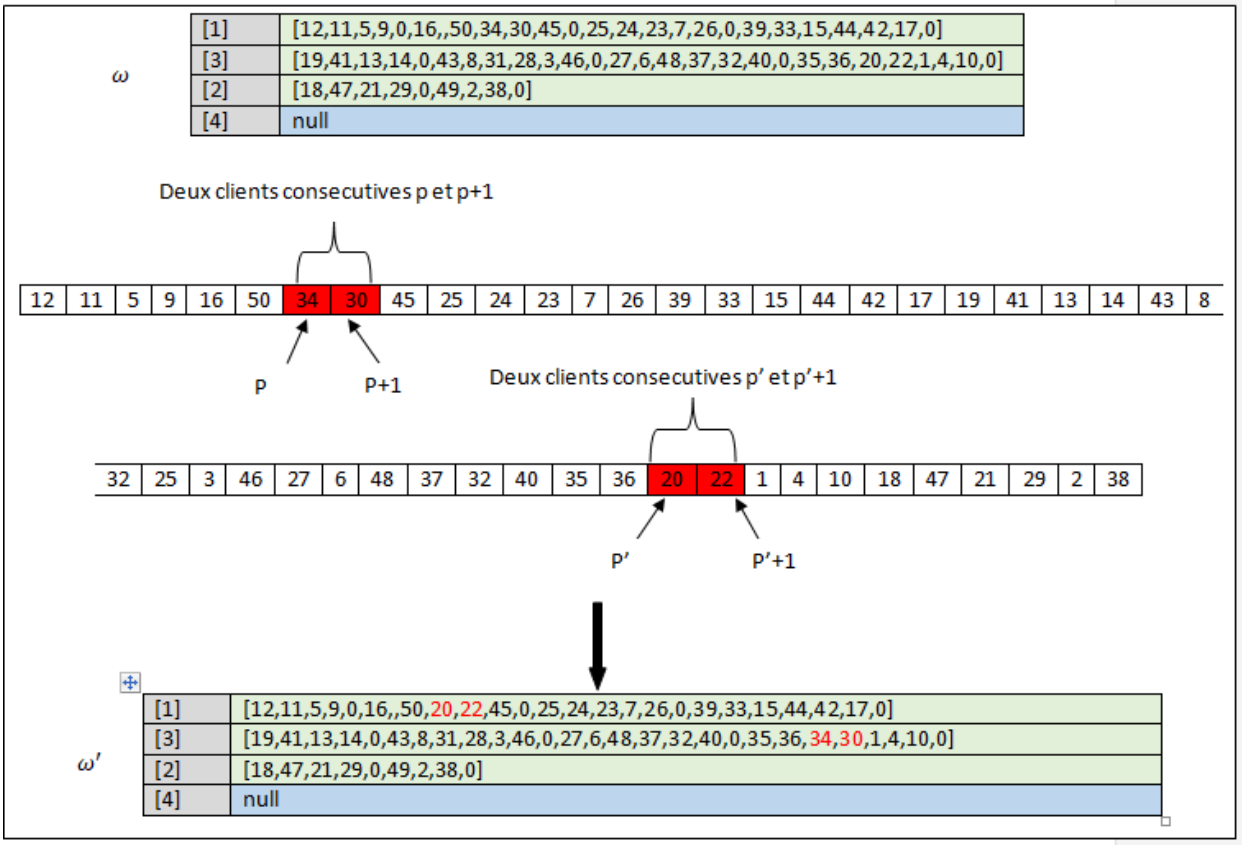

Figure 3.4 : Opérateur de la substitution simple de CROA-MDVRP

Input M

Out put $M(\omega')$

Obtenir la structure moléculaire ω de M

Créer une tournée géante à partir des tournées de véhicules existées.

Choisir au hasard deux positions P', P dans ω Echanger les deux clients consécutives ayant respectivement les positions $Petp + 1$ par les clients consécutives ayant la position $p'etp' + 1$,

Obtenir ω' et Respecter les restrictions de capacité des dépôts et des véhicules.

Si $PE(\omega) + KE\omega \geq PE(\omega')$ condition de conservation d'énergie de la substitution simple est satisfaite **Alors**Accepter la nouvelle molécule $M(\omega')$;

Remplacer $M(\omega)$ par $M(\omega')$;

Sinon Rejeter $M\ (\omega')$;

FinSi ;

Algorithme 3.1 Opérateur de substitution simple

B) La decomposition

Cet opérateur représente la situation où la molécule $M\omega$ percute la paroi et ensuite se divise en deux parties $M\omega 1'$ et $M\omega 2'$ [Lam et Li, 2010]. Dans cet opérateur, nous avons choisi une position P aléatoirement de ω , la nouvelle solution $\omega 1'$ prend la même partie $[0, P]$ de ω, la partie restante est générée aléatoirement. La même chose pour $\omega 2'$ elle prend la partie $[P, size\ (\omega)]$ de ω et la partie restante est générée aléatoirement .Tout en respectant les contraintes de capacité (véhicules, dépôts).

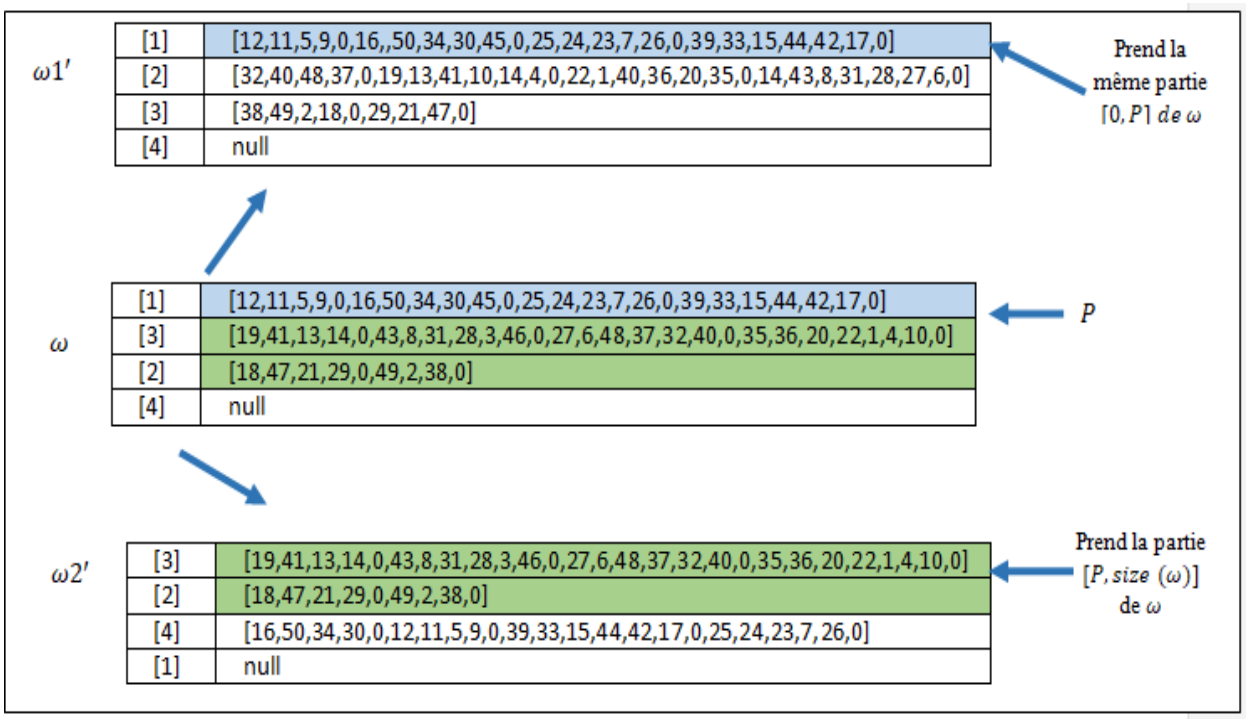

Figure 3.5 : Opérateur de la décomposition de CROA-MDVRP

Input M, $Buffer$

Output $M\omega 1'$, $M\omega 2'$, $Buffer$

Obtenir la structure moléculaire ω de M

Choisir une position aléatoire P dans ω

$\omega 1'$, $\omega 2'$ Prennent la même partie $[0, P]$, $[P, size(\omega)]$ de ω respectivement la partie restante de $\omega 1'$, $\omega 2'$ est générée aléatoirement.

Calculer $temp1 = PE\omega + KE\omega - PE\omega 1' - PE\omega 2'$

Si (condition de conservation d'énergie de décomposition est satisfaite) **Alors**

Générer $\delta 1$ et $\delta 2 [0,1] \in 3$

$Buffer = Buffer \times (1 - \delta 1 \delta 2)$

Si $PE(\omega) + KE\omega + \delta 1 \delta 2\, buffer \geq PE(\omega 1') + PE(\omega 2')$ **Alor**

Remplacer $M(\omega)$ $par M(\omega 1')$ $et M(\omega 2')$;

Mise à jour du Buffer;

Sinon

Rejeter $M(\omega 1') + M(\omega 2')$;

FinSi ;

Algorithme 3.2 : Opérateur de décomposition

C) La substitution double

La substitution double fait référer à la situation où deux molécules entrent en collision et produisent ensuite deux nouvelles molécules. Elle est également préférée pour renforcer l'intensification dans l'espace de recherche comme la substitution simple [Lam et Li, 2010]. Pour générer deux solutions $\omega 1'et\omega 2'$à partir de $\omega 1$ $et\omega 2$. Nous générons deux position de deux dépôt $p1$ $etp2$ de ωaléatoirement et nous avons échangé les clients de dépôt de position $p1$ par les client de dépôt de position $p2$ ensuite on copie les partie restant de $\omega 1'$ a partir de $\omega 1$ sans rien changer , nous Faisons le même processus pour générer une autre solution de $\omega 2'$ a partir de $\omega 2$ Tout en respectant les contraintes de capacité (véhicules, dépôts).

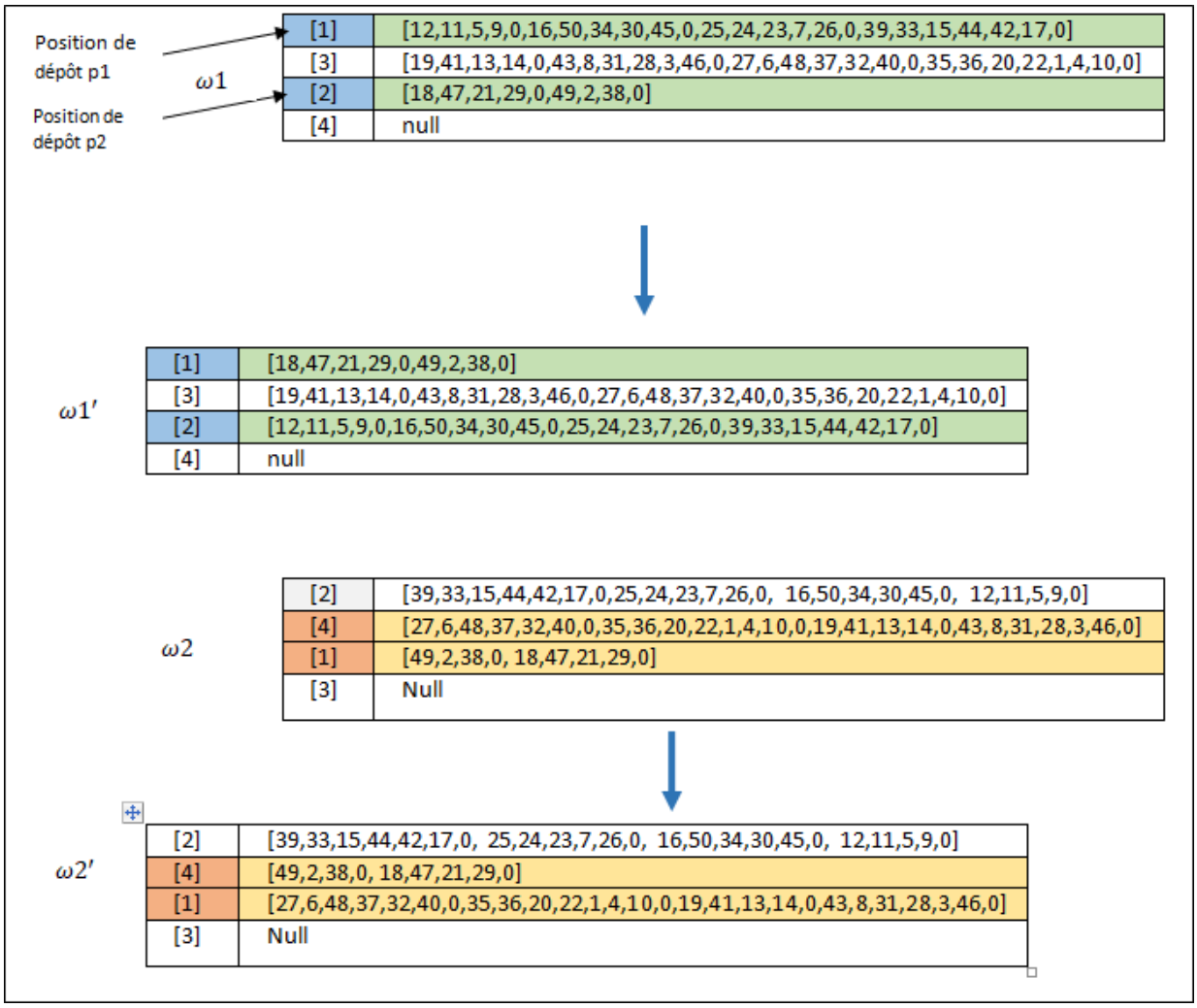

Figure 3.6 : Opérateur de la substitution double de CROA-MDVRP

Input $M1, M2$

Output $M\omega1', M\omega2'$

Obtenir les structures moléculaires $\omega1$ et $\omega2$ de $M\omega1, M\omega2$

Choisir deux position de deux dépôt p1 et p2 de ωaléatoirement, échangé les clients de dépôt de position p1 par les clients de dépôt de position p2,

On copie les partie restant de $\omega1'$ a partir de ω sans rien changer

La même chose pour générer une autre solution de $\omega2'$ a partir de $\omega2$

Si $PE\omega1 + KE\omega1 + PE\omega2 + KE\omega2 \geq PE\omega1' + PE\omega2$ Alors

Remplacer $M(\omega1)\ etM(\omega2)\ parM(\omega1')\ etM(\omega2')$;

Sinon

Rejeter M (ω1') et M (ω2')

Finsi ;

Algorithme 3.3 Opérateur de substitution double

D) La synthèse

La synthèse est le contraire de la décomposition, elle représente la situation quand deux molécules $M\omega1$ et$M\omega2$ entrent en collision et donnent une nouvelle molécule $M\omega'$. **[**Lam et Li, 2010]**.** on a généré un nombre aléatoire R, Rest compris entre [0,1] si R est entre $0\ et\ 0{,}5$ Nous avons choisi au hasard une position $P \in [0, size(\omega1) - 1]$ de$\omega1$. Ensuite, ω' prend la première partie de la solution $[0, P]$ à partir de $\omega1$ et et la seconde partie à partir de $\omega2$. sinon si$R$ est entre $0{,}5\ et\ 1$ Nous avons aussi choisi au hasard une position $P \in [0, size(\omega1) - 1] et\omega'$ prend la première partie de la solution $[0, P]$ à partir de $\omega2$ et la seconde partie à partir de $\omega1$. Tout en respectant les contraintes de capacité .

Comme montre la figure3.7 et la figure 3.8 :

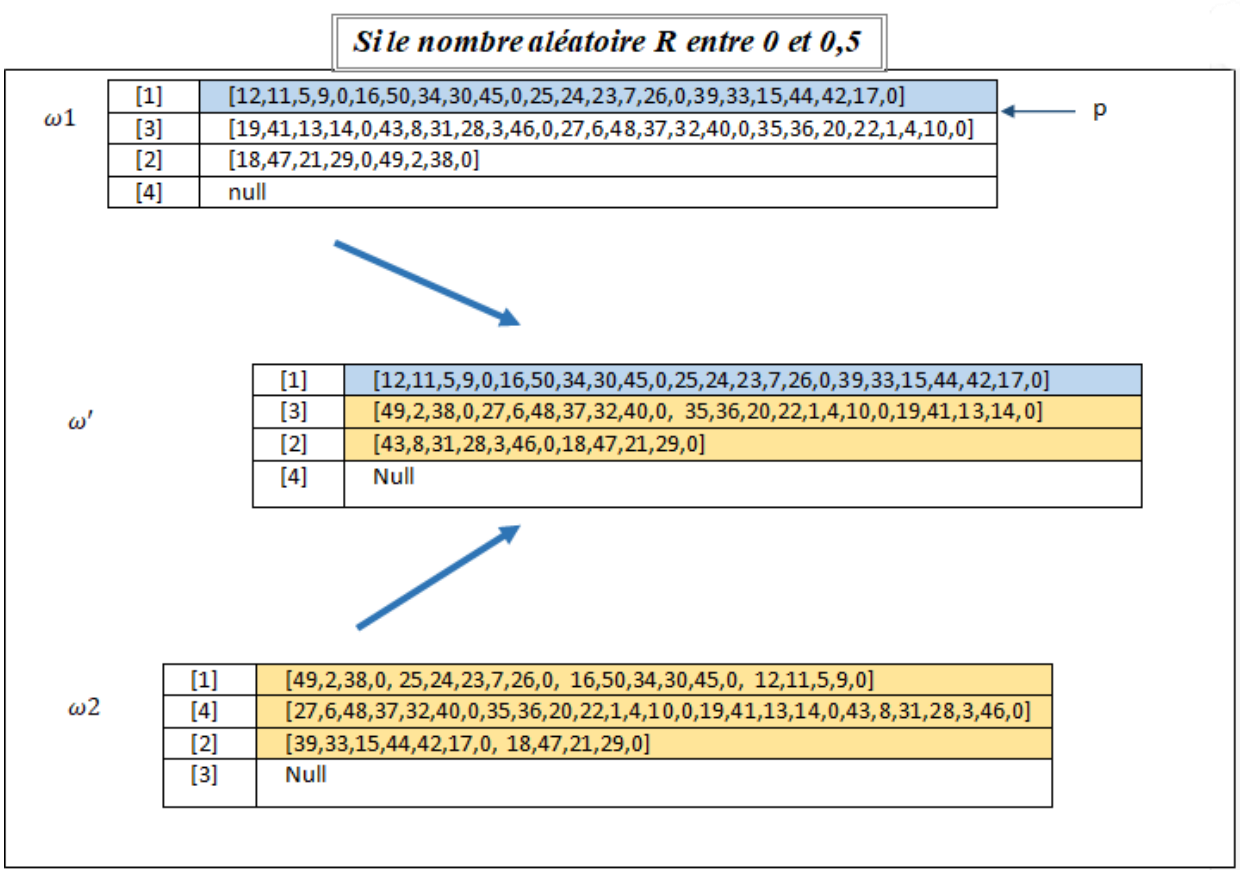

Figure 3.7 : Opérateur de synthése de CROA-MDVRP (si R entre 0 et 0,5)

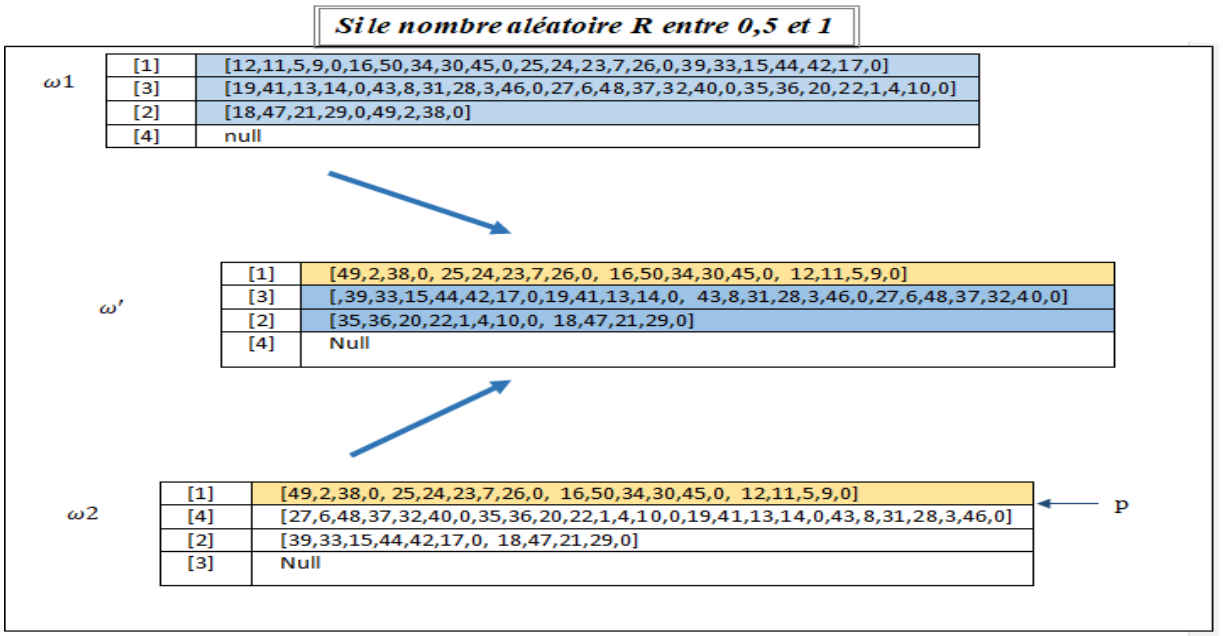

Figure 3.8 : Opérateur de synthése de CROA-MDVRP (si R entre 0,5 et 1)

Input $M1, M2$

Output $M\omega'$

Obtenir les structures moléculaires $\omega1$ et$\omega2$ $de M1, M2$

si Rest entre 0 et 0,5 ***alors***

Choisi au hasard une position$P \in [0, size(\omega1) - 1]$ $de\omega1$

$\omega' \leftarrow$Prend la même partie de la solution [0, P] à partir de $\omega1$ et la seconde partie à partir de ω2

sinonsi

choisi au hasard une position $P \in [0, size(\omega1) - 1]$ $de\omega2$

$\omega' \leftarrow$Prend la même partie de la solution $[0, P]$ à partir de $\omega2$ et la seconde partie à partir de $\omega1$

Si $PE\omega1 + PE\omega2 + KE\omega1 + KE\omega2 \geq PE\omega'$ ***Alors***

$M\ (\omega1)$ $et M\ (\omega2)$sont remplacées par $M\ (\omega')$

Sinon Rejeter $M\ (\omega')$;

FinSi;

Algorithme 3.4 : Opérateur de synthése.

3.3.4 La phase finale

Enfin, la population est mise à jour et le processus est répété jusqu'à ce qu'un critère d'arrêt soit atteint. Dans notre cas, le nombre maximum d'itérations effectuées a été choisi comme critère d'arrêt. A ce stade, le processus CROA se termine par l'affichage de la meilleure solution trouvée et de sa valeur de fonction objectif.

3.4 Conclusion

Dans ce chapitre, nous présentons notre approche proposée pour résoudre le MDVRP basée sur l'algorithme inspiré des réactions chimiques CRO.

Nous avons détaillé les différentes étapes de l'algorithme, en commençant par l'amélioration que nous avons fait dans la phase initiale de l'algorithme. Nous avons créé la population initiale en utilisant l'heuristique « route first-cluster second », on a introduire la notion de la densité pour construire la solution initiale, Apres nous avons détaillé les différents opérateurs proposés et adaptés pour le MDVRP.Dans le prochain chapitre, nous ferons une étude expérimentale et nous discuterons les résultats obtenus.

CHAPITRE 4:

Implémentation et Résultats Expérimentaux

4.1 Introduction

Ce chapitre est principalement consacré aux expériences réalisées dans le cadre de notre travail. Nous commençons par décrire les outils utilisés pour le développement du projet, tels que le choix du langage de programmation, et le matériel utilisé. Les résultats obtenus de l'application de l'algorithme CROA pour résoudre le MDVRP sont également communiqués et expliquées. Pour finir, une étude expérimentale des résultats apportés par cet algorithme est détaillée.

4.2 Environnement de développement

4.2.1 Environnement matériel

Ce travail a été implémenté sur un PC, caractérisé comme suit :

- Marque: Acer
- Un Processeur : ntel(R) Core(TM) i3-2348M CPU @ 2.30GHz 2.30 GHz
- Une RAM : 4,00 Go
- Systèmed'exploitation: Windows 10 professionnel 64 bits

4.2.2 Environnementlogiciel

- **Eclipse** : est une plate-forme open source pour les frameworks, les outils et les runtimes d'applications de développement de logiciels extensibles, créée à l'origine en tant qu'environnement de développement intégré (IDE) basé sur Java.Le système d'exécution d'Eclipse est basé sur une collection de projets open source construits par Equinox Open Services Gateway Initiative (OSGi) couvrant Java IDE, les langages statiques / dynamiques, les frameworks client épais / léger et côté serveur, la modélisation / les rapports commerciaux et Embarqués / systèmes mobiles.[web5]

4.2.3 Langage de programmation

Java : Java est un langage de programmation orienté objet, lancé par Sun Microsystems en 1995 est l'un des deux langages de programmation les plus utilisés aujourd'hui, notamment pour la création de programmes de gestion d'entreprise ou de jeux vidéo, Il s'inspire du langage C. Il comporte deux composants : la machine virtuelle Java (Java VM) et l'interface de programmation d'applications Java (Java API).Il contient un ensemble d'outils (JDK Java Development Kit) et un ensemble de packages (ensemble de classes). Ces différentes classes de base couvrent de nombreux domaines (entrées/sorties, interfaces graphiques, réseau, etc.) Les avantages de Java sont les suivants:

- ✓ Java a été conçu pour être facile à utiliser et donc facile à écrire, compiler, déboguer et apprendre que d'autres langages de programmation
- ✓ Orienté objet.
- ✓ Indépendant de la plateforme.

4.3 Description des instances

Les instances proposées pour le MDVRP utilisées dans la littérature pour évaluer les performances des algorithmes proposés pour résoudre le problème. Dans ce travail, nous avons utilisé 33 instances Les instances 1 à 7 ont été créées par Christofides et Eilon[ChristofidesetEilon, 1929], Les instances 8 à 11 ont été introduit par Gillett et Johnson [Gillett et Johnson, 1976], et les instances 12 à 23 ont été proposées par Chao et al [Chao et al, 1993]. Enfin les instances 24 à 33 ont été introduites par Cordeau et al[Cordeau et al., 1997].

La figure 4.1 représente un exemple d'un fichier de test « p02 » qui contient 50 clients, 2 véhicules et 4 dépôts, il contient aussi les données pour chaque client.

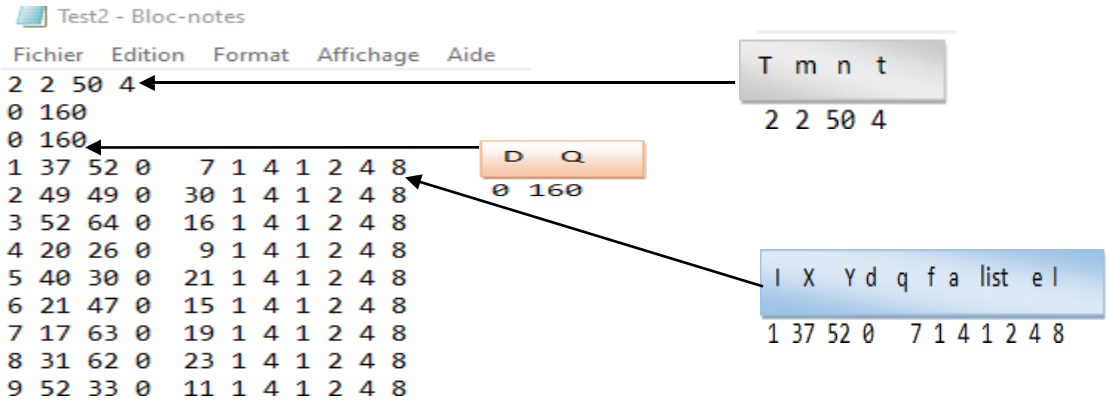

Figure4.1 : Exemple de test p02 avec 50 clients, 2véhicules et 4 dépôts.

Ces informations sont définit comme suit [web6] :

La première ligne contient les informations suivantes :

- Type t : 2(MDVRP)
- m : nombre de véhicules
- n : nombre de client
- t : nombre de dépôt

Les lignes suivantes contiennent, pour chaque véhicule, les informations suivantes:

- D : la durée maximale d'une route (inutiledans le cas de MDVRP)
- Q : charge maximale d'un véhicule (capacité du véhicule)

Les lignes suivantes contiennent pour chaque client, les informations suivantes:

- **I** : numéro de client
- **X**: coordonnées x
- **Y**: coordonnées y
- **d** : durée de service o (inutiledans le cas de MDVRP)
- **q** : demande
- **f** : fréquence de visite (inutiledans le cas de MDVRP)
- **a** : nombre de combinaison de visite (inutiledans le cas de MDVRP)
- **list** : liste de toutes les combinaisons de visites possibles (intitule dans le cas de MDVRP)
- **e** : début de la fenêtre horaire (inutiledans le cas de MDVRP)
- **l** : fenêtre horaire de fin (inutiledans le cas de MDVRP.

Les informations sur toutes les instances sont présentées dans les deux tableaux suivants tels que le tableau 4.1 représente les instances de p01 à p23 et le tableau 4.2 représente les instances de pr01 à pr10.

Instance	Nbr clients	Nbrdépôts	Nbrvéhicules	Capacité de véhicule	Capacité de dépôt
P01	50	4	4	80	320
P02	50	4	2	160	320
P03	75	5	3	140	420
P04	100	2	8	100	800
P05	100	2	5	100	500
P06	100	3	6	500	3000
P07	100	4	4	500	2000
P08	249	2	14	500	7000
P09	249	3	12	500	6000
P10	249	4	8	60	480
P11	249	5	6	60	360
P12	80	2	5	60	300
P13	80	2	5	60	300
P14	80	2	5	60	300
P15	160	4	5	60	300
P16	160	4	5	60	300
P17	160	4	5	60	300
P18	240	6	5	60	300
P19	240	6	5	60	300
P20	240	6	5	60	300
P21	360	9	5	60	300
P22	360	9	5	60	300
P23	360	9	5	60	300

Tableau 4.1 : Les informations des instances de p01 à p23. [Wang et al., 2016].

Instance	Nbr clients	Nbr dépôts	Nbr véhicules	Capacité de véhicule	Capacit de dépôt
Pr01	48	4	1	200	200
Pr02	96	4	2	195	390
Pr03	144	4	3	190	570
Pr04	192	4	4	185	740
Pr05	240	4	5	180	900
Pr06	288	4	6	175	1050
Pr07	72	6	1	200	200
Pr08	144	6	2	190	380
Pr09	216	6	3	180	540
Pr10	288	6	4	170	680

Tableau 4.2 : Les informations des instances de pr01à p10. [Wang et al., 2016].

4.4 Présentation de l'application

Nous avons créé une interface graphique qui permet d'afficher la meilleure solution trouvée et le coût minimal, pour cela nous utilisons le frameworkjavaFX sous éclipse.

JavaFX est un framework et une bibliothèque d'interface utilisateur du projet OpenJFX qui permet aux développeurs Java de créer des interfaces graphiques pour les applications.

La figure 4.2 montre la page d'accueil de notre application

Figure4.2 : page d'accueil.

Le bouton « Choisir le fichier de test » permet de spécifier le chemin d'accès au fichier, lorsqu'on clique sur ce bouton, une fenêtre sera ouverte Cette fenêtre permet de parcourir les différents répertoires de l'ordinateur, on choisit le dossier «Tests mdvrp» qui contient les 33 instances et nous sélectionnons le test que nous voulons tester .Voir la Figure 4.3

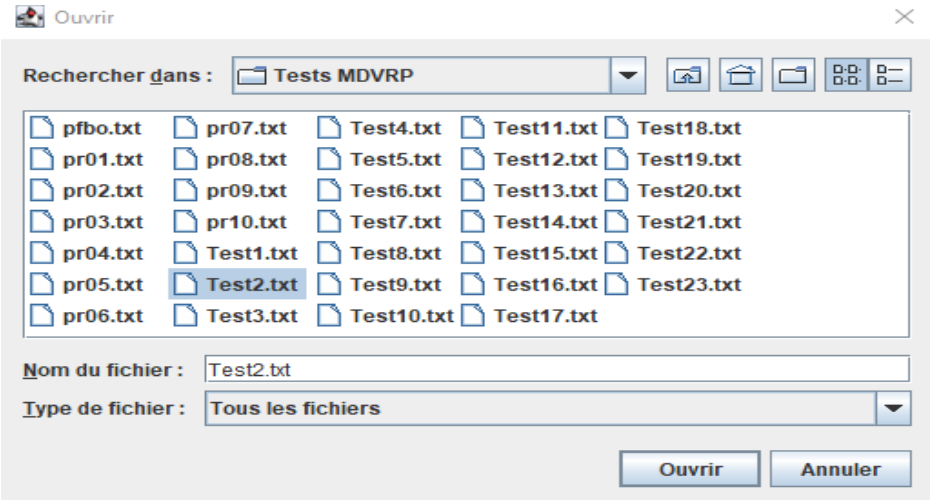

Figure4.3 :l'interface qui contient les fichiers

Ensuite, nous cliquant sur le bouton « suivant » pour passer à la deuxième interface. Cette interface contient les données de fichier: nombre de véhicules, capacité de véhicules, nombre des clients et deux tableaux, le premier tableau contient les données des clients et le deuxième tableau contient les informations sur les dépôts. Voir la figure 4.4

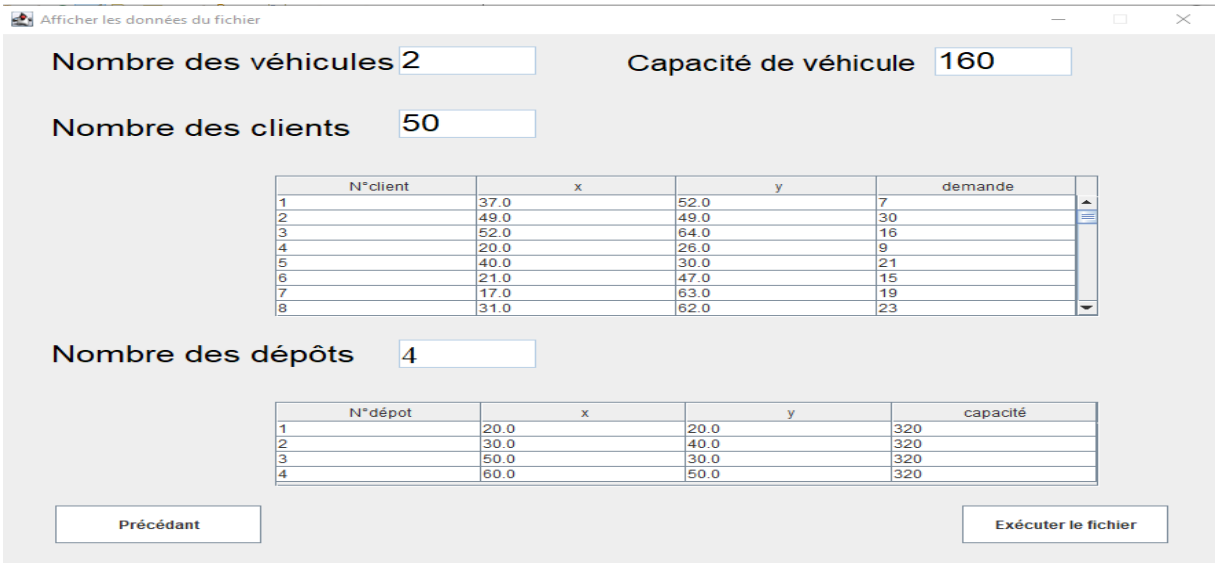

Figure4.4: Afficher les données du fichier.

Le bouton « Exécuter le fichier » permet de lancer l'exécution de l'algorithme CROA, une fois l'exécution terminée une troisième interface sera affiché, cette interface affiche la meilleure solution et le coût minimal de cette solution. Voir la figure 4.5

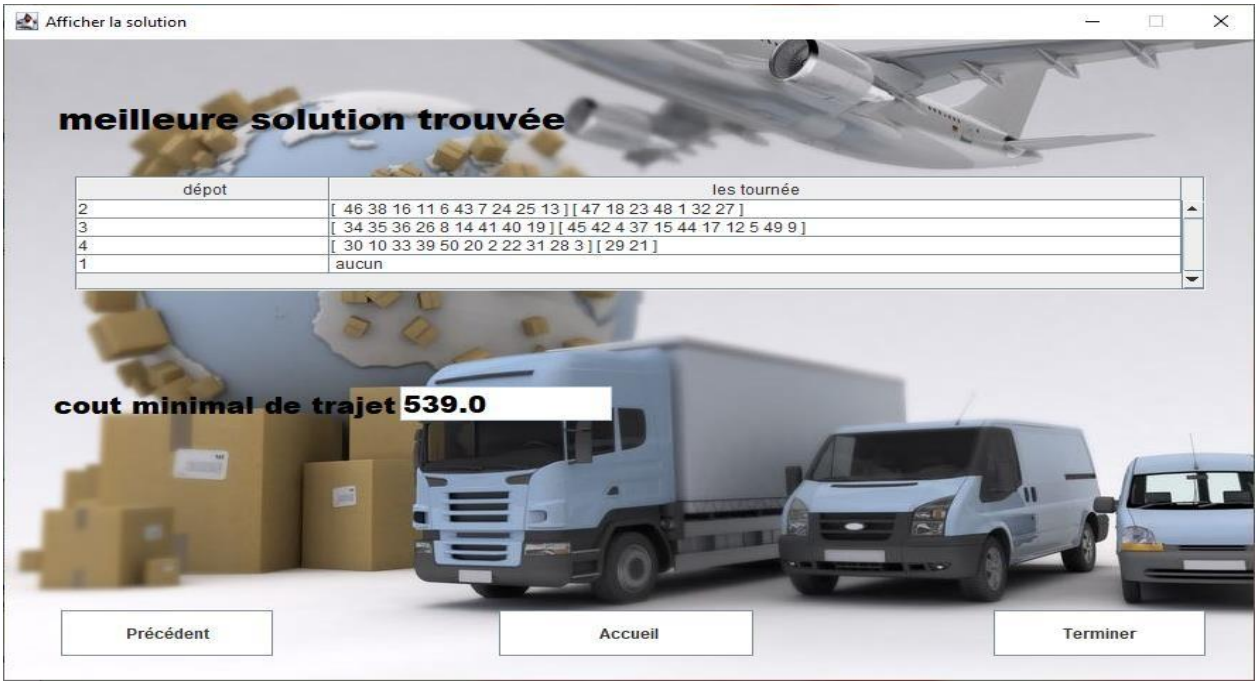

Figure 4.5 : Lameilleure solution trouvée pour le test choisi

4.5 Etude expérimentale et discussion des résultats

Nous avons testé notre algorithme CROA-MDVRP sur les 33 instances proposées dans la littérature. Pour démontrer l'efficacité de notre approche, nous comparons les résultats que nous avons obtenus avec les résultats de plusieurs algorithmes utilisés pour résoudre le même problème et avec les meilleures solutions trouvées dans la littérature (BKS). Enfin, nous analysons les résultats de chaque algorithme sur ces instances et nous avons fait une étude comparative.

4.5.1 Choix des paramètres

Le réglage des paramètres joue un rôle important dans l'efficacité d'un algorithme, il a une influence sur la qualité des résultats trouvés. CROA commence par la génération d'une population de molécules(PopSize). De plus, CRO initialise les paramètres qui sont:

$Felimite$,$CollRate$, $Initial\ energy\ buffer$, $KELossRate$, $InitKE$, α , β

Les deux paramètresα et βcontrôlent l'intensification et la diversification, le paramètre $CollRate$ a pour but d'équilibrer les collisions et a une influence importante sur les résultats obtenus.

Les résultats trouvés sont basés sur 12 exécutions indépendantes pour obtenir la meilleure combinaison de ces paramètres.

Les différents paramètres de l'algorithme CROA_MDVRP qui ont donné les meilleurs résultats trouvés sont dans le tableau 4.3

Paramètres	valeur
Felimite: Nombre d'itération	200000
Popsize : la taille initiale de la population	40
CollRate : Le paramètre responsable de choix des réactions unimoléculaires ou intermoléculaires	0.01
Initial energy buffer : L'énergie initiale de buffer	0
KELossRate : cette valeur représente le taux de perte d'énergie dans l'algorithme	0.01
InitKE : représente l'énergie cintique initiale dans l'algorithme	50000
α **(Decthres):** le seuil de décomposition	8000
β**(Synthres):** le seuil de synthèse	11000

Tableau4.3 : Les paramètres de CROA-MDVRP

4.5.2 Tests et Résultats

Pour prouver l'efficacité de notre méthode, nous comparons les résultats obtenus sur les 33 instances avec les résultats de plusieurs algorithmes et avec la meilleure solution trouvée dans la littérature (BKS : Best Known Solution).

Le tableau 4.4 présente une comparaison des résultats obtenus. La première colonne représente le nom de l'instance et la deuxième colonne montre la meilleure solution (BKS) trouvée dans la littérature et les colonnes de 3 à 8 contiennent les solutions obtenues par notre algorithme CROA- MDVRP et les différents algorithmes utilisés dans la comparaison: MHDT [Shi et al. 2020],GA [Ombuki et al., 2009], PSO [Wang et al., 2016], GTS [Escobar , 2013] , CRO-MDVRP**[**Ameur et Hadii,2022].

Instance	BKS	GA	PSO	MHDT	GTS	CRO-MDVRP	CROA-MDVRP
P01	576.87	622.18	576.87	602.05	576.87	640	643
P02	473.53	480.04	473.53	503.26	473.53	530	536
P03	641.19	706.88	641.19	672.37	641.19	720	**649**
P04	1001.59	1024.78	1001.59	1065.28	1001.04	1096	1133
P05	750.03	785.15	750.03	786.25	750.03	970	**963**
P06	876.5	908.88	876.5	912.36	856.5	1020	1029
P07	885.8	918.05	885.8	920.24	884.66	960	968
P08	4437.68	4690.18	4437.68	4537.73	4371.66	5150	**5090**
P09	3900.68	4240.08	3900.22	4065.49	3880.85	4600	4620
P10	3663.02	3984.78	3686.24	3868.75	3629.6	4359	**4223**
P11	3554.18	3880.65	3554.18	3756.28	3545.18	4250	**4237**
P12	1318.95	1318.95	1318.95	1389.65	1318.95	1419	1442
P13	1318.95	1318.95	1318.95	1375.29	1318.95	1418	1445

P14	1360.12	1365.69	1360.12	1423.96	1360.12	1405	**1398**
P15	2505.42	2579.25	2514.97	2638.19	2505.42	2610	2621
P16	2572.23	2587.87	2572.23	2689.63	2572.23	2625	2659
P17	2709.09	2731.37	2719.77	2837.26	2709.09	2800	**2742**
P18	3702.85	3903.85	3733.58	3905.32	3702.85	3998	**3914**
P19	3827.06	3900.61	3827.06	3989.24	3827.06	4010	4025
P20	4058.07	4097.06	4153.13	4268.47	4058.07	4170	4077
P21	5474.84	5926.49	5902.53	5837.28	5474.84	5647	5703
P22	5702.16	5913.59	6163.11	5996.25	5702.16	5987	**5910**
P23	6095.46	6145.58	6719.36	6356.65	6095.46	8047	**6896**

Tableau 4.4 : Comparaison des résultats des tests de p01 à p23

D'après le tableau 4.4, nous notons que nos résultats CROA-MDVRP sont meilleurs que ceux obtenus par l'algorithme CRO-MDVRP (proposé par PFE de l'année précédant) et MHDT dans plusieurs instances. Cela signifie que l'utilisation de la densité comme critère d'ordonnancement des clients dans la phase de la création de la population initiale est plus efficace que l'utilisation de la génération aléatoire de la population (utilisé dans CRO-MDVRP).

Cette amélioration faite dans la création de la population initiale donne à l'algorithme la chance de lancer la recherche avec des solutions de bonnes qualités contrairement à la méthode aléatoire, et permet de trouver des meilleures solutions par rapport à CRO-MDVRP [Ameur et Hadii, 2022]

Nous concluons qu'il n'y a pas de différence significative entre nos résultats (CROA-MDVRP), BKS et les autres algorithmes de comparaison.

La figure 4.6 montre la représentation graphique des résultats obtenus.

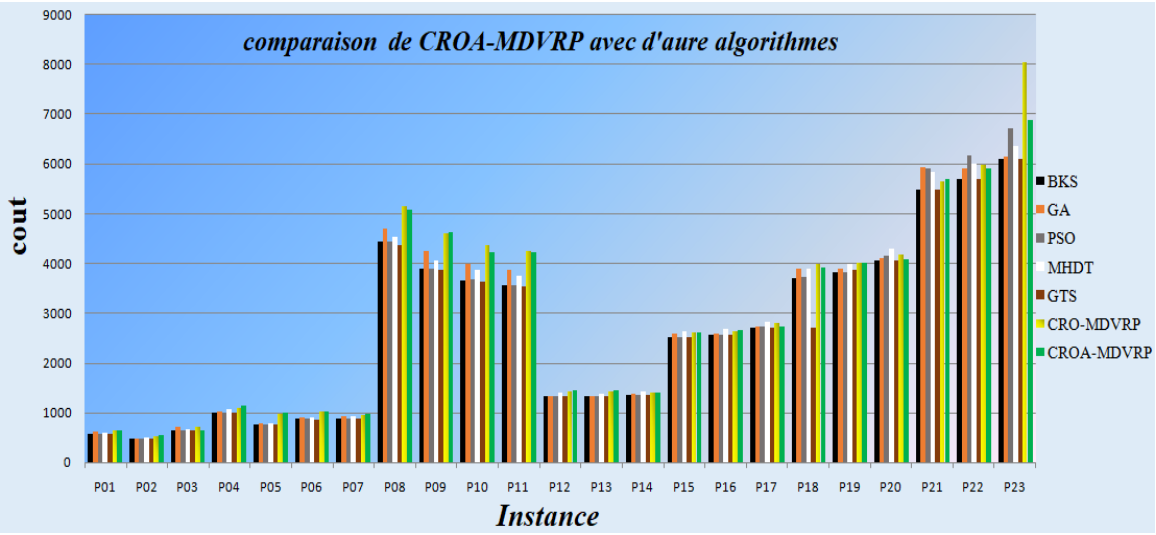

Figure4.6 : Comparaison des algorithmes pour les tests p01 à p23.

D'après cette représentation, on remarque qu'à partir de p01 à p07 et p12 à p22 nos résultats sont plus proches aux BKS et aux autres algorithmes et on trouve aussi que nos résultats sont si proches de CRO-MDVRP dans de nombreuses instances. Dans les instances p03, p05, p08, p11, p14, p17, p18, p22, p23, nos résultats sont meilleurs par rapport à CRO-MDVRPtel que le nombre de dépôts et de clients dans ces instances varie entre :

- P03 : nombre de clients (75), nombre de dépôts (5).
- P04 : nombre de clients (100), nombre de dépôts (5).
- P08 : nombre de clients (249), nombre de dépôts (5).
- P11 : nombre de clients (249), nombre de dépôts (5).
- P14 : nombre de clients (80), nombre de dépôts (2).
- P17 : nombre de clients (160), nombre de dépôts (5).
- P18 : nombre de clients (240), nombre de dépôts (6).
- P22 : nombre de clients (360), nombre de dépôts (9).

Dans les instances p08 à p11, On peut dire que nos résultats sont un peu loin aux BKS et les autres algorithmes.

Statistiquement, il n'y a pas de différence significative entre les algorithmes mentionnés dans le tableau ce qui reflète l'efficacité de l'algorithme proposé.

Instance	BKS	GA	PSO	GTS	CRO-MDVRP	CROA-MDVRP
Pr01	861.32	861.32	861.32	861.32	896	**874**
Pr02	1307.61	1307.61	1307.61	13011.11	1335	**1329**
Pr03	1806.6	1806.6	1806.6	1803.8	2012	**1935**
Pr04	2072.52	2072.52	2072.52	2064.11	2137	2140
Pr05	2385.77	2425.55	2385.77	2349.63	2499	**2488**
Pr06	2723.27	2747.71	2768.14	2710.3	2954	**2902**
Pr07	1089.56	1094.19	1089.56	1089.56	1175	**1094**
Pr08	1666.6	1666.6	1666.6	1665.5	1821	**1761**
Pr09	2153.1	2214.52	2232.65	2151.45	2340	2347
Pr10	2921.85	3079	3076.58	2910.78	3195	**3169**

Tableau 4.5: Comparaison des résultats des instances de pr01 à pr10

A travers ce tableau, on peut dire que nos résultat sont proches de BKS et comparables aux autres algorithmes GA, PS0, GTS dans toutes les instances de test.

Il est claire que les résultats obtenus par CROA-MDVRP donne des résultats meilleurs que CRO-MDVRP dans la plupart des instances, cela montre l'efficacité de l'amélioration apporté pour l'algorithme CRO. En effet, l'utilisation de la densité au lieu de la distance seule apporte des gains sur la qualité des solutions trouvées.

La figure 4.7 montre une représentation des résultats statistiquement, cela confirme qu'il n'y a pas de différence significative entre nos résultats et d'autres algorithmes.

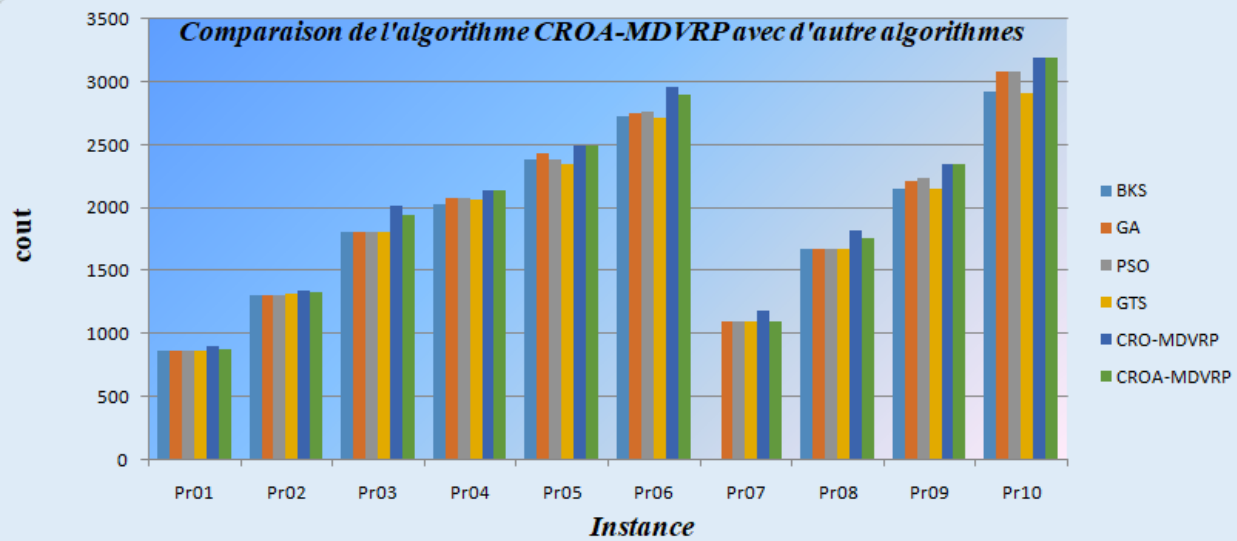

Figure4.7 : Comparaison des algorithmes pour les tests pr01 à pr10.

Dans les instances pr05, pr06, pr07, p08, pr10 nos résultats sont meilleurs par rapport à CRO-MDVRP Tel que le nombre de dépôts et de clients dans ces instances varie entre:

- Pr01 : nombre de clients (48), nombre de dépôts (4).
- Pr02 : nombre de clients (96), nombre de dépôts (4).
- Pr03 : nombre de clients (144), nombre de dépôts (4)
- Pr05 : nombre de clients (240), nombre de dépôts (4).
- Pr06 : nombre de clients (288), nombre de dépôts (4).
- Pr07 : nombre de clients (72), nombre de dépôts (6).
- Pr08 : nombre de clients (144), nombre de dépôts (6).
- Pr10 : nombre de clients (288), nombre de dépôts (6).

Selon l'étude expérimentale faite dans cette section, on peut dire que notre approche CROA-MDVRP est efficace pour résoudre le MDVRP.Ellea des performances comparables à celle des autres approches qui sont conçues pour résoudre ce type de problème.

On peut dire que l'amélioration apportée dans notre travail donne des résultats de bonne qualité plus précisément dans les instances avec un nombre de client élevé 288,240 ,360.

4.6 Conclusion

Dans ce dernier chapitre nous avons présenté les résultats d'implémentation de notre approche. Premièrement nous avons présenté l'interface graphique de notre application. Ensuite, nous avons expliqué en détails les résultats obtenus avec notre algorithme et les différents algorithmes de comparaison existés dans la littérature. Enfin nous avons fait une étude expérimentale et nous avons discuté les résultats obtenues.

Conclusion générale

Parmi les problèmes de la chaîne logistique, on peut citer le problème de transport où les entreprises de distribution doivent assurer le service de la livraison afin de satisfaire leurs clients.

Dans le cadre de ce mémoire, nous nous sommes intéressés par une variante importante de ces problèmes qui est le problème de tournées de véhicules multi-dépôts (MDVRP). Le MDVRP consiste à élaborer des tournées de véhicules associées à chaque dépôt afin de satisfaire la demande d'un ensemble de clients. L'objectif est de servir tous les clients tout en minimisant un coût total de la livraison et en respectant des contraintes de capacité imposées sur les véhicules et les dépôts.

Dans la partie état de l'art nous avons focalisé notre recherche premièrement, sur les problèmes d'optimisation combinatoire et les principales approches utilisées pour la résolution de ces problèmes. En Deuxième lieu, nous avons présenté en détail le problème de MDVRP, ces définitions, formelle et mathématique et les différentes approches proposées dans la littérature pour le traiter.Nous avons fais une étude comparative des ces méthodes afin d'extraire leurs avantages et inconvénients.

La deuxième partie est consacrée à la proposition et l'implémentation de notre approche proposée. Le travail élaboré dans ce mémoire porte essentiellement sur l'amélioration et l'adaptation de l'algorithme inspiré des réactions chimiques CRO afin de résoudre le problème de MDVRP.

Notre amélioration est basée sur l'utilisation de l'heuristique de « Route first-Cluster second » pour créer la population initiale et aussi l'utilisation de la notion de la densité pour construire la tournée géante.

Nous avons testé les performances de notre approche sur plusieurs instances du problème traité et nous avons fait une étude expérimentale détaillée pour discuter les résultats trouvés. Les résultats des tests montrent que notre approche est compétente pour résoudre le MDVRP, elle a des performances comparables à celle des autres approches qui sont conçues pour résoudre ce type de problème. Cela montre que l'amélioration que nous avons réalisée donne des résultats de bonne qualité et des fois surpassent ceux des autres algorithmes de comparaison.

Il est important de noter que l'efficacité de CRO pour les problèmes d'optimisation combinatoire peut varier en fonction de la nature du problème et des paramètres choisis. Les résultats obtenus reflètent l'importance du CRO comme outil pour résoudre d'autres problèmes d'optimisation combinatoire rencontrés dans divers domaines.

Durant ce travail, nous avonsrencontré des difficultés résumées comme suit :

- Vu qu'il n'y a pas assez de temps, il n'a pas été possible d'améliorer bien les résultats à travers des différentes méthodes de recherche locale.
- Le nombre de paramètres élevés de CRO qui rend son utilisation dans la résolution de problèmes difficile. Cela est l'inconvénient majeur à améliorer dans cet algorithme

Bibliographie

[AartsetKorst] Aarts, Korst J Simulatedannealing, Boltzmann machines a stochasticapproach to combinatorial and neural computing. Wileyedition, Chichester, 1989.

[Ameur et Hadii, 2022] Ameur N, Hadii H, Développement d'une méthode pour le problème de tournées de véhicules multi-dépôts.Memoire de fin d'étude, 2023.

[ArchettietSperanza, 2002] Archetti C, Hertz A etSperanza MG. A Tabu Search Algorithm for the Split Delivery Vehicle Routing Problem. Mathematics of Information Technology and Complex Systems, These de doctorat, 2002.

[Baptiste, 2006] BAPTISTE A, Les métaheuristiques en optimisation combinatoire, conservatoire national des arts et metiers paris, 2006.

[Benkaddour et Aribi, 2013] Benkaddour et Halima A, Métaheuristiques parallèles pour la résolution des problèmes difficiles, Mémoire de fin d'étude,2013.

[Bettinelliet al, 2011] Bettinelli A, Ceselle A, et Righini G.A Branch-and-cut-and-price Algorithm for the Multi-depot Heterogeneous Vehicle Routing Problem with Time Windows." Transportation Research Part C-Emerging Technologies, 19:723–740, 2011.

[Bezerra et al., 2018] Bezerra, S.S. A GVNS Algorithm for Solving the Multi-depot Vehicule Routing Problem. Electronic Notes in Discrete Mathematics, 66, 167-174, 2018.

[Bianchi, 2000] Bianchi L .Notes on dynamic vehicle routing - the state of the art. 2000.

[Bourazza ,2006] BourazzaS.Variantes d'algorithmes génétiques appliqués aux problèmes d'ordonnancement, Thèse Doctorat en mathématiques appliquées et informatique, Université du Havre, 2006.

[Brandao, 2006] Brandao J. A new tabu search algorithm for the vehicle routing problem with backhauls. European Journal of Operational Research, 173:540–555, 2006.

[Chao et al., 1993] Chao, I.M. A new heuristic for the multi-depot vehicule routing problem that improves upon best-known solutions .American Jornal of Mathematical and Management Scineces, 13, 371è406, 1993.

[Chen et al., 2006] Chen, Yang, G.K, Z.M Wu. Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. Journal of Zhejiang University science A. 7(4):607-614, 2006.

[ChristofidesetEilon, 1929] Christofides, Eilon. An algorithm for the vehicle-dispatching problem. Oper. Res. Q. 20(3), 309–318, 1969.

[Clarke et Wright, 1964] Clarke, Wright G, J.W. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12, 568–581, 1964.

[ContardoetMartinelli, 2014]Contardo, MartinelliR . A new Exact Algorithm for the Multi-depot Vehicle Routing Problem under Capacity and Route Length Constraints." Discrete Optimization 12: 129–146. Cordeau, J. F., M. Gendreau, and G. Laporte. 1997. "A Tabu Search Heuristic for Periodic and Multi-depot Vehicle Routing Problems. Networks 30: 105–119. 2014.

[Cordeau et al., 1997] Cordeau, J. M. A tabu search heuristic for periodic and multidepotvehicule routing problems. Networks, 30, 105-119, 1997.

[Cordeau et al., 2005] Cordeau, LaporteG,Vigo D. Savelsbergh, Vehicle Routing, Handbooks in Operations Research and Management Science.2005.

[Cordeau et Savelsbergh, 2005] Cordeau, Laporte G et M.W.P. Sa. Vehicle Routing Handbooks in Operations Research and Management Science. PhD thesis, 2005.

[CrispimetBrandao, 2005] Crispim J, BrandaoJ.Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls. Journal of the OR Society, Vol 56, no.11, pp. 1296–1302, 2005.

[DantizetRamser, 1959] Dantiz ,Ramser. The truck dispatching problem. Management Science, 6(1), 80-91, 1959.

[Dantziget Fulkerson, 1954] Dantzig, Fulkerson, Johnson S. Solution of a large –scale traveling – salesman problem. Journal of the Operations Research Society of America, 2(4): pp.393-410,1954.

[Devarenne, 2007] Devarenne I. Études en recherche locale adaptative pour l'optimisation combinatoire, Thèse de doctorat, 2007.

[Dhaenens et Espinouse.2002] Dhaenens, Espinouse ML .Problèmes combinatoires clasiques.In Recherche opérationnelle et réseaux : méthodes d'analyse spatiale Hermès Science Publications, 2002.

[Escobar, 2013] Escobar V. Heuristic algorithms for the Capacited Location-Routing Problem and the Multi-depot Vehicule Routing Problem, 2013.

[Feigenbaun et Feldman, 1963] Feigenbaum, Feldman. Computers and Though. McGraw-HillINC. 6, New York, 1963.

[Fonseca et Fleming, 1993] Fonseca C, Fleming .Genetic Algorithm for Multiobjective Optimization: Formulation, Discussion and Generalization. In Proceedings of the Fifth Internationl Conference on Genetic Algorithms. pp. 416-423. San Mateo, California. 1993.

[Ghali, 2005] Ghali k .Méthodologie de conception système a base de plateformes reconfigurables et programmables. Thèse de doctorat. Université Paris XI, France,2005.

[Gherbouj, 2013] Gherbouj A, Méthodes de résolution de problèmes difficiles académiques, Thèse Doctorat en informatique, université Constantine, 2013.

[Gillett et Johnson, 1976] Gillett, Johnson. Multi-terminal vehicule-dispatchalgorithm. Omega, 4, 711-718.1976.

[Glover, 1986] Fred G. Future paths for integer programming and links to artificial intelligence. Computer Operations Research, 13, 553-549, 1986.

[Goldberg, 1989] Goldberg D. Genetic algorithm in search, optimization, and machine learning. Addision Wesley, reading, MA, USA, 1989.

[Hafez, 1999] Hafez N. Conditions d'équilibre et gestion d'unités de transport en libre service avec demandes aléatoire, Thèse Doctorat en informatique, Université de Metz-France, 1999

[Hoff etLøkketangen, 2006] Hoff A,Løkketangen A .Creating lassosolutions for the traveling salesman problem with pickup and delivery by tabusearch.Central European, 2006.

[Hoosetstutzle, 2005]Hoos H, Stutzle T, Stochastic Local Search: Foundations and Applications, Morgan Kaufmann, 2005.

[Jacquin, 2015] Jacquin S, hybridation des métaheuristiques et de la programmation dynamiquepour les problèmes d'optimisation mono et multi-objectif : application à laproductiond'énergie, Thèse Doctorat en informatique, Université de Lille 1, 2015.

[Kabachi et al., 2017] Kaabachi, I.J. An Improved Ant Colony Optimization for Green Multi Depot Vehicle Routing Problem with Time Windows. . IEEE Journal, 26-28, 2017.

[Kennedy et Russell, 1995] Kennedy, Russell. Particle swarm optimisation. Proceedings of the IEEE International Conference Neural Networks, 4, pp, 1995.

[Kripatrick et al., 1983] Kirkpatrick, S. G. Optimization by simulated annealing. Jornal of Science, 220(4598), 671-680, 1983.

[Lacomme et al., 2005]Lacomme P, Prins, Ramdane C, .Evolutionary algorithms for periodic arc routing problems.European Journal of OR. V 165. P.535-553, 2005.

[Lam AYS, Li VOK 2010] Lam A, Li V. Chemical-reaction-inspired metaheuristic for optimization. IEEE TransEvolComput 14(3):381–399,2010.

[Lam et Li, 2010] Lam, A.-Y. a.-V. (2010). Chemical-reaction-inspired metaheuristic for optimization. IEEETransactions on Evolutionary Computation, 14(3), 381-399,2010.

[Lam et Li, 2012.] Lam, J. J. Readed-Coded Chemical Reaction Optimization. IEEE Transactions on Evolutionary Computation, 16(3), 339-353, 2012.

[Lam et Li., 2012] Lam A , Li V. Chemical reaction optimization: a tutorial Memetic Comp., pp.3–17, Springer, 2012.

[Laporte, 2009] Laporte G. Fifty years of vehicle routing. Transportation Science, 43(4): 408–416, 2009.

[LaporteetAprin 1984]Laporte, Nobert Y, and Arpin D. 1984. “Optimal Solutions to Capacitated Multi-depot Vehicle Routing Problems.” CongressusNumerantium 44: 283–292, 1984.

[LaporteetSemet, 2002] Laporte G, Semet F. Classical heuristics for the capacitated VRP. In : The vehicle routing problem. Society for Industrial and Applied Mathematics, p. 109-128, 2002.

[Layeb et al., 2013] LayebA ,Ammi M. and Chikhi S. A GRASP Algorithm Based on New Randomized Heuristic for Vehicle Routing Problem. Journal of Computing and Information Technology, CIT 21, 1, 35– 46. 2013.

[Li et al., 2015] Li j ,P.Iterated local serachembdded adaptive neighborhood selection approch for the multi-depot vehicule routing problem with simultaneous deliveries and pickups. Expert Sustems with Applications, 42(7), 3551-356,2014.

[Mao-Xiang, 2006] Mao-Xiang. Study on the model and algorithm for multi-depot vehicle schedling problem, Journal of Transportation Systems Engineering and Information Technology, Vol. 16, No. 15, pp.65–70, 2006.

[Michallet, 2013] Michallet J, Problèmes de tournées de véhicules périodiques avec contraintes de sécurité ou de qualité de service, Thèse Doctorat en optimisation, Troyes, 2013.

[Mingozzi, 2005] Mingozzi .The multi-depot periodic vehicle routing problem. Thése de doctorat, 2005.

[Mirabi et al., 2010] Mirabi M, S.F .Efficient stochastic hybrid heuristics for the multidepotvehicule routing problem. Robotics and Computer-Integrated Manufacturing, 26, 564-569, 2010.

[Montoya, 2016] Montoya Ja, Electric VehiculRoutingProblem:models and solution approchésThéseDoctorat en informatique; university d’angrs 2016.

[Mounir et Ould Ahmed, 2009] Mounir, Ould Ahmed M, Contribution à la résolution du sac-à-dos à contraintes disjonctives. Thèse Doctorat en informatique, Université de Picardie Jules Verne, 2009.

[Ombuki et al., 2009] Ombuki B, B .Usinggeneticalgorithms for multi-depotvehiculerouting. In Bio-Inspired algorithmes for the vehicule routing problem, 77-99.2009

[Papadimitriou etSteiglitz, 1982] Papadimitriou ,Steiglitz K. Combinatorial optimization - algorithms and complexity. Prentice Hall, 1982.

[Parragh et al., 2006a]Parragh S, Doerner, Hartl. A survey on pickup and delivery models Part I: Transportation between customers and depot, 2006.

[PrinsetBouchenoua, 2004] Prins C, Bouchenoua S .A Memetic Algorithm Solving the VRP, the CARP and General Routing Problems with Nodes, Edges and Arcs. Studies in fuzziness and soft computing. Vol 166, pages 65-86, 2004.

[Rego et al 1994] Regoet C, Roucairol. Le problème de tournées de véhicules : Etude et Résolution Approchée. Technical Report. INRIA, 1994.

[Renaud et al., 1996] Renaud, J. G. A tabu search heuritic for the multi-depot vehicule routing problem. Computers & Operations Research, 23, 229-235, 1996.

[Ribeiro et Maculan., 1994] Ribeiro N. Maculan.Applications of combinatorial optimization. Annals of Operations Research 50, 1994.

[Rogers, 1987] Rogers H .Theory of recursive functions and effective computability. The MIT Press, Cambridge, MA, USA.1987

[Salhiet Sari, 1997]Salhi, Sari .A multi-level Composite heuristic for the multi-depotvehiculefleet mix problem. European journal of Operational Research, 103, 95-112, 1997.

[Shi et al., 2020] Yanjun Shi, L.L. A Heuristic Solution Method for Multi-Depot Vehicule Routing-Based Waste Collection Problems. Applied Sciences, 10(7), 2403, 2020.

[Shuihua et al., 2016] Wang Sh, S.L. Fitness-scaling adaptive genetic algorithm with local search for solving the Multiple Depot Vehicle Routing Problem. Simulation, 92(7), 601-616.2016.

[SombunthanetKachitvichyanukul, 2010] Sombuntham, Kachitvichyanukul, Multi-depot vehicle routing problem with pickup and delivery requests' IAENG Transactions on Engineering Technologies, 5, 71-85.2010.

[Sun et al., 2011] Sun J, Wang Y, Li J, Gao K. Hybrid algorithm based on chemical reaction optimization and Lin-Kernighan local search for the traveling salesman problem. in Seventh International Conference on Natural Computation, pp.1518–1521, 2011.2011.

[Talbi, 2009] Talbi. Metaheuristics: From Design to Implementation, Wiley. 11.2009

[Thangiah, 1995] Thangiah .Vehicle routing with time windows using genetic algorithms. Application handbook of genetic algorithms: new frontiers, II: 253{277, 1995

[Tricoireet Romauch.2010] Tricoire F, Romauch M, Doerner F, and R, Hartl F. Heuristics for the multi-period orienteering problem with multiple time windows. Computers & Operations Research, 37:351367, 2010.

[Truong et al., 2013] Truong T, Li K, Xua Y. Chemical reaction optimization with greedy strategy for the 0-1 knapsack problem .Applied Soft Computing, 13(4):1774–1780, 2013.

[Wang et al., 2016] Wang, X.G .The min-max split delivery milti-depot vehicule routing problem with minimum service time requirement. Computers & Operations Research, 71, 110-126. Récupéré sur https://doi.org/10.1016/j.cor.2016.01.008,2016

[Zaghdoud, 2015] Zaghdoud R, Hybridation d'algorithme génétique pour les problèmes des véhicules intelligents autonomes : applications aux infrastructures portuaires de moyennetaille, Thèse Doctorat en génie informatique, Ecole centrale de Lille, 2015.

Webographie

[web1]Olivier Ezratty. "Opinions Libres" ,26 juillet 2018 https://www.oezratty.net/wordpress/2018/comprendre-informatique-quantique-complexite/

[web2]Le pseudo code de la méthode du recuit Simulé | DownloadScientificDiagram (researchgate.net)

[web3]https://www.researchgate.net/figure/Illustration-dune-Solution-VRP_fig3_337840545

[Web4]https://www.researchgate.net/figure/A-schematic-example-of-an-MDVRP-model_fig1_359368657

[web5] https://www.techno-science.net/definition/517.html

[web6] https://github.com/fboliveira/MDVRP-Instances