

UNIVERSITÉ MOULOUD MAMMERI DE TIZI-OUZOU  
FACULTÉ DES SCIENCES  
DÉPARTEMENT DE MATHÉMATIQUES



---

## MÉMOIRE DE MASTER

En Mathématiques

Option : Recherche Opérationnelle

### THÈME

**Problème de tournées de véhicules  
électriques avec contraintes de capacité**

---

PRÉSENTÉ PAR :

CHAOU THINHINANE

DEVANT LE JURY COMPOSÉ DE :

M. TALEM	Djamel	UMMTO	Président
Mme. AKLI	Meriem	UMBB	Encadrant
Mme. HARRACHE	Fazia	UMMTO	Co-encadrant
Mme. AKLOUCHE	Fariza	UMMTO	Examinateur

# REMERCIEMENTS

*Avant toute chose, je rend grâce à dieu, le Tout-puissant, pour m'avoir accordé la santé, la patience et la persévérance nécessaire à l'accomplissement de ce travail. .*

*Je tiens à exprimer ma profonde gratitude et mes sincères remerciements à mon encadrante Mme Akli Meriem, pour son soutien inestimable, sa disponibilité, ses conseils précieux et son accompagnement tout au long de la réalisation de ce mémoire de master. Je suis sincèrement reconnaissante d'avoir travaillé sous son encadrement, je remercie également ma co-encadrante Mme Harrache Fazia pour sa présence dans le cadre de ce travail, et que j'ai eu le plaisir d'avoir comme enseignante au cours de mon parcours.*

*Je remercie Mr Talem Djamel pour avoir accepté de présider le jury. Je remercie également Mme Aklouche Fariza, membres du jury d'avoir accepté l'examen et l'évaluation de ce travail. .*

*Je remercie également tous les enseignants pour la qualité de leur enseignement tout au long de ma formation .*

*Une pensée toute particulière à tous les membres de ma famille pour leurs soutiens particulièrement mes parents pour leur amour inconditionnel, prière et leur encouragements .*

*Enfin, je remercie toutes les personnes que j'ai eu la chance de rencontrer au cours de mon parcours universitaire, et qui, d'une manière ou d'une autre, m'ont soutenue, encouragée ou apporté leur aide. Leur présence a été un grand réconfort au long de ce parcours.*

# Table des matières

<b>Remerciements</b>	<b>1</b>
<b>Introduction</b>	<b>5</b>
<b>1 État de l'Art du Problème de Tournées de Véhicules</b>	<b>8</b>
1.1 Introduction . . . . .	8
1.2 Problème du voyageur de commerce . . . . .	8
1.2.1 Paramètres du PVC . . . . .	9
1.2.2 Modélisation mathématique du PVC . . . . .	9
1.2.3 Variantes du PVC . . . . .	11
1.2.4 Champs d'application du PVC . . . . .	12
1.3 Le problème de tournées de véhicule . . . . .	13
1.3.1 Champs d'application du VRP . . . . .	13
1.3.2 Paramètres du VRP . . . . .	14
1.3.3 Variantes du VRP . . . . .	16
1.3.4 Formulation mathématique du VRP . . . . .	19
1.4 Problème de tournées de véhicule électrique avec contraintes de capacité . . . . .	22
1.5 Problèmes NP-Complets . . . . .	23
1.6 Conclusion . . . . .	23
<b>2 Différentes Approches de Résolution</b>	<b>24</b>
2.1 Introduction . . . . .	24
2.2 Méthodes exactes . . . . .	24
2.2.1 Définition des Méthodes exactes . . . . .	24
2.2.2 Quelques approches exactes . . . . .	25
2.2.3 Avantages et Inconvénients des Méthodes exactes . . . . .	27
2.3 Approches Heuristique . . . . .	28
2.3.1 Définitions . . . . .	28
2.3.2 Méta-Heuristiques . . . . .	29
2.3.3 quelques approches heuristiques . . . . .	29

2.4	Approches Hybrides . . . . .	38
2.4.1	Hybridation exacte-exacte : . . . . .	39
2.4.2	Hybridation exacte-heuristique : . . . . .	39
2.4.3	Hybridation heuristique-heuristique . . . . .	39
2.5	conclusion . . . . .	40
<b>3</b>	<b>État de l'art des Réseaux de Neurones Artificiels</b>	<b>41</b>
3.1	Introduction . . . . .	41
3.2	Historique et évolution des Réseaux de Neurones . . . . .	41
3.3	Réseaux de Neurones . . . . .	42
3.3.1	Neurones Biologique . . . . .	42
3.3.2	Neurone Formel . . . . .	43
3.3.3	Apprentissage Automatique . . . . .	44
3.4	Application des Réseaux de Neurones . . . . .	45
3.5	Structure des Réseaux de Neurones . . . . .	46
3.5.1	Réseaux de Neurones non Bouclés . . . . .	46
3.5.2	Réseaux de Neurones Bouclés . . . . .	47
3.6	Architecture des Réseaux de Neurones Artificiels . . . . .	49
3.6.1	Transformers . . . . .	49
3.6.2	Réseaux de Neurones Convolutifs . . . . .	49
3.6.3	Réseaux de Neurones Récurrents . . . . .	50
3.6.4	Réseaux de Neurones Graphiques . . . . .	50
3.7	Limites des Réseaux de Neurones . . . . .	51
3.8	Conclusion . . . . .	51
<b>4</b>	<b>Approche Hybridée G-RNA pour la Résolution de CE-VRP</b>	<b>53</b>
4.1	Introduction . . . . .	53
4.2	Paramètres spécifiques au CEVRP . . . . .	53
4.2.1	Taux de consommation d'énergie . . . . .	53
4.2.2	Stations de recharge alternative . . . . .	54
4.2.3	Temps d'attente à la station de ravitaillement . . . . .	54
4.2.4	Stratégies de ravitaillement . . . . .	54
4.3	Modélisation Mathématique du CEVRP . . . . .	54
4.4	Avantages et Inconvénients . . . . .	56
4.5	Structure de l'approche hybridée G-RNA pour la Résolution du CE-VRP . . . . .	57
4.6	Approche hybridée G-RNA proposée pour la résolution du CE- VRP . . . . .	62
4.7	Conclusion . . . . .	63

<b>5</b>	<b>Analyses des Résultats</b>	<b>65</b>
5.1	Introduction . . . . .	65
5.2	Environnement d'implémentation de l'approche hybridée G-RNA . . . . .	65
5.3	Description des instances . . . . .	66
5.4	Analyse des Résultats . . . . .	69
5.5	Comparaison entre l'approche proposée G-RNA et un AG classique . . . . .	76
5.6	Conclusion . . . . .	77

# Introduction

La logistique urbaine est définie comme l'ensemble de services qui gèrent, efficacement, le mouvement des marchandises dans un milieu urbain tout en apportant des solutions innovantes à la demande. L'évolution technologique a profondément impacté le domaine de la logistique en apportant d'importants progrès. En effet, des petits robots terrestres urbains circulent en Californie pour assurer différents services, les grands messagers, comme La Poste, FedEx ou UPS, font un usage croissant des véhicules de livraison électriques dans les zones urbaines ... Cependant, ces technologies sont toujours limitées à des espaces bien définis, mais restent des solutions prometteuses.

Dans ce contexte marqué par une transition énergétique due au besoin croissant d'utilisation de technologies vertes, les véhicules électriques représentent une solution au problème d'émission de gaz à effet de serre. Toutefois, l'intégration de ces véhicules dans le système logistique soulève de nouveaux défis, notamment en matière d'autonomie limitée et de gestion de recharge. C'est ainsi qu'Erdogan et Miller-Hooks(2012) ont introduit le Green-VRP et précisé le problème de routage de véhicules électriques avec contraintes capacité(CEVRP). Le CEVRP est une extension du problème de routage de véhicules classique qui inclut, outre ses contraintes, des contraintes d'énergie et de capacité.

Le Problème de tournées de véhicules dit aussi, the Vehicle Routing Problem en terminologie anglaise, un problème fondamental en optimisation combinatoire. Il consiste à planifier efficacement des itinéraires des véhicules d'une flotte devant desservir un ensemble de clients dispersés géographiquement. Le VRP est une généralisation du problème de voyageur de commerce. L'optimisation d'un VRP se décline en deux étapes : la modélisation du problème et le choix d'une méthode de résolution.

La modélisation du VRP consiste à traduire un problème réel de distribution ou de logistique en un modèle mathématique efficace caractérisé

par l'ensemble des clients, la flotte de véhicules, le dépôt et les paramètres du problème, tels que la distance, la capacité, et la demande. Toutefois, certains problèmes réels sont difficiles à modéliser, vu leur complexité et leur dépendance environnementale.

Le choix de la méthode de résolution est une étape cruciale dans le processus de résolution d'un problème. De nombreuses méthodes ont été proposées dans la littérature, elles sont réparties, essentiellement, en trois catégories : méthodes exactes, méthodes approchées et méthodes hybrides. Plus récemment, une quatrième catégorie est apparue, il s'agit des nouvelles approches hybrides.

Les méthodes exactes garantissent des solutions optimales mais engendrent des coûts (temps de calcul, espace mémoire) importants. En revanche, les méthodes approchées fournissent des solutions efficaces en un temps raisonnable, mais ne garantissent en aucun cas l'optimalité de la solution.

Dans ce mémoire, nous allons développer les algorithmes génétiques, qui représentent une classe importante des approches méta-heuristiques, auxquels nous allons associer un processus d'apprentissage de type CNN afin d'améliorer son efficacité. Le travail est entamé par une introduction générale suivie de cinq (5) chapitres et finalisé par une conclusion générale.

Le premier chapitre fait l'objet de la présentation de l'état de l'art du problème de tournées de véhicules en exposant ses différentes composantes, sa modélisation mathématique et un bon nombre de ses variantes.

Le deuxième chapitre se portera sur les approches de résolution des problèmes d'optimisation combinatoire, à savoir, les approches exactes, les méthodes approchées, les approches hybrides.

Le troisième chapitre est axé sur les réseaux de neurones, notamment leurs définitions, leurs architectures et leurs structures.

Dans le quatrième chapitre, nous allons traiter l'approche de résolution que nous avons conçue pour la résolution d'un problème de tournées de véhicules électriques avec contrainte de capacité (CEVRP). Elle est la résultante d'une hybridation d'un algorithme génétique et un processus d'apprentissage de type réseaux de neurones convolutifs au niveau de la sélection de solutions à croiser.

Le cinquième chapitre est consacré à l'analyse des résultats obtenus par l'approche proposée pour la résolution du CEVRP.

Le travail est clôturé par une conclusion, dans laquelle une évaluation des résultats obtenus est effectuée et la définition des perspectives en conséquence.

# Chapitre 1

## État de l'Art du Problème de Tournées de Véhicules

### 1.1 Introduction

Parmi les problèmes d'optimisation combinatoire, on cite le problème de voyageur de commerce dit aussi PVC. LE PVC est un problème classique d'optimisation combinatoire. Il a été mis en avant depuis plusieurs années. Ce problème se distingue par sa formulation simple. Il admet plusieurs variantes flexibles qui seront définies et détaillées dans ce qui suit .L'ajout de contraintes spécifiques conduit à des problèmes complexes et représentatifs des situations réelles, comme le problème de routage de véhicules (PVR),qui constituera la première partie de notre chapitre.

La deuxième partie de ce chapitre sera consacrée à une variante particulière du (VRP). Cette extension est spécifique car elle est liée aux voitures électriques à capacités limitées CEVRP (Capacitated Electric Vehicle Routing Problem). Ensuite, définir la complexité du CEVRP afin de lui construire une approche de résolution.

### 1.2 Problème du voyageur de commerce

Le problème du voyageur de commerce (PVC) est un problème de transport largement étudié. Il est en apparence simple : un voyageur de commerce doit visiter un nombre ( $n$ ) de villes, en passant par chacune d'entre elles une et une seule fois, et revenir au point de départ dit dépôt [1]. L'objectif est que la longueur totale parcourue par le voyageur de commerce soit la plus courte possible et que le coût de son voyage soit minimal.

Structure sous forme d'un graphe  $G = (X, E)$  de la manière suivante :  
 $X = (x_1, x_2, \dots, x_n)$  est l'ensemble des sommets qui représente les villes à visiter.

$E = \{(x_i, x_j) \mid x_i, x_j \in X, i \neq j\}$  l'ensemble des arêtes qui lient les sommets ou les villes à visiter, autrement dit les connexions entre les villes.

Souvent, les arêtes sont associées à une fonction  $c$  de valeur positive qui représente le coût ou la longueur à parcourir.

### 1.2.1 Paramètres du PVC

#### Ensemble des villes

Représenté par un ensemble  $X$  contenant  $n$  villes que le voyageur de commerce doit visiter.

#### Point de départ

La ville où le voyageur de commence et termine sa tournée.

#### Distance intervalles

Peut-être formulée comme une matrice où chaque élément  $C(i, j)$  représente un coût ou une longueur.

### 1.2.2 Modélisation mathématique du PVC

La résolution de tout problème mathématique commence par une modélisation mathématique. Dantzig et al en 1954 [2] a proposé une formulation mathématique du PVC.

Définition des variables :

$$x_{ij} = \begin{cases} 1 & \text{si l'arc } (i, j) \text{ est utilisé dans la tournée} \\ 0 & \text{sinon} \end{cases}$$

$C_{ij}$  le coût du parcours de l'arc  $(i, j)$ .

$$\text{Minimiser } \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (1.1)$$

$$\text{sous contraintes } \sum_{i \in X} x_{ij} = 1 \quad \forall j \in X \quad (1.2)$$

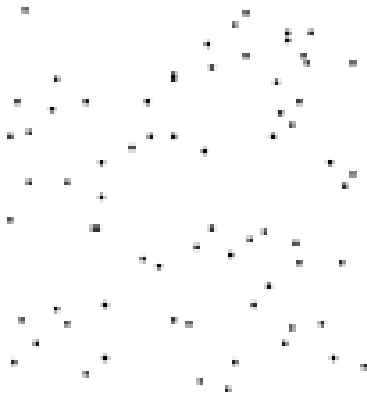
$$\sum_{j \in X} x_{ij} = 1 \quad \forall i \in X \quad (1.3)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset X, 2 \leq |S| \leq n - 2 \quad (1.4)$$

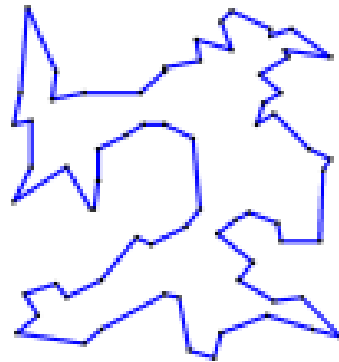
$$x_{ij} \in \{0, 1\} \quad \forall i \in X, \forall j \in X \quad (1.5)$$

Les contraintes (1.2) et (1.3) assurent que le voyageur passe une et une seule fois par chaque sommet (client, ville). La contrainte (1.5) garantit l'élimination de la formation de sous-tour (cycles partiels). L'objectif principal est de réaliser une tournée de longueur minimale (1.1).

*Remarque :* La solution du PVC est un *cycle hamiltonien* de coût minimal, c'est-à-dire un cycle qui passe une et une seule fois par chaque sommet du graphe.



**Figure 1 :** Ensemble de sommets du PVC



**Figure 2 :** Solution du PVC

### 1.2.3 Variantes du PVC

#### **PVC statique**

Un PVC est statique si toutes les données du problème sont connues avec certitude.

#### **PVC dynamique ou évolutif**

PVC dynamique : dans cette variante, les instances sont changeantes. Les conditions de trafic et la demande des clients changent au fil du temps. La prise de décision se fait en temps réel. Le PVCD peut être étendu à un problème de ramassage et de livraison dynamique (DPDP) [3].

PVC stochastique : le PVC stochastique intègre une dimension probabiliste à certains paramètres du problème. Par exemple, le cas où les dates de sortie sont inconnues. En effet, le livreur ne connaît pas le temps d'arrivée des colis (envoyés par un fournisseur). Cela s'explique par le fait que leur préparation se déroule en parallèle avec le processus de distribution, ce qui rend les dates de sortie elles aussi dynamiques.

Un autre exemple est celui des coûts inconnus à l'avance, qui sont révélés progressivement au fil du temps, renforçant l'aspect stochastique du problème [4].

#### **PVC avec contraintes temporelle**

PVC à temps dépendant (PVC-TD) : toutes les données nécessaires sont à notre disposition, y compris les coûts, les demandes et les temps de service. Ce qui caractérise cette extension, c'est que le coût d'un arc dépend du moment où il est emprunté, en fonction du temps. Par exemple, il peut y avoir des embouteillages le matin, ce qui impacte la durée totale du trajet [5].

PVCFT : le PVCFT est une variante où chaque ville doit être visitée dans un intervalle de temps donné, appelé fenêtre temporelle [6].

E-PVCFT : est une approximation de la variante précédente sur les véhicules électriques où on doit respecter le temps de recharge de la batterie du véhicule [8].

#### **PVC avec flexibilité de visite**

PVC généralisé : dans le PVC généralisé, l'ensemble des villes est divisé en plusieurs sections ou clusters, et le voyageur doit visiter exactement une ville par cluster, tout en minimisant la distance totale. Ce problème est plus complexe que le PVC classique, car il inclut une dimension supplémentaire :

le choix de la ville à visiter dans chaque cluster [9].

PVCN : cette variante divise les villes en un certain nombre de quartiers. Nous cherchons le plus court chemin pour atteindre chacun de ces quartiers sans avoir de précision sur la ville à visiter, il est donc possible d'avoir la flexibilité de visiter la ville de son choix [10].

### **PVC Hybride**

Le PVC Hybride est une généralisation du PVC classique qui intègre des contraintes spécifiques. L'exemple le plus répandu est le VRP, et le plus récent est la livraison par drone.

PVC avec drone : la propagation des drones dans les différents domaines a mené à les utiliser dans la livraison à domicile. Ainsi, la flotte considérée ici est une flotte de drones guidés par un camion, collaborant pour offrir une meilleure expérience de livraison [11].

VRP : une extension large du PVC qui intègre une flotte de véhicules pour servir les clients depuis un dépôt. L'objectif est d'optimiser la tournée en termes de coûts, de temps, etc.

## **1.2.4 Champs d'application du PVC**

Sous toutes ses formes, le PVC a pris place dans divers domaines pratiques. En logistique, aujourd'hui, les entreprises sont face à une baisse de quantité commandée alors que le nombre de commandes augmente. Ce qui nécessite une optimisation des distances parcourues, car elles sont directement liées aux coûts du déplacement.

Comme on trouve des applications du PVC dans le domaine médical, l'industrie pharmaceutique et la biologie (par exemple dans la reconstruction d'ADN) [12]. Le PVC rentre dans le domaine de l'agriculture avec ses variantes pour optimiser la répartition des tâches d'inspection sur les champs et des interventions spécifiques à certaines plantes à l'aide de robots ou de drones tout en minimisant le temps global des opérations [13].

Dans le même contexte robotique, concernant la méthode de placement automatisée des fibres (AFP) en industrie aérospatiale, optimiser le temps de travail de la machine est nécessaire. C'est là où intervient le problème du voyageur de commerce pour minimiser le temps où la machine AFP n'ajoute pas de matériau à la structure [14].

## 1.3 Le problème de tournées de véhicule

Le problème de tournées de véhicules (VRP) est une extension du PVC à une flotte de véhicules. Ces véhicules forment une flotte répartie sur un ou plusieurs dépôts et qui doivent desservir un ensemble de clients ou de villes dispersées géographiquement. La tournée d'un véhicule est définie par l'ensemble des villes (clients) qu'il a visitées en respectant le fait que chaque tournée doit commencer et se terminer au dépôt. L'objectif est de minimiser le temps total de la longueur totale des tournées tout en satisfaisant les clients.

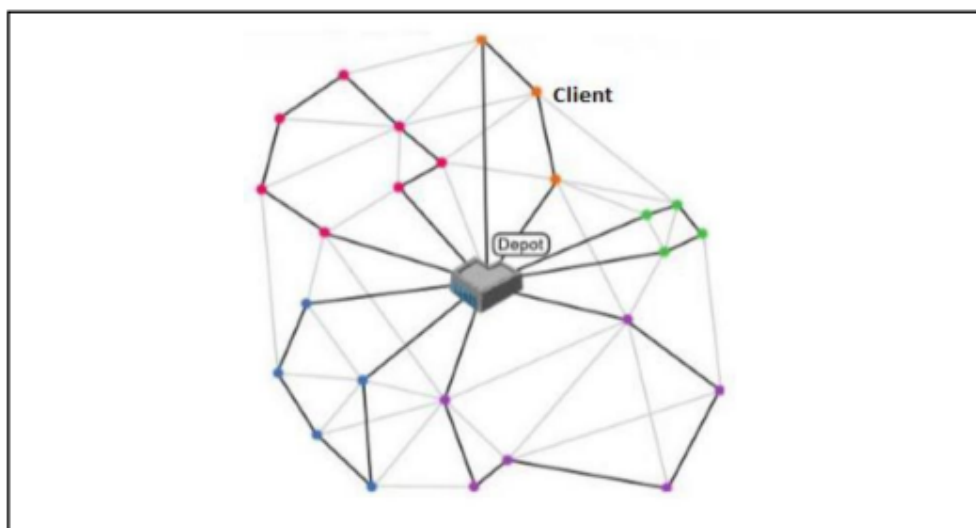


FIGURE 1.1 – VRP

### 1.3.1 Champs d'application du VRP

Le VRP est largement utilisé dans la logistique et le transport. Il rentre dans la livraison express et il est utile dans la livraison à temps réel, permet de mieux répartir les départs de livraison et dans le transport des marchandises sensibles. Il intervient dans le transport des personnes ou le transport qui nécessite un respect strict d'horaire. La logistique inverse comme le recyclage et la gestion des retours utilise le VRPB, une variante du VRP où le véhicule doit effectuer des livraisons et des collectes. Le VRP est aussi utilisé dans des domaines sensibles tels que les opérations militaires pour optimiser l'acheminement des ressources, des troupes, du matériel, ainsi que les missions de secours et d'évacuation. La data collection VRP s'applique

dans le domaine des réseaux de capture il permet de planifier le trajet pour récupérer des données des capteurs de manière efficace. Dans la surveillance environnementale, il collecte des données sur l'air, la température, etc. En général, le VRP prend place dans divers domaines, et les études sur ce sujet ne cessent de se multiplier pour répondre aux défis de la logistique moderne.

### 1.3.2 Paramètres du VRP

Les principaux paramètres qui définissent le VRP sont le dépôt, les clients, la flotte de véhicules, la matrice de distances interclients et la fonction objectif.

#### Dépôt

Le dépôt est un point de départ et de fin de la tournée d'un véhicule. Chaque véhicule doit commencer sa tournée à partir du dépôt et y revenir après avoir servi les clients. Dans le VRP classique, on considère un seul dépôt, mais cela n'empêche pas d'en avoir plusieurs comme dans le Multipl-Depot VRP

#### Clients

Les clients représentent le vif du problème, en effet ils représentent les sommets du graphe que les véhicules doivent visiter une seule fois. L'ensemble des véhicules doit couvrir toute la clientèle. Un client est caractérisé par sa localisation qui détermine la distance entre les clients et le dépôt ou entre deux clients, puis par sa demande qui peut être un service ou une marchandise qui ne doit pas dépasser la capacité  $Q_i$  du véhicule  $i$ . Le nombre de clients est un élément qui contribue à la complexité du problème. Plus le nombre de clients augmente, plus le problème devient difficile.

#### Flotte de véhicule

Elle représente l'ensemble des véhicules disponibles dans le système, le nombre de ces véhicules peut être fixe ou variable. Elle peut être composée de véhicules homogènes, qui possèdent la même capacité et des coûts identiques, ou de véhicules hétérogènes, où chaque véhicule a une capacité et un coût spécifique. Généralement, l'optimisation d'un VRP avec des véhicules hétérogènes est complexe.

#### Fonction objectif

Le VRP dans son allure classique vise à minimiser la distance totale parcourue, le coût de transport et le nombre de véhicules utilisés. Cependant, d'autres objectifs apparaissent selon les besoins des entreprises ou des clients,

et d'autres fois selon les défis qui surgissent avec l'évolution technologique. On cite ci-dessous quelques-uns :

Minimiser

- Le temps total du parcours.
- Le temps d'attente et le temps d'arrêt au dépôt ou chez les clients.
- Le déséquilibre des charges entre les dépôts.
- La consommation d'énergie et l'utilisation des stations de recharge.
- Le temps de fonctionnement des véhicules pour le VRP électrique
- L'espérance des coûts.

Maximiser

- Le profil total
- La satisfaction des clients
- L'efficacité de chargement de véhicule
- L'autonomie des véhicules dans le cas des véhicules électriques.

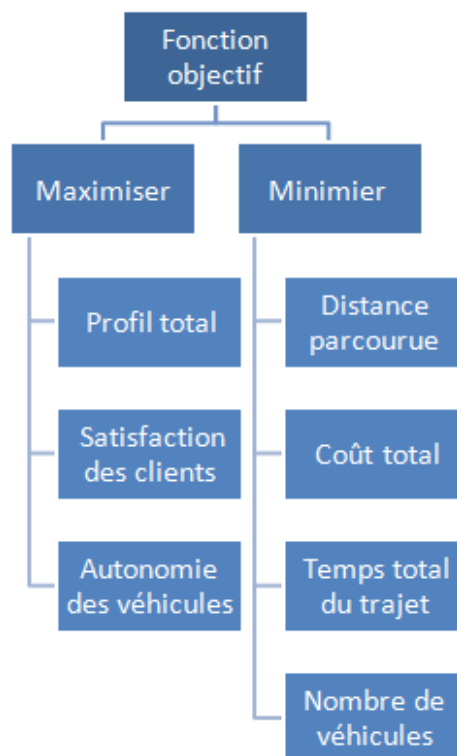


FIGURE 1.2 – Différentes fonctions objectifs

### 1.3.3 Variantes du VRP

Dans ce qui suit, nous allons présenter les différentes variantes du VRP.

#### **Selon des contraintes de capacité**

**CVRP** : Ce problème se décrit comme un VRP, tel que, chaque véhicule possède une capacité limitée, et la flotte est homogène, avec une capacité bien définie. Le CVRP a plusieurs applications, la planification des livraisons des entreprises logistiques, livraisons avec des véhicules aériens sans pilote et également dans les problèmes de ramassage [15].

**CVRPTW** : Une variante à laquelle on ajoute une contrainte de temps, appelée fenêtre temporelle, et pour laquelle on doit calculer l'heure d'arrivée, et une contrainte de capacité. L'objectif global est de minimiser le coût total, la distance et le temps associés à toutes les visites en véhicules. Cette contrainte supplémentaire reflète des situations réelles où les visites doivent être effectuées à des moments précis, comme l'optimisation des horaires des véhicules de transport public [16].

**CEVRP** : optimisation des itinéraires des véhicules électriques Tout en tenant compte de leur autonomie et de la capacité limitée des stations de recharge [17].

#### **Selon la structure du réseau**

**Dépôt-Multiple VRP** : c'est un VRP qui utilise plus d'un dépôt pour l'approvisionnement.

**Open-VRP** : ce cas se distingue par le fait que les véhicules ne doivent pas nécessairement terminer leur tournée au dépôt après avoir desservi le dernier client. Le OVRP est adapté aux entreprises qui ne possèdent pas de flotte de véhicules ou dans le cas où la flotte ne satisfait pas la demande du client, par exemple le transport de marchandises à l'exportation.

#### **Selon la flotte du véhicule**

**HFVRP** : le VRP à flotte hétérogène se caractérise par la variabilité des caractéristiques des véhicules. Cela signifie que certains véhicules peuvent avoir des capacités différentes, des coûts d'exploitation variés ou encore une consommation d'énergie distincte pour chaque véhicule [18].

**EVRP** : une flotte de véhicules électriques, des contraintes supplémentaires sont prises en compte selon le type de cette flotte.

VRPD : consiste à optimiser les délais de livraison dans une chaîne logistique composée de véhicules et de drones qui fonctionnent sans synchronisation (véhicules et drones livrent indépendamment) ou avec synchronisation. (Les véhicules et les drones effectuent des livraisons dépendantes).

### **Selon la demande**

DVRP : le VRP dynamique suppose que la demande du client n'est pas connue depuis le début et elle peut changer au fil du temps. Ce qui signifie qu'à tout moment, de nouvelles demandes peuvent apparaître [19].

SVRP : une variante qui intègre une part d'aléatoire ou d'incertitude de la demande. Le livreur ne connaît la demande qu'au moment de la livraison.

SDVRP : une extension où le client peut être servi plusieurs fois par différents véhicules [20].

VRP-Pickup and Delivery : le véhicule doit non seulement faire sa tournée et servir le client mais aussi collecter des marchandises depuis le client et les rapporter au dépôt, comme le font les camions qui livrent des bouteilles de boisson en verre et récupèrent les bouteilles vides [21].

### **Selon le temps**

Time Dependent-VRP : consiste à concevoir la meilleure route lorsque le temps du trajet sur un arc varie en fonction des spécificités de la route, telles que les embouteillages, les conditions météorologiques ou les obstacles mobiles [22].

VRP Time Deadlines : dans ce cas, on impose un délai avec une heure maximale de service. Ce problème est courant dans les livraisons urgentes (médicaments, colis express, etc).

VRP With Multiple Time Windows : c'est un problème où un client peut avoir plusieurs créneaux horaires pour être servi [24].

### **VRP Particulier**

VRP Collaboratif : Dans le transport de produits pharmaceutiques, plusieurs extensions fusionnent pour résoudre un problème donné. Par exemple, le transport de médicaments nécessite des véhicules frigorifiques, des fenêtres de temps et des multi-dépôts, ce qui correspond à un MDVRPTW (Multi-

Depot VRP with Time Windows). En vue des ressources limitées (véhicules frigorifiques, coûts élevés), on collabore avec un OVRP (Open VRP). Comme les clients sont répartis en clusters, (CluVRP) est nécessaire [25].

Crowdshipping VRP : Il s'agit d'un concept de livraison où, en plus des véhicules appartenant à l'entreprise, on fait appel à des chauffeurs occasionnels. Ce sont des personnes ayant du temps libre qui décident d'effectuer des services de livraison, souvent via des plateformes digitales. En Algérie, on peut citer des exemples comme Yassir ou DZ-Express [26].

Green VRP : Une extension du VRP qui vise à réduire l'impact des émissions sur l'environnement écologique, et cela en minimisant la consommation de carburant ou en adoptant des véhicules à carburant alternatif (VCB) uniquement, tels que l'électricité, le gaz naturel et l'hydrogène. Ou avec les véhicules à combustion interne existants dans les flottes [27].

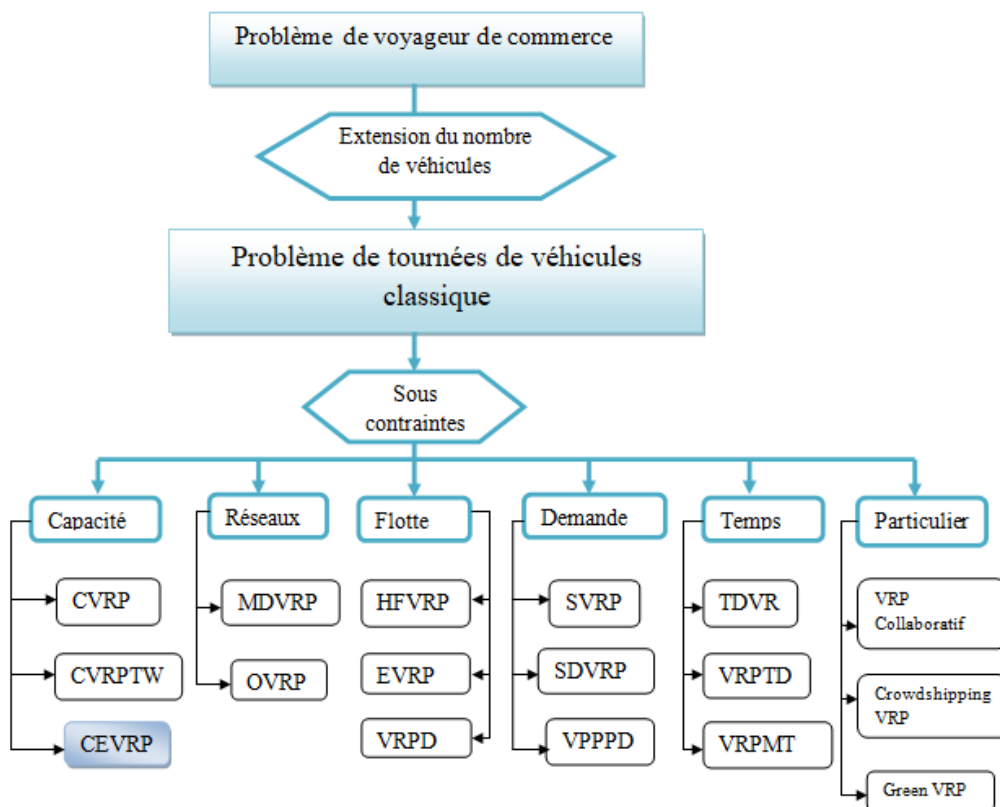


FIGURE 1.3 – variantes du VRP

### 1.3.4 Formulation mathématique du VRP

Le VRP classique peut être formulé de plusieurs manières. Le point commun est sa représentation graphique  $G=(X, U)$  qui représente le réseau routier. ou :

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

Ci-dessous deux des formulations les plus utilisées, suivies des paramètres correspondants :

#### Formulation 1

La formulation mathématique du VRP classique la plus utilisée dans la littérature est celle proposée par Laporte, Rego and Roucairol, Toth and Vigo, Crainic and Semet [28]. En effet, elle nécessite la définition de  $n \times n$  variables décisionnelles de type binaire, à trois indices suivantes :

$$x_{ij}^k = \begin{cases} 1 & \text{si l'arc } (i, j) \text{ est parcouru par le } k^{\text{ème}} \text{ véhicule} \\ 0 & \text{sinon} \end{cases} \quad (1.6)$$

Autrement dit :

$$x_{ij}^k \in \{0, 1\}; \quad i = 0, \dots, n; \quad j = 0, \dots, n; \quad k = 1, \dots, m \quad (1.7)$$

Le problème se modélise comme suit :

$$\min \sum_{i=1}^n \sum_{j=1}^n C_{ij} \sum_{k=1}^m x_{ij}^k \quad (1.8)$$

Sous les contraintes :

$$\sum_{i=1}^n \sum_{k=1}^m x_{ij}^k = 1; \quad j = 2, \dots, n \quad (1.9)$$

$$\sum_{j=1}^n \sum_{k=1}^m x_{ij}^k = 1; \quad i = 1, \dots, n \quad (1.10)$$

$$\sum_{i=1}^n x_{ip}^k - \sum_{j=1}^n x_{pj}^k = 0; \quad k = 1, \dots, m; \quad p = 1, \dots, n \quad (1.11)$$

$$\sum_{i=1}^n q_i \left( \sum_{j=1}^n x_{ij}^k \right) \leq Q_k; \quad k = 1, \dots, m \quad (1.12)$$

$$\sum_{i=1}^n s_i^k x_{ij}^k + \sum_{i=0}^n \sum_{j=0}^n t_{ij}^k x_{ij}^k \leq T_k; \quad k = 1, \dots, m \quad (1.13)$$

$$\sum_{j=1}^n x_{0j}^k \leq 1; \quad k = 1, \dots, m \quad (1.14)$$

$$\sum_{i=1}^n x_{i0}^k \leq 1; \quad k = 1, \dots, m \quad (1.15)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij}^k \leq |S| - 1; \quad \forall S \subset N; \quad 2 \leq |S| \leq n - 1 \quad (1.16)$$

La formule (1.8) représente la fonction objectif du VRP classique.

Les contraintes :

- (1.9) et (1.10) assurent que chaque client est visité exactement une fois et par un seul véhicule ;
- (1.11) assure que le véhicule quitte le client une fois qu'il l'a servi ;
- (1.12) garantit que la capacité du véhicule n'est pas dépassée ;
- (1.13) assure que la durée d'une tournée ne dépasse pas la durée maximale autorisée ;
- (1.14) et (1.15) garantissent le respect de la disponibilité de chaque véhicule : un véhicule quitte le dépôt et y retourne une seule fois ;
- (1.16) veille à l'élimination des sous-tours.

## Formulation 2

C'est une formulation proposée par Fisher and Jaikumar, en (1978) et (1981) [28], basée essentiellement sur deux catégories de variables binaires, une à trois indices et une autre à deux indices :

$$x_{ij}^k = \begin{cases} 1 & \text{si l'arc } (i, j) \text{ est parcouru par le } k^{\text{ème}} \text{ véhicule} \\ 0 & \text{sinon} \end{cases} \quad (1.17)$$

$$y_{ik} = \begin{cases} 1 & \text{si le } i^{\text{ème}} \text{ client est servi par le } k^{\text{ème}} \text{ véhicule} \\ 0 & \text{sinon} \end{cases} \quad (1.18)$$

Autrement dit :

$$y_{ik} \in \{0, 1\}; \quad i = 1, \dots, n; \quad k = 1, \dots, m \quad (1.19)$$

Le problème se modélise comme suit :

$$\min \sum_{i=1}^n \sum_{j=1}^n C_{ij} \sum_{k=1}^m x_{ij}^k \quad (1.20)$$

Sous les contraintes :

$$\sum_{k=1}^m y_{ik} = 1; \quad i = 1, \dots, n \quad (1.21)$$

$$\sum_{k=1}^m y_{0k} = m \quad (1.22)$$

$$\sum_{i=1}^n q_i y_{ik} \leq Q_k; \quad k = 1, \dots, m \quad (1.23)$$

$$\sum_{j=1}^n x_{ij}^k = \sum_{j=1}^n x_{ji}^k = y_{ik}; \quad k = 1, \dots, m; \quad i = 1, \dots, n \quad (1.24)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij}^k \leq |S| - 1; \quad \forall S \subseteq \{2, \dots, n\}; \quad k = 1, \dots, m \quad (1.25)$$

- $n$  : nombre de clients ;
- $m$  : nombre de véhicules ( $m \leq n$ ) ;
- $Q_k$  : capacité du  $k^{\text{ème}}$  véhicule ;
- $T_k$  : temps maximal de la tournée du  $k^{\text{ème}}$  véhicule ;
- $a_j^s$  : nombre d'arcs sortants du sommet  $j$  ;
- $a_j^e$  : nombre d'arcs entrants du sommet  $j$  ;
- $q_j$  : demande du  $j^{\text{ème}}$  client,  $q_0 = 0$  ;
- $s_{kj}$  : durée du service du  $j^{\text{ème}}$  client par le  $k^{\text{ème}}$  véhicule,  $s_{k0} = 0$  ;
- $t_{ij}$  : temps nécessaire pour passer du  $i^{\text{ème}}$  client au  $j^{\text{ème}}$  client,  $t_{kj} = +\infty$  ;
- $d_{ij}$  : distance séparant le  $i^{\text{ème}}$  client du  $j^{\text{ème}}$ ,  $d_{kj} = +\infty$  ;
- $C_{ij}$  : coût du déplacement du  $i^{\text{ème}}$  client au  $j^{\text{ème}}$  client.

La formule (1.20) est la même que la formule (1.8), elle représente la fonction objectif.

Les contraintes :

- La contrainte (1.21) assure que chaque client n'est servi qu'une seule fois par un seul véhicule.
- (1.22) assure que tous les véhicules se trouvent au dépôt.
- (1.23) garantit que la capacité de chacun des véhicules ne soit pas dépassée.
- (1.24) permet de lier les deux types de variables et assure que le véhicule quitte le client dès qu'il a été servi.
- (1.25) garantit l'élimination des sous-tours.

## 1.4 Problème de tournées de véhicule électrique avec contraintes de capacité

Le problème de tournées de véhicule électrique avec contraintes de capacité (CEVRP) rentre dans la catégorie des Green VRP, comme une extension du problème classique de routage de véhicule. Ce cadre a des enjeux à la fois opérationnels et environnementaux. En effet, l'objectif est d'optimiser les itinéraires d'une flotte de véhicules électriques à capacité limitée, permettant de réduire l'impact environnemental. Le CEVRP combine les contraintes classiques du CVRP et les contraintes spécifiques des véhicules électriques liées à la recharge de leur batterie. La formulation que nous abordons a été proposée par Erdogan et Miller-Hooks (2012)[49].

Le problème de tournées de véhicules électriques se formule de la manière suivante :

Soit  $G = (V, E)$  un graphe complet non orienté.  $V = I \cup \{v_0\} \cup F$ , avec  $v_0$  un dépôt unique,  $I = \{v_1, \dots, v_n\}$  est l'ensemble des clients.  $F = \{u_1, \dots, u_s\}$  est un ensemble de stations de rechargement. Nous admettons en plus que le dépôt permet le rechargement de la batterie. Toute station de rechargement a une capacité illimitée.

À chaque arête  $(v_i, v_j)$  appartenant à  $E$  (avec  $v_i, v_j \in V$  et  $i < j$ ) est associé un temps de parcours  $t_{ij}$ , un coût de consommation  $c_{ij}$  et une distance  $d_{ij}$ . Les temps de parcours sont donc supposés constants sur les arcs. En plus, il n'y a pas de limite sur le nombre de stations à lesquelles un véhicule peut s'arrêter dans son trajet. Si un véhicule s'arrête dans une station, il se charge à plein.

L'objectif de ce problème est de visiter chaque client exactement une fois en utilisant  $m$  tours au minimum, en minimisant la distance totale parcourue tout en respectant les contraintes de consommation.

## 1.5 Problèmes NP-Complets

une grande part des problèmes d'optimisation combinatoire appartiennent à la classe NP-Complète. elle représente l'ensemble des problèmes les plus difficiles de la classe NP. Un problème est dit NP-Complet si tout problème de la classe NP peut se réduire polynomialement à lui, c'est-à-dire s'il existe un algorithme polynomial capable de transformer toute instance  $x$  d'un problème  $p$  en une instance  $y$  d'un problème  $Q$ , telle que  $x$  est une instance valide si et seulement si  $y$  l'est également.

Cette propriété est très forte, car elle implique que si l'on parvient à trouver un algorithme polynomial pour un seul problème NP-Complet, alors on obtiendrait, par réduction, un algorithme polynomial pour tous les problèmes de la classe NP. Cela reviendrait à démontrer que  $P = NP$ , ce qui constitue l'une des plus grandes conjectures non résolues en informatique théorique.

Ce résultat repose sur le théorème de Cook[29], qui a établi l'existence d'un premier problème NP-Complet (le problème de la satisfiabilité booléenne, SAT), posant ainsi les bases de la théorie de la complexité des algorithmes.

## 1.6 Conclusion

Le VRP est un problème NP-difficile, ce qui implique que le CEVRP considéré comme une généralisation multi-contraintes du VRP, le dépasse en complexité et appartient également à la classe des problèmes NP-difficiles. Les classes de complexité aident à comprendre les limites de l'informatique théorique et guident le choix des méthodes de résolution (exactes vs approchées). Ces méthodes feront l'objet du prochain chapitre.

# Chapitre 2

## Différentes Approches de Résolution

### 2.1 Introduction

Le chapitre inclue les différentes méthodes de résolution des problèmes à grande difficulté, toujours en se concentrant sur l'objectif principal de notre travail : trouver la meilleure solution pour un CEVRP, qui est défini dans la classe NP-C. Nous allons passer en revue les différentes méthodes de résolution.

La première partie va définir les méthodes exactes tout en illustrant des exemples concrets et réels.

La deuxième partie va s'intéresser aux méthodes approchées, qui se divisent en trois branches importantes : les heuristiques, les méta-heuristiques, et enfin les méthodes hybrides, que nous allons développer tout au long de ce chapitre.

### 2.2 Méthodes exactes

#### 2.2.1 Définition des Méthodes exactes

les méthodes exactes sont une classe de méthodes qui permettent de résoudre les problèmes d'optimisation combinatoire. Elles assurent une solution optimale suite à un parcours exhaustif du domaine de réalisabilité du problème.

## 2.2.2 Quelques approches exactes

### 2.2.2.1 Programmation linéaire

L'objectif de la programmation linéaire est de maximiser ou de minimiser une fonction objectif tout en satisfaisant un ensemble de contraintes portant sur des variables réelles, entières ou naturelles [32]. Elle se formule de la manière suivante :

$$(P) = \begin{cases} \text{Min } Z = C \cdot x \\ \text{s.c. } A \cdot x = b \\ x \geq 0 \end{cases} \quad (2.1)$$

où :

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

La résolution de ce type de problème peut se faire graphiquement, en suivant la méthode des polyèdres initiée par J. Fourier. Elle peut également être abordée à travers la procédure d'élimination de Fourier-Kuhn. Cependant, c'est la méthode du simplexe, introduite par G. B. Dantzig en 1947[32], qui s'est imposée comme référence.

Le principe du simplexe consiste à parcourir les sommets d'un polyèdre, de sommet en sommet, de manière à ce que la valeur de la fonction objectif augmente strictement à chaque étape. Si cette fonction est bornée et que le nombre de sommets est fini, alors l'optimum est atteint en un nombre fini d'itérations [31].

La méthode du simplexe reste aujourd'hui l'une des plus reconnues. En effet, plusieurs autres méthodes de résolution nécessitent son intervention pour parvenir à une solution exacte en pratique. Elle a également permis le développement de codes de calcul performants, capables de résoudre un grand nombre de problèmes réels.

Le seul inconvénient du simplexe est qu'il ne garantit pas une complexité polynomiale dans le pire des cas. Toutefois, Steve Smale a démontré qu'en moyenne, et pour un nombre fixe de contraintes, son comportement reste polynomial.

### 2.2.2.2 Programmation dynamique

La résolution d'un problème d'optimisation à l'aide de la programmation dynamique repose sur deux étapes essentielles :

- décomposition du problème en une suite de sous-problèmes,
- établissement d'une relation de récurrence entre les solutions optimales de ces sous-problèmes.

Bien que cette méthode ait été connue de manière abstraite dans les milieux scientifiques, c'est R. Bellman [31] qui a posé pour la première fois le principe d'optimalité dans une formulation simple : dans une suite de décisions optimales, quelle que soit la première décision prise, les décisions suivantes doivent constituer elles aussi une politique optimale en tenant compte de cette première décision.

Ci-dessous un extrait tiré de l'autobiographie de Bellman [31] illustrant cette pensée :

*« My first task was to find a name for multistage decision processes. In the first place I was interested in planning, in decision making, but planning is not a good word for various reasons. I decided therefore to use the word "programming". I wanted to get across the idea that this was dynamic, this was multistage, this was time-varying. I thought, let's kill two birds with one stone. Let's take a word that has an absolutely precise meaning, namely dynamic, in the classical physical sense. »*

La programmation dynamique s'applique à travers deux approches principales. La première, dite procédure avant, consiste à découper un problème en  $n + 1$  périodes avec un ensemble d'états possibles. En s'appuyant sur l'hypothèse de monotonie, cette approche permet de calculer les politiques optimales et de résoudre le problème progressivement.

La seconde, appelée procédure arrière, commence par l'état final et remonte successivement vers l'état initial en suivant la même logique de calculs. De nombreux domaines tirent avantage de cette méthode, notamment la gestion des stocks, les problèmes d'allocation de ressources, le sac à dos ou encore le problème du plus court chemin de type bottleneck.

La principale difficulté réside dans l'identification des cas où cette méthode est applicable. Déterminer si un problème peut être formulé de manière séquentielle et si le principe d'optimalité s'applique ne repose sur aucune règle générale. Cette reconnaissance dépend essentiellement de l'expérience et de la réflexion du modélisateur.

### 2.2.2.3 Méthode Branch and Bound

La méthode Branch and Bound a été introduite pour la première fois par Land et Doig [33]. Elle s'appuie sur la représentation d'un arbre divisé récursivement en branches, chacune correspondant à un sous-problème. Tout au long du processus, une stratégie d'élimination est appliquée : lorsqu'une solution réalisable a déjà été trouvée, et qu'une nouvelle solution candidate issue d'une branche offre une meilleure valeur pour la fonction objectif, cette dernière est adoptée. Le sous-problème correspondant est alors subdivisé à son tour. Si, au contraire, le sous-problème ne peut pas améliorer la solution actuelle, il est simplement abandonné.

Lorsque plus aucune itération n'est possible, la meilleure solution trouvée est retenue comme solution optimale du problème initial. La méthode repose sur trois éléments fondamentaux :

- une stratégie de recherche,
- une stratégie de branchement,
- et une stratégie d'élagage, qui permet de décider si un sous-problème doit être écarté ou non.

Ces trois composantes structurent le processus en deux grandes phases, la recherche de solutions et la vérification de optimalité. La complexité de la méthode Branch and Bound dépend de deux facteurs :

- le facteur de branchement, c'est-à-dire le nombre maximal de sous-problèmes générés à partir d'un nœud,
- la profondeur de la recherche, soit la longueur maximale du chemin entre la racine et une feuille de l'arbre.

La difficulté réside dans le fait que les performances de l'algorithme sont souvent très sensibles aux conditions initiales. Pour améliorer l'efficacité, des extensions basées sur des heuristiques ont été proposées, permettant de générer de meilleures solutions initiales ou d'optimiser les sous-problèmes. Dans certains cas, des algorithmes hybrides sont également utilisés pour résoudre les sous-problèmes efficacement.

### 2.2.3 Avantages et Inconvénients des Méthodes exactes

les méthodes exactes représentent un avantage majeur : elles garantissent une solution optimale et offrent des résultats fiables et efficaces pour des instances petites à moyenne de taille. Bien que les méthodes de recherche exacte garantissent des solutions optimales, elles restent principalement adaptées

aux petites instances en raison de leur complexité algorithmique. Elles exigent une grande capacité temporo-spataile, ce qui les rend coûteuses en ressources, et leur temps de résolution peut s'étendre sur de très longues périodes. De plus, ces méthodes manquent de flexibilité et d'adaptabilité, ce qui limite leur efficacité dans des environnements dynamiques ou à contraintes évolutives.

## 2.3 Approches Heuristique

En général, une méthode est dite heuristique si elle conduit à une solution réalisable, mais pas nécessairement optimale. C'est-à-dire une solution estimée comme « bonne », obtenue dans un temps raisonnable pour des problèmes difficiles.

### 2.3.1 Définitions

Les méthodes heuristiques sont des approches qui visent à fournir des solutions réalisables, sans assurance d'optimalité. Elles sont souvent utilisées lorsque les méthodes exactes deviennent trop coûteuses en temps de calcul. Herbert Simon, prix Nobel d'économie en 1978 [23] est considéré comme le père des heuristiques.

Les heuristiques sont largement utilisées dans la recherche opérationnelle, notamment dans des cas comme le problème du plus court chemin, mais aussi en psychologie dans des situations de prise de décision rapide sous incertitude, ainsi que dans de nombreux autres domaines.

Les principales catégories des méthodes heuristiques sont :

- Heuristiques de construction : ces méthodes génèrent des solutions étape par étape à partir de zéro. Un exemple courant est l'algorithme du plus proche voisin, qui consiste à choisir, à chaque étape, le client le plus proche à desservir.
- Heuristiques d'amélioration locale : ce type de méthode part d'une solution initiale et cherche à l'améliorer en appliquant des mouvements locaux. Par exemple, on peut supprimer des arêtes et reconnecter des segments, ou déplacer un client d'une tournée à une autre afin de modifier la solution et améliorer son coût.
- Heuristiques de type "ruine et recréation" : ces méthodes détruisent une partie de la solution initiale, puis la reconstruisent partiellement, dans l'objectif d'explorer de nouvelles configurations plus performantes.

Le principal inconvénient des heuristiques est qu'elles sont conçues pour

un type particulier de problème. En effet, elles reposent sur des méthodes empiriques qui ne s'adaptent pas facilement à d'autres problèmes ou variantes. Chaque nouveau cas nécessite souvent une reprogrammation, car ces méthodes dépendent fortement de la forme des données d'entrée, de leur nombre, ainsi que du type et du nombre de contraintes à respecter.

## **2.3.2 Méta-Heuristiques**

Face au coût de recherche prohibitif des méthodes exactes et à la spécificité des heuristiques, trouver une solution moins exigeante était primordial. Pour cela, en 1980, les études sur les méta-heuristiques ont véritablement commencé.

Le mot heuristique peut sembler familier, mais l'ajout du préfixe méta change toute la perspective. Méta, qui signifie « au-delà », indique que l'on passe à un niveau supérieur dans la résolution des problèmes complexes. Ces problèmes se divisent généralement en deux types : les problèmes discrets, dont les plus connus sont les problèmes NP-difficiles comme le VRP, et les problèmes continus, tels que la recherche de paramètres dans des modèles numériques pour mieux approcher des données réelles.

Ces méthodes se distinguent par leur nature stochastique, souvent d'origine discrète, et s'inspirent généralement de phénomènes naturels : le recuit simulé, basé sur un processus métallurgique ; les algorithmes évolutionnaires et génétiques, inspirés des principes de l'évolution biologique ; les algorithmes basés sur l'intelligence d'essaim, comme l'algorithme d'optimisation par essaim de particules ou l'algorithme de colonies de fourmis, qui imitent le comportement social de certaines espèces vivant en groupe.

L'un des grands avantages des méta-heuristiques est leur capacité à éviter les optima locaux. Elles peuvent être utilisées seules ou combinées à d'autres techniques dans le cadre d'hybridations.

## **2.3.3 quelques approches heuristiques**

### **2.3.3.1 Le recuit simulé**

Proposé en 1982 par le trio de chercheurs d'IBM (S. Kirkpatrick, C. D. Gellat et M. P. Vecchi) et publié en 1983, le recuit simulé est une méthode d'optimisation inspirée d'un procédé métallurgique. Elle imite le processus

physique du recuit, où un métal est chauffé pour augmenter son énergie puis refroidi lentement pour atteindre une configuration stable à faible énergie.

Le paramètre de contrôle principal est la température. Un refroidissement rapide piège le matériau dans une configuration désordonnée (minimum local), tandis qu'un refroidissement lent favorise l'atteinte d'une structure ordonnée (minimum global).

Les paramètres fondamentaux de cette méthode sont :

- La température initiale,
- La longueur des chaînes de Markov (nombre de solutions explorées par palier de température),
- La loi de décroissance de la température,
- La condition d'arrêt.

Le critère d'arrêt peut être un nombre maximal d'itérations ou la stagnation de la solution pendant plusieurs paliers.

Le principal inconvénient de cette méthode reste le temps de calcul important. Pour y remédier, deux variantes ont vu le jour :

### **Recuit simulé parallèle**

Les travaux de Aarts et al. (1986) proposent d'exécuter plusieurs chaînes de Markov en parallèle à l'aide de plusieurs processus. Une autre approche consiste à paralléliser les calculs au sein d'une même chaîne. L'architecture « Fermier / Travailleurs » est également utilisée pour coordonner les traitements.

### **Recuit simulé distribué**

Cette variante divise l'espace de recherche en sous-espaces, chacun étant affecté à un agent. L'interaction entre ces agents exécutant indépendamment un recuit simulé permet de converger vers une bonne solution globale.

### pseudo-code du recuit simulé

**Input:** Une solution initiale  $x$ , température initiale  $T_0$ , facteur de décroissance  $\alpha$ , condition d'arrêt

**Output:** Une solution approchée du minimum global

```
1 begin
2    $x_{\text{courante}} \leftarrow x$ 
3    $x_{\text{meilleure}} \leftarrow x$ 
4    $T \leftarrow T_0$ 
5   while la condition d'arrêt n'est pas satisfaite do
6     Générer une solution voisine  $x'$  de  $x_{\text{courante}}$ 
7     Calculer  $\Delta = f(x') - f(x_{\text{courante}})$ 
8     if  $\Delta < 0$  then
9       Accepter  $x'$  comme nouvelle solution courante
10       $x_{\text{courante}} \leftarrow x'$ 
11      if  $f(x') < f(x_{\text{meilleure}})$  then
12        Mettre à jour la meilleure solution trouvée
13         $x_{\text{meilleure}} \leftarrow x'$ 
14      end
15    else
16      Accepter  $x'$  avec une probabilité  $p = \exp(-\Delta/T)$ 
17    end
18    Réduire la température :  $T \leftarrow \alpha \cdot T$ 
19  end
20  return  $x_{\text{meilleure}}$ 
21 end
```

**Algorithm 1:** Algorithme du recuit simulé

#### 2.3.3.2 Recherche tabou

La recherche tabou tire son nom du fait que certaines solutions doivent être rendues *taboues*, c'est-à-dire interdites, afin d'éviter de revenir sur des optimums locaux déjà explorés. À l'instar du recuit simulé, cette méthode consiste à partir d'une solution  $x_n$  à l'étape  $n$ , puis à calculer la meilleure solution  $x^*$  dans un sous-voisinage  $V^*$  de  $V(x_n)$ . Le processus se poursuit ainsi, permettant une exploration plus large de l'espace de recherche dans le but d'atteindre l'optimum global.

Afin d'éviter les convergences répétées vers des minima locaux, la méthode de recherche tabou utilise la notion de mémoire. Pour diversifier les choix et empêcher les retours vers des solutions déjà explorées, des listes de tabou sont utilisées. Ces listes conservent en mémoire les dernières solutions rencontrées,

ou certaines de leurs caractéristiques, et excluent celles-ci si elles se présentent à nouveau ou si des solutions similaires apparaissent. Ces listes jouent donc un rôle important dans la diversification de la recherche.

La taille de la liste de tabou est limitée, ce qui empêche l'enregistrement de toutes les solutions rencontrées. C'est pourquoi la liste de tabou fonctionne selon un principe FIFO (First In, First Out) : l'élément le plus ancien laisse sa place au plus récent.

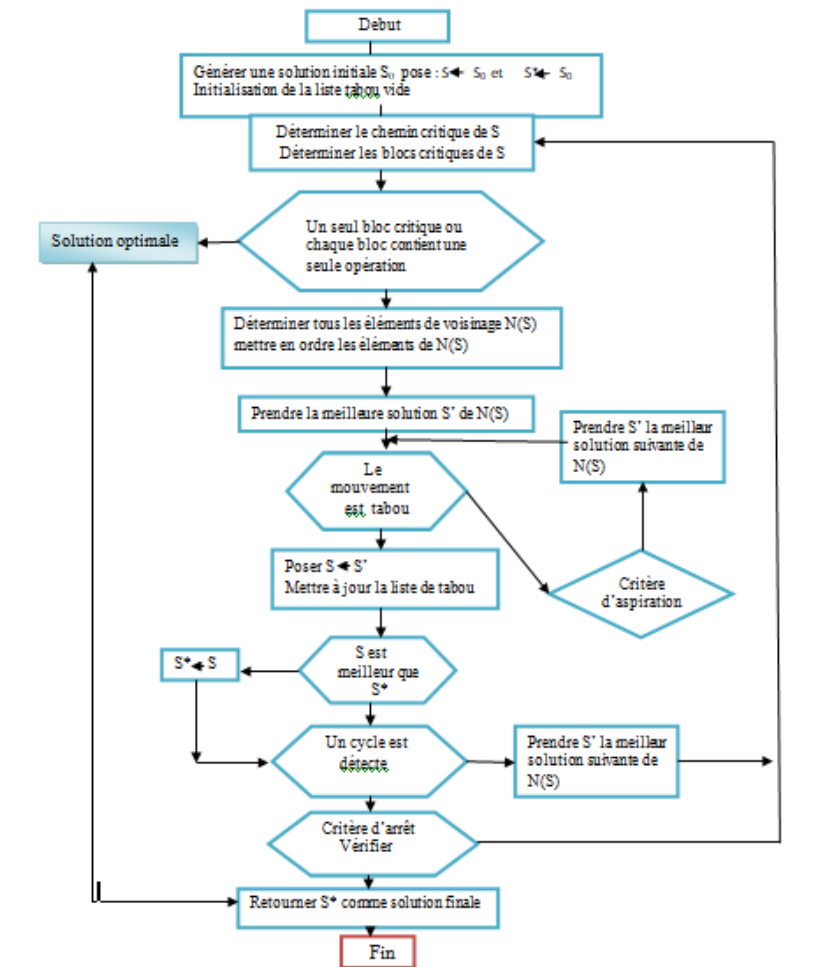


FIGURE 2.1 – Méthode tabou

### 2.3.3.3 Colonie de fourmis

L'optimisation par colonie de fourmis est une méta-heuristique inspirée du comportement de recherche de nourriture observé chez les fourmis. Ce comportement repose principalement sur une forme de communication indi-

recte entre les individus : lorsqu'elles cherchent de la nourriture, les fourmis commencent par explorer de manière aléatoire la zone autour de leur nid. Une fois qu'une fourmi découvre une source de nourriture, elle en évalue la quantité et la qualité, puis en rapporte une partie au nid. Durant le trajet de retour, elle dépose sur le sol une traînée de phéromones chimiques. Cette substance volatile permet aux autres fourmis de repérer les chemins empruntés et d'identifier ceux qui mènent aux meilleures sources, favorisant ainsi les itinéraires les plus courts.

L'algorithme de colonie de fourmis a été introduit pour la première fois par Marco Dorigo [32] au début des années 1990, puis formalisé en 1999 avec ses collègues. Le principe de base de cet algorithme repose sur une population de fourmis virtuelles qui construisent des solutions au problème à résoudre. Plus une solution est bonne, plus la quantité de phéromones qui y est déposée est élevée, ce qui augmente la probabilité que d'autres fourmis suivent ce chemin lors des itérations suivantes.

L'algorithme fonctionne de la manière suivante : au départ, toutes les arêtes du graphe représentant le problème sont initialisées avec une quantité nulle (ou minimale) de phéromones. Ensuite, chaque fourmi virtuelle construit une solution pas à pas, en choisissant ses déplacements de manière probabiliste selon les quantités de phéromones et des heuristiques locales. Une fois toutes les solutions construites, les meilleures sont renforcées par l'ajout d'une quantité importante de phéromones sur les arcs qu'elles ont empruntés.

L'algorithme comprend aussi un mécanisme d'évaporation des phéromones. IL consiste à diminuer intensité du phéromone au fil du temps. Ce processus permet à l'algorithme de « se souvenir » moins fortement des anciennes solutions, favorisant ainsi l'exploration de nouveaux chemins et évitant un enfermement prématuré dans des solutions sous-optimales.

Ces étapes sont répétées jusqu'à atteindre un critère d'arrêt, généralement défini par un nombre fixe d'itérations ou par la stabilité des solutions obtenues.

#### **2.3.3.4 Algorithmes génétiques**

Les algorithmes génétiques font partie de la catégorie des algorithmes évolutifs, inspirés de la théorie de l'évolution de Darwin[32]. J. Holland a conçu ces algorithmes pour démontrer comment les espèces évoluent au fil du temps. Par la suite, D. Jongh les a appliqués à l'optimisation de fonctions à variables réelles. La particularité de cette méthode réside dans le fait qu'elle traite et évalue une population de solutions, contrairement à la recherche

taboue ou au recuit simulé, qui ne traitent qu'une seule solution à la fois.

Le vocabulaire utilisé pour définir les différents éléments de l'algorithme est celui de la biologie : gène, chromosome, individu, population et génération. Un gène représente une partie de la solution ; dans la plupart des cas, il s'agit d'un bit, d'un entier, d'un réel ou même d'un caractère. Le chromosome est un ensemble de gènes, ce qui définit le contenu d'une solution. L'individu fait référence à un ou plusieurs chromosomes et représente une solution au problème. Un ensemble d'individus est appelé une population. Enfin, une génération correspond au résultat d'un cycle complet d'évolution, c'est-à-dire la nouvelle population générée après avoir subi les différentes étapes de l'algorithme, telles que la sélection, le croisement et la mutation. Ces étapes constituent ensemble les opérations génétiques, qui, initialement appliquées à une population aléatoire, ont pour but de produire des solutions de plus en plus optimisées au fil des générations successives.

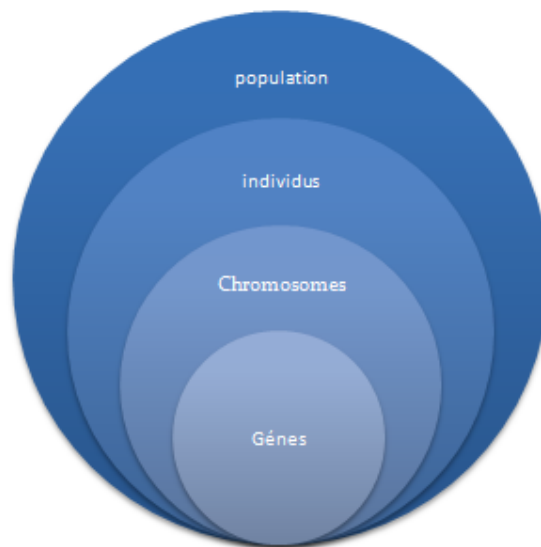


FIGURE 2.2 – Structure génétique des solutions dans un algorithme génétique

Les algorithmes génétiques sont composés des opérations suivantes : codage des individus (représentation), fonction de fitness (évaluation), sélection, croisement et mutation.

### **Codage des individus**

Avant de soumettre les données de la solution initiale à l'algorithme, il est nécessaire de traduire l'ensemble de cette population dans une forme que l'algorithme puisse comprendre. Cette forme est appelée individu. Chaque

individu, constitué d'un ou plusieurs chromosomes, est composé de gènes qui représentent un paramètre à coder. Il existe plusieurs types de codage :

- Codage binaire : chaque solution est représentée par une suite de 0 ou 1, qui peuvent faire référence à une décision, un choix ou une partie de la solution. Ce codage est utilisé, par exemple, pour les problèmes de MAX SAT ou les problèmes du sac à dos.
- Codage entier : la solution est codée sous forme de nombres entiers. Il est particulièrement utile pour les problèmes d'ordonnancement ou le *voyageur de commerce*.
- Codage réel : ce codage est adapté aux solutions continues et est très utilisé dans les problèmes d'optimisation numérique.
- Codage combiné (hybride) : parfois, un seul type de codage ne suffit pas. Dans ce cas, deux types peuvent être combinés, par exemple binaire et réel.

### **Fonction de fitness (évaluation)**

Les solutions initiales passent par l'étape d'évaluation pour classer les individus, choisir les meilleurs pour la sélection et, bien entendu, éliminer les plus faibles. Ce processus est réalisé à l'aide d'une fonction appelée *fonction de fitness*, qui renvoie la performance de chaque individu.

### **Sélection**

La sélection est une étape cruciale dans l'algorithme. Elle choisit les individus qui participeront à la reproduction. En utilisant les scores de fitness, elle désigne les meilleurs individus pour favoriser les générations suivantes et maintenir la diversité afin d'éviter l'optimum local.

Il existe plusieurs approches de sélection, parmi elles :

- Sélection par tournoi déterministe : il s'agit de choisir aléatoirement  $k$  individus, puis de sélectionner celui ayant la meilleure performance. Ce processus est répété jusqu'à obtenir le nombre de parents désiré.
- Sélection par rang : cette méthode consiste à classer les individus selon leurs performances de manière décroissante, puis à attribuer un rang à chaque individu (par exemple, le meilleur reçoit le rang 1, et ainsi de suite). Ensuite, la probabilité d'être sélectionné est calculée en fonction du rang, et non du fitness absolu.

### **Croisement**

Le croisement est inspiré de la reproduction sexuée en biologie, où les gènes de deux parents sont combinés pour créer un ou plusieurs enfants. Dans ce principe, si  $x$  et  $y$  sont deux solutions sélectionnées, un opérateur de croisement génère une ou plusieurs nouvelles solutions  $x'$  et  $y'$  qui héritent des

caractéristiques des parents.

Le taux de croisement est un paramètre important qui définit la probabilité avec laquelle deux parents sont combinés. Par exemple, si ce taux est de 0,8, cela signifie que 80% des paires de parents génèrent des enfants par croisement, tandis que 20% sont copiés sans modification. Ce facteur dépend de la représentation des individus et de la diversité des solutions.

On distingue plusieurs types de croisements.

le croisement à un point : Ce type consiste à choisir un point de coupure aléatoire sur le chromosome et à échanger les segments entre les deux parents.



FIGURE 2.3 – Un point

Croisement à deux points : choisir deux points de coupure puis échanger les segments entre ces deux points.

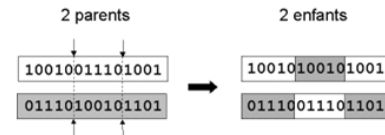


FIGURE 2.4 – Deux points

Croisement uniforme : choisir aléatoirement avec une certaine probabilité souvent 50% des parents de source.

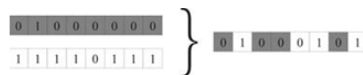


FIGURE 2.5 – Uniforme

### Mutation

La mutation est une étape où un individu subit de simples modifications au niveau d'un ou plusieurs gènes. L'objectif de cette opération est d'échapper aux optima locaux, tout en explorant de nouvelles solutions et en évitant que les individus deviennent trop similaires. Le mécanisme de mutation

dépend du type de représentation des chromosomes. Pour un codage binaire, on utilise le bit flip : il s'agit de changer un bit de 0 à 1 ou de 1 à 0. Pour les codages réels, on applique une perturbation aléatoire, c'est-à-dire ajouter un bruit à une ou plusieurs valeurs. Pour les problèmes de permutation ou d'ordonnement, on utilise généralement la mutation par swap, qui échange deux positions dans la séquence.

### **Évaluation**

Les individus générés par croisement et mutation sont ensuite évalués pour mesurer leur fitness, vu que les changements apportés entraînent de nouveaux scores. Ces scores détermineront les individus qui seront retenus pour la reproduction.

### **Remplacement**

À cette phase, le nombre de solutions peut augmenter et dépasser la limite fixée. Il devient donc nécessaire de choisir entre les parents et les enfants pour constituer la nouvelle population. Voici quelques méthodes classiques de remplacement :

- Remplacement générationnel : toute la population est remplacée par les enfants.
- Remplacement stable : seuls quelques individus sont remplacés à chaque génération.
- Remplacement par élitisme : les meilleurs individus sont conservés intacts (copiés directement dans la nouvelle génération).

### **Critère d'arrêt**

À la fin de chaque génération, avant de passer à la suivante, il est nécessaire de vérifier un critère d'arrêt. Voici les trois types de conditions généralement utilisées :

- Une limite supérieure de générations est atteinte.
- Une fitness suffisante est atteinte.
- La probabilité d'amélioration significative devient très faible, voire nulle — autrement dit, il y a peu ou pas de chance de trouver de meilleures solutions.

Les deux premiers critères nécessitent une bonne connaissance du problème afin de fixer des limites pertinentes. Le troisième, plus flexible, s'adapte dynamiquement à l'évolution de l'algorithme. Dans ce sens, on distingue deux types de critères adaptatifs :

- **Critère génotypique** : l'algorithme s'arrête lorsqu'un grand nombre de gènes deviennent identiques selon un seuil, par exemple 90 %.

- **Critère phénotypique** : il consiste à observer si la qualité des solutions continue de s'améliorer. Si la fitness reste stable sur plusieurs générations, l'algorithme s'arrête automatiquement.

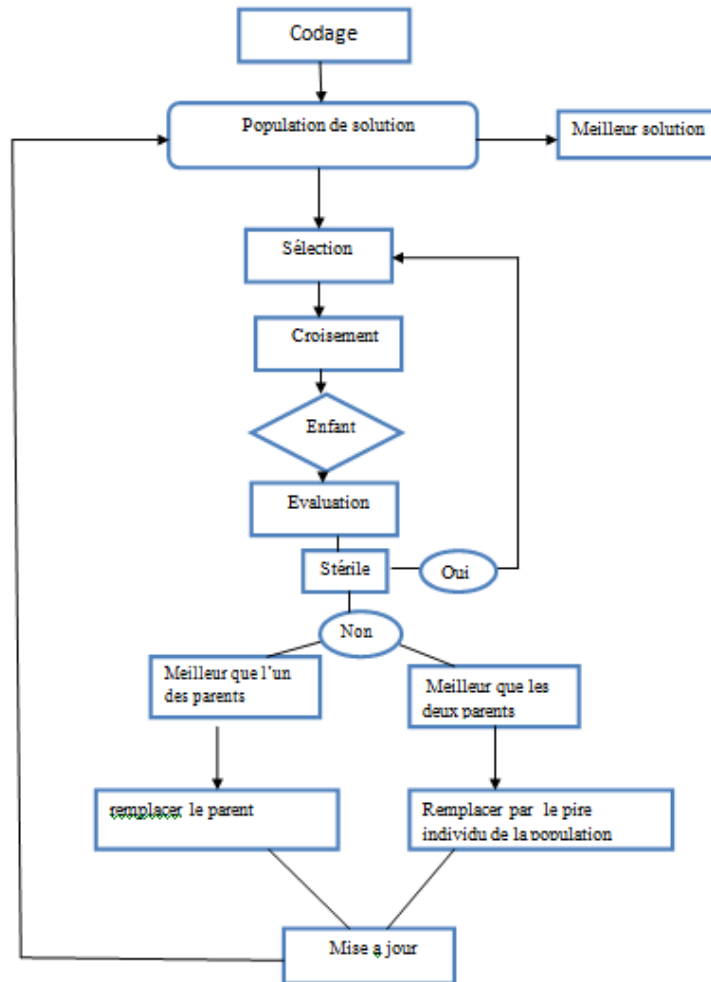


FIGURE 2.6 – Uniforme

## 2.4 Approches Hybrides

Les méthodes hybrides sont des approches algorithmiques qui résultent de la combinaison de deux ou plusieurs techniques. L'objectif principal est de tirer profit des avantages de chaque méthode tout en compensant leurs limites respectives. En d'autres termes, une méthode peut pallier les faiblesses d'une autre lorsqu'elles sont utilisées ensemble, ce qui améliore l'efficacité globale

de la résolution du problème. Les méthodes hybrides visent à approfondir l'exploration autour des meilleures solutions connues tout en explorant de nouvelles zones de l'espace de solution.

Il existe plusieurs formes d'hybridation, parmi lesquelles on peut citer :

#### **2.4.1 Hybridation exacte-exacte :**

Cette forme d'hybridation consiste à combiner plusieurs méthodes exactes, souvent dans le but de traiter différentes parties du problème ou de renforcer la performance générale. Par exemple, dans le problème de planification des horaires des trains, la programmation linéaire en nombres entiers (PLNE) est utilisée pour optimiser les trajets, tandis que la programmation par contraintes intervient pour gérer les conflits de voies, les intervalles de sécurité ou encore les contraintes temporelles. D'autres applications combinant ces deux méthodes incluent l'ordonnancement de tâches, la gestion des stocks ou encore la conception de réseaux.

#### **2.4.2 Hybridation exacte-heuristique :**

Ce type d'hybridation permet de bénéficier à la fois de la rapidité et de la flexibilité des méthodes heuristiques, ainsi que de la rigueur et de la garantie d'optimalité des méthodes exactes. Une heuristique peut ainsi fournir rapidement une bonne solution initiale, qui sera ensuite améliorée ou validée par une méthode exacte. Inversement, une méthode exacte peut servir à résoudre des sous-problèmes complexes dans une approche heuristique globale.

#### **2.4.3 Hybridation heuristique-heuristique**

Cette hybridation assemble plusieurs heuristiques ou méta-heuristiques distinctes pour bénéficier de leurs qualités respectives. Le but est souvent de renforcer soit l'intensification, soit la diversification de la recherche. Par exemple, une heuristique gloutonne peut être utilisée pour développer rapidement une solution initiale, qui sera ensuite affinée par une approche plus exploratoire comme la recherche tabou ou un algorithme génétique.

**Remarque 1** *Le monde a connu une avancée technologique majeure, ce qui a entraîné une évolution significative des méthodes de résolution des problèmes d'optimisation combinatoire. Cette évolution s'est traduite notamment par l'intégration de certains processus, tels que les réseaux de neurones, dans les approches mentionnées précédemment.*

## 2.5 conclusion

En conclusion, un ensemble de méthodes de résolution de problèmes d'optimisation combinatoire a été présenté, ainsi que quelques hybridations. Il serait intéressant de développer une de ces approches pour un problème d'optimisation combinatoire combiné à un processus d'intelligence artificielle, tel que les réseaux de neurones, qui feront l'objet du chapitre suivant.

## Chapitre 3

# État de l'art des Réseaux de Neurons Artificiels

### 3.1 Introduction

Les réseaux de neurones artificiels (RNA) sont un outil fondamental de l'apprentissage automatique, qui est un sous-domaine de l'intelligence artificielle (IA). L'IA regroupe toutes les techniques visant à simuler l'intelligence humaine. Les RNA sont inspirés du fonctionnement du cerveau humain.

La première partie de ce chapitre est consacrée à la définition des réseaux de neurones biologiques ainsi qu'artificiels, suivie de leurs applications dans divers domaines.

la deuxième partie vise à présenter une vue d'ensemble des principaux types de RNA ainsi que leurs architectures. Enfin, les défis liés à leur utilisation seront abordés et expliqués.

### 3.2 Historique et évolution des Réseaux de Neurons

L'histoire des réseaux de neurones artificiels (RNA) commence au milieu du 20<sup>e</sup> siècle.

En 1943, McCulloch et Pitts [36] introduisent le concept de neurones formels, imitant les neurones biologiques capables de mémoriser des fonctions booléennes simples.

1949 Donald Hebb publie 'the organization of behavior' où il énonce les règles de Hebb posant ainsi les bases d'apprentissages synaptiques.

Frank Rosenblatt a introduit en 1957 [36] Le perceptron, le premier modèle

capable d'apprentissage automatique.

En 1960 Widrow et Hoff développent ADALIN, employé pour le traitement des signaux électriques.

Entre 1969-1980 une baisse des financements et un désintérêt pour les RNA dû aux recherches de Minsky et Papert qui ont montré l'incapacité de ces réseaux à résoudre des problèmes non linéaires.

C'est en 1982 que John Hopfield[48] propose les réseaux de Hopfield adaptés à la mémoire associative.

En 1985, Rumelhart, Hinton et Williams redonnent vie aux RNA en introduisant la rétropropagation du gradient. C'est un algorithme d'apprentissage adapté aux réseaux de neurones multicouches.

### **3.3 Réseaux de Neurones**

Les réseaux de neurones artificiels (RNA) sont des modèles informatiques qui imitent la façon dont les neurones d'un cerveau traitent l'information. Les réseaux de neurones visent à reproduire des propriétés des mémoires biologiques, notamment : le traitement de plusieurs informations simultanément, l'apprentissage adaptatif et la mémoire distribuée.

#### **3.3.1 Neurones Biologique**

Le système nerveux est composé d'environ cent (100) milliards de cellules nerveuses appelées aussi neurones biologiques. Ces cellules sont responsables de transmettre, traiter et stocker les informations. Un neurone est composé de : le soma, les dendrites, l'axone et les synapses.

Le soma : c'est le centre du neurone, qui intègre les signaux reçus.

Les dendrites : des branches qui reçoivent des signaux des autres neurones.

L'axone : une longue fibre qui transmet le signal à d'autres neurones.

Les synapses : des points à l'extrémité de l'axone qui permettent aux messages de passer vers les dendrites des autres cellules.

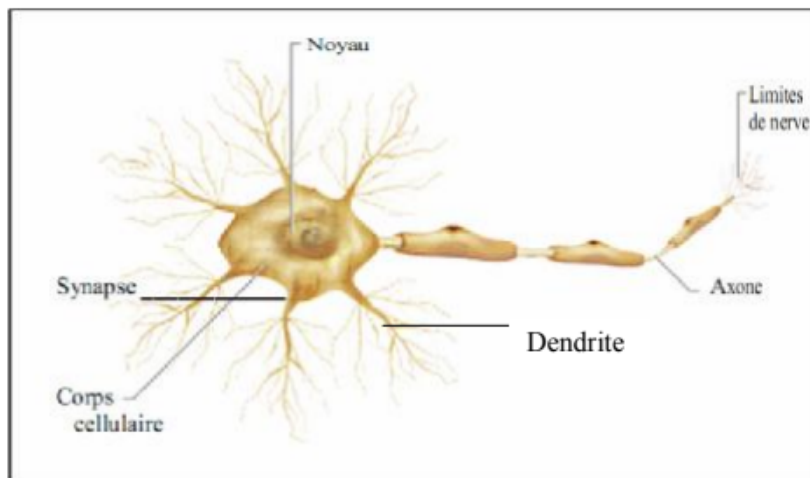


FIGURE 3.1 – schéma du neurone biologique

### 3.3.2 Neurone Formel

Un neurone formel est une imitation du neurone biologique. Il représente l'unité de base des réseaux de neurones. C'est un modèle mathématique conçu pour traiter des informations à l'aide de techniques d'apprentissage automatique. En effet, il reçoit des signaux, les traite et les traduit en réponse. Un neurone est composé de : entrées, poids synaptiques, fonction d'activation et sortie.

Entrées :  $(x_1, x_2, \dots, x_n)$  représentent les signaux reçus.

Poids synaptiques :  $(w_1, w_2, \dots, w_n)$  — chaque entrée est associée à un poids qui modélise l'intensité de la connexion synaptique.

Fonction d'activation : elle transforme la somme pondérée des entrées en une sortie.

La sortie du neurone :  $(y)$  réponse produite

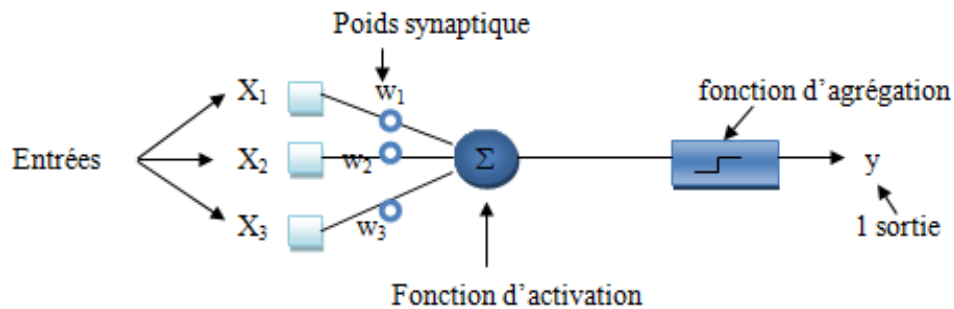


FIGURE 3.2 – schéma du neurone formel

Pour faire la liaison avec les neurones artificiels, les dendrites représentent les entrées, le soma représente la fonction de sommation et d'activation, les synapses sont les points synaptiques dans le réseau artificiel et l'axone est la sortie.

### 3.3.3 Apprentissage Automatique

L'apprentissage automatique consiste à analyser automatiquement un ensemble de données à partir d'une base de données préalablement analysée. L'apprentissage automatique se divise en trois catégories principales : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement [40].

#### 3.3.3.1 Apprentissage Supervisé

il consiste à apprendre à partir d'exemples déjà corrigés, où chaque exemple d'entraînement est associé à une réponse correcte. Le but est de prédire une sortie à partir des entrées. Par exemple : la prédiction de prix ou la reconnaissance d'images.

#### 3.3.3.2 Apprentissage non Supervisé

il s'agit d'un modèle dans lequel il n'existe pas de réponse connue à l'avance. L'algorithme explore les données pour découvrir des similarités ou des motifs cachés. Il est utilisé notamment dans le clustering et la réduction de dimension.

### 3.3.3.3 L'apprentissage par Renforcement

c'est un type d'apprentissage utilisé dans les cas où les données ne sont pas disponibles à l'avance. Le modèle apprend à travers des expériences interactives avec un environnement de décision. L'objectif est de maximiser une récompense au fil du temps.

## 3.4 Application des Réseaux de Neurones

Les réseaux neuronaux artificiels (RNA) sont des outils puissants, capables de résoudre divers défis dans de nombreux domaines. Ils sont employés dans le contrôle industriel, les prédictions médicales avancées, ainsi que l'amélioration des processus pour optimiser la qualité et la performance.

Le traitement du langage naturel (NLP) est un domaine de l'intelligence artificielle qui permet à la machine de comprendre et de manipuler le langage humain. Les machines peuvent ainsi effectuer des traductions, corriger des fautes de grammaire, résumer automatiquement des textes ou encore assurer une assistance vocale (ex. : Siri, Alexa). Parmi les techniques utilisées, on trouve les architectures Transformers, les réseaux de neurones et l'apprentissage automatique.

Dans l'industrie, les RNA sont utilisés pour l'analyse d'images et de vidéos, notamment afin de détecter des défauts et d'évaluer la qualité des produits. Ces derniers sont ensuite classés comme conformes ou non conformes selon des critères appris à partir de données d'apprentissage. Ils sont également exploités pour la maintenance prédictive : grâce à l'analyse de motifs inhabituels dans les signaux mesurés, les RNA permettent d'anticiper les pannes avant qu'elles ne se produisent.

Les réseaux de neurones sont aussi utilisés pour effectuer des prévisions dans le domaine des énergies renouvelables, facilitant ainsi la production et la gestion de l'électricité. Par exemple, dans le domaine de l'énergie solaire, les RNA permettent d'estimer la performance des panneaux solaires en analysant des données telles que l'ensoleillement, la température ou encore les conditions météorologiques, afin de prédire la quantité d'électricité produite.

Dans le domaine du marketing, les RNA permettent de prévoir les changements du marché et d'en évaluer l'impact. Ils analysent le com-

portement des clients, les classent en différentes catégories et estiment ainsi les ventes de chaque produit. Par ailleurs, les campagnes marketing sont désormais automatisées grâce aux RNA : les entreprises peuvent envoyer des messages personnalisés, au bon moment et à la bonne personne, en se basant sur l'activité de recherche et d'interaction de l'utilisateur.

Les RNA jouent également un rôle majeur dans le domaine de la santé. Ils peuvent analyser les données des électrocardiogrammes pour aider à détecter certaines maladies, voire proposer des traitements adaptés. Ils contribuent à la reconnaissance et à la classification des tumeurs, et permettent de prédire la maladie de Parkinson en détectant ses premiers symptômes et en suivant son évolution.

Toutes ces applications ne représentent qu'un échantillon de ce que les RNA peuvent offrir dans les différents domaines. Aujourd'hui, et dans le futur, les réseaux de neurones ne cessent d'évoluer et d'apporter de nouvelles technologies au service de la société.

## 3.5 Structure des Réseaux de Neurones

La structure des réseaux de neurones désigne la façon dont ils sont organisés et connectés entre eux. Elle détermine le type de réseau et le traitement des données. Classifier les réseaux par la manière dont les neurones sont connectés entre eux permet de distinguer deux grandes familles [48] : les réseaux statiques (non bouclés) et les réseaux dynamiques (bouclés ou récurrents).

### 3.5.1 Réseaux de Neurones non Bouclés

L'information circule dans une seule direction, des neurones d'entrée à ceux de sortie, sans revenir en arrière, ce qui empêche la création de boucles dans les connexions. Ils sont représentés sous forme d'un graphe sans cycle. Les réseaux statiques réalisent une ou plusieurs fonctions algébriques à partir des entrées, en combinant les fonctions effectuées par chaque neurone. Si un neurone fournit directement une sortie du réseau, on l'appelle neurone de sortie ; les autres, situés entre l'entrée et la sortie, sont appelés neurones cachés ; leur rôle est d'effectuer des calculs intermédiaires. Dans cette classe, il existe deux sous-types : les réseaux complètement connectés et les réseaux à couches (ou multicouches).

**Réseaux complètement connectés** : Dans un réseau complètement connecté, les entrées et les neurones sont numérotés séquentiellement. Chaque neurone reçoit toutes les entrées du réseau et les sorties des neurones de numéro inférieur au sien. Sa sortie alimente les neurones de numéro supérieur. Cette organisation crée un flux de données unidirectionnel et strictement hiérarchisé (comme une chaîne de montage), où les calculs s'effectuent dans l'ordre croissant des numéros. Un neurone ne peut jamais recevoir de signal d'un neurone "placé après lui" dans la séquence, garantissant ainsi l'absence totale de boucles.

**Réseaux multicouches** : dans ce réseau, les neurones cachés sont organisés en couches et suivent certaines règles : les neurones d'une même couche ne communiquent pas entre eux, un neurone ne reçoit d'informations que de la couche précédente et, en parallèle, n'envoie des signaux qu'à la couche suivante. C'est-à-dire que les connexions entre couches non adjacentes sont interdites.

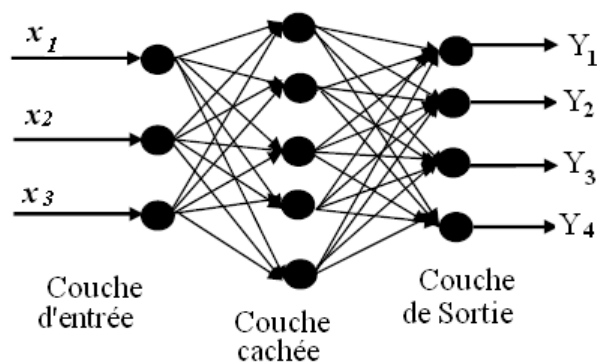


FIGURE 3.3 – Réseaux de neurones non bouclés multicouches

Ces architectures sont très utilisées, notamment en classification ou en reconnaissance de formes (comme la reconnaissance d'écriture ou de visages). À noter que dans ce type de réseau, le temps ne joue aucun rôle : si les entrées ne bougent pas, les sorties restent stables.

### 3.5.2 Réseaux de Neurones Bouclés

Les réseaux de neurones bouclés représentent une structure complexe. Les connexions peuvent former des boucles, c'est-à-dire qu'il est possible de trouver au moins un chemin qui revient à son point de départ. Cela signifie que la sortie d'un neurone peut dépendre d'elle-même, mais à un instant précédent. Afin d'éviter des contradictions comme une valeur qui dépendrait d'elle-même au même instant, les connexions récurrentes dans les

réseaux de neurones intègrent un retard temporel (souvent noté  $z^{-1}$ ), permettant de décaler l'information d'au moins un pas de temps. Ces réseaux sont généralement modélisés par des équations [48] aux différentielles, de la forme :

$$x(k+1) = \Phi(x(k), u(k+1)) \quad (3.1)$$

où :

- $x(k)$  est l'état du système à l'instant discret  $k$ ,
- $u(k+1)$  est une entrée (ou commande) à l'instant  $k+1$ ,
- $\Phi$  est une fonction (linéaire ou non linéaire) décrivant la dynamique du système.

Cela signifie que l'état futur du réseau dépend à la fois de son état actuel et des nouvelles entrées. La sortie du réseau peut également être exprimée à l'instant  $k+1$  à l'aide d'une fonction de sortie :

$$y(k+1) = \Psi(x(k+1), u(k+1)) \quad (3.2)$$

où :

- $y(k+1)$  : Sortie du système à l'instant  $k+1$
- $x(k+1)$  : État interne (calculé par  $x(k+1) = \Phi(x(k), u(k+1))$ )
- $u(k+1)$  : Commande externe
- $\Psi$  : Fonction d'observation

Grâce à leur capacité à mémoriser des séquences, les réseaux récurrents sont particulièrement utiles dans des domaines comme le traitement du langage naturel, la prédiction de séries temporelles ou encore le pilotage de systèmes dynamiques.

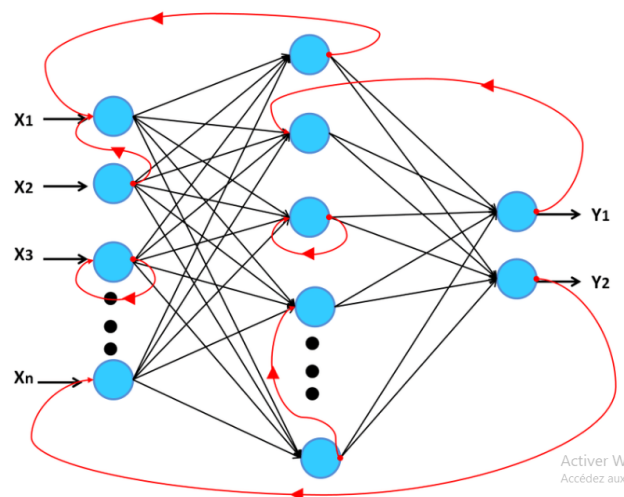


FIGURE 3.4 – Réseaux de neurones bouclé

## 3.6 Architecture des Réseaux de Neurones Artificiels

L'architecture représente la forme ou la structure du réseau. Elle définit l'organisation des couches (entrée, cachées, sortie). Chaque modèle possède une architecture de réseau différente en termes de nombre de couches, de types de couches, de formes de couches et de connexions entre les couches. Ces architectures varient selon le type de problème à résoudre. Ci-dessous, quelques exemples de problèmes avec les architectures correspondantes.

### 3.6.1 Transformers

Transformers utilisent un mécanisme d'attention [39]. Ce mécanisme permet de pondérer l'importance de chaque élément d'une séquence et de capturer les relations et significations entre ces éléments. Cette architecture est structurée comme suit :

Encodeur : transforme l'entrée en représentations contextuelles.

Décodeur : génère la sortie de manière séquentielle.

Chaque bloc contient

Couche d'attention multi-têtes : permet au modèle de focaliser son attention sur différentes parties de la séquence.

Couche de normalisation : son rôle est de stabiliser l'apprentissage.

Couche feedward : applique séparément des transformations à l'information acquise, pour la rendre plus abstraite ou plus utile pour la suite.

Ils sont utilisés dans le traitement du langage naturel. Les modèles les plus courants de ce type sont BERT et RoBERTa. BERT est un modèle développé par Google, de type Transformer, qui est uniquement un encodeur. BERT se caractérise par sa bidirectionnalité, c'est-à-dire qu'il peut lire le texte dans les deux sens en même temps. Le pré-entraînement fait référence à la capacité de BERT à deviner des mots masqués ou à prédire si deux phrases sont consécutives. RoBERTa est une version améliorée de BERT développée par Facebook IA. Il présente un entraînement plus long et un masquage dynamique, où les mots masqués changent à chaque époque.

### 3.6.2 Réseaux de Neurones Convolutifs

Les réseaux convolutifs ont été introduits pour la première fois par Fukushima, qui a développé une architecture de réseau neuronal hiérarchique

et travail de recherche de Hubel [7]. Un réseau neuronal convolutionnel se compose de plusieurs couches.

Couches convolutives : Les couches convolutives constituent le noyau du réseau convolutif. Ces couches se composent d'une grille rectangulaire de neurones qui dispose d'un champ réceptif étendu à travers toute la profondeur du volume d'entrée.

Couches de pooling : Après chaque couche convolutive vient une couche de pooling. Son rôle est de réduire la dimensionnalité de la carte de caractéristiques pour diminuer la complexité et le nombre de paramètres.

Couches totalement connectées : c'est la dernière couche qui vient après les couches convolution et pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches totalement connectées. En effet la dernière couche du réseau convolutif fait une classification spécifique en combinant toutes les caractéristiques spécifiques détectées par les couches précédentes.

### 3.6.3 Réseaux de Neurones Récurrents

Les réseaux neurones récurrents (RNN) se distinguent par une mémoire interne, qui permet de conserver des informations sur les étapes précédentes d'une séquence. Ils fonctionnent en boucle interne ; en effet, ils renvoient la sortie à eux-mêmes ou à d'autres neurones comme entrée, ce qui leur permet de garder en mémoire ces données. D'autre part, grâce à leur mémoire interne, ils peuvent se souvenir des informations sur plusieurs étapes temporelles [41].

Ils sont souvent utilisés pour traiter des séries temporelles. Une série temporelle est un ensemble de données ordonnées dans le temps, par exemple, le prix d'une action en bourse mesuré chaque heure ou bien chaque jour, pour prédire les tendances du marché.

### 3.6.4 Réseaux de Neurones Graphiques

Les réseaux de neurones graphiques sont une classe de modèle d'apprentissage profond conçus pour traiter des données structurées sous forme de graphe. Le principe de fonctionnement de ces réseaux est le suivant : ils traitent des données structurées en graphes. Ils travaillent sur les relations entre les nœuds (objets), chaque nœud partage ses informations avec ses voisins et reçoit en retour d'autres informations. Ce processus permet la mise à jour de chaque nœud. En résumé, on a un graphe, un message, une agrégation et une fonction d'activation (qui détermine les caractéristiques mises à jour)[42]. Les réseaux de neurones graphiques offrent des solutions aux problèmes de contrôle des flux d'énergie, de détection des pannes dans

le système, de compréhension des relations entre les composants du système et de prédiction de la demande. Un exemple est le domaine de l'électronique de puissance, qui est un domaine important et en constante évolution. Il intervient dans des secteurs tels que les énergies renouvelables, les véhicules électriques, les réseaux intelligents, les alimentations sans coupure et bien d'autres.

### 3.7 Limites des Réseaux de Neurones

Les réseaux de neurones sont des outils puissants capables de révolutionner plusieurs domaines. Les obstacles qui limitent leurs performances sont principalement.

Les modèles nécessitent un grand nombre de données étiquetées, ce qui engendre des difficultés lorsque la source de données est médiocre. Par exemple, dans le traitement du langage, les réseaux de neurones rencontrent des difficultés avec des langues régionales comme le tamazight. De même, dans le domaine de la santé, ils font face à des maladies rares.

Le traitement de données requiert un matériel puissant qui coûte cher, et l'accès à des supercalculateurs reste limité. De plus, cela consomme une grande quantité d'énergie, engendrant des coûts considérables. Le temps de travail sur ces modèles est également important. Par exemple, ImageNet nécessite des GPU haut de gamme qui coûtent plus de 10 000 € l'unité, un prix souvent inaccessible aux petites structures.

Les réseaux de neurones ne fournissent pas d'explication sur les résultats obtenus. Par exemple, dans le cadre du recrutement automatisé, où un algorithme est utilisé pour trier les CV, certains candidats peuvent être rejetés sans que le recruteur sache pourquoi. Est-ce une réelle absence de compétence ou un biais appris par le modèle ?

Enfin, le fait que l'algorithme oublie les tâches apprises après être passé à de nouvelles données est également un inconvénient des réseaux de neurones.

### 3.8 Conclusion

Ce chapitre a permis de poser les bases des connaissances nécessaires à la compréhension des RNA. Ces connaissances seront essentielles pour aborder

les travaux pratiques dans les prochains chapitres.

# Chapitre 4

## Approche Hybridée G-RNA pour la Résolution de CE-VRP

### 4.1 Introduction

Ce chapitre fera l'objet de l'application d'une approche hybride pour la résolution d'une variante du VRP.

L'approche hybride proposée est un algorithme génétique doté d'un processus d'apprentissage sous forme de réseaux de neurones artificiels à une certaine étape de l'algorithme génétique classique (comme détaillé ci-dessous), afin d'améliorer ses performances.

La variante du VRP choisie pour cette application est le CE-VRP, connu sous le nom de problème de tournées de véhicules électriques avec contraintes de capacité. Cette variante sera amplement développée dans ce chapitre.

### 4.2 Paramètres spécifiques au CEVRP

#### 4.2.1 Taux de consommation d'énergie

Les véhicules électriques possèdent une capacité de stockage d'énergie limitée (une batterie) ce qui entraîne une autonomie réduite. Pour éviter qu'ils ne tombent en panne de charge, il est primordial d'estimer la quantité d'énergie nécessaire pour son déplacement jusqu'au retour au dépôt [27].

## 4.2.2 Stations de recharge alternative

Intégrer des arrêts reliant les stations de recharge au réseau de clients est nécessaire. Les véhicules ont une autonomie courte et nécessitent une stratégie de ravitaillement. Intégrer des arrêts pour la recharge représente un vrai défi car ça prend du temps et varie d'un véhicule à un autre [27].

## 4.2.3 Temps d'attente à la station de ravitaillement

Les stations de recharges ont des bornes limitées et souvent occupées, donc le véhicule doit attendre. Le temps d'attente d'un véhicule dépend de la disponibilité des bornes et du nombre de véhicules présents. Dans certains modèles, chaque station est représentée comme une file d'attente (M /M /1) [27].

## 4.2.4 Stratégies de ravitaillement

Pour les véhicules électriques, il existe deux façons de recharger la batterie : Recharge complète : consiste à attendre que la batterie soit entièrement rechargée, ou bien remplacer directement une batterie vide par une pleine, ce qui entraîne un temps de recharge fixe.

Recharge partielle : Le véhicule peut repartir avant que la batterie soit complètement chargée. Le temps de recharge dépend de l'énergie qu'on ajoute[27].

## 4.3 Modélisation Mathématique du CEVRP

La modélisation du CEVRP représente une étape essentielle pour optimiser la tournée tout en respectant les contraintes liées à la capacité et à l'autonomie des véhicules électriques sous les paramètres suivants :

- $n$  : Nombre de clients.
- $V = \{0, 1, 2, \dots, n\}$  : Ensemble des nœuds, où 0 représente le dépôt.
- $K$  : Nombre de véhicules électriques disponibles.
- $Q$  : Capacité de charge maximale d'un véhicule (en termes de marchandise).
- $B$  : Capacité maximale de la batterie (en kWh ou en autonomie).
- $d_{ij}$  : Distance entre les nœuds  $i$  et  $j$ .
- $q_i$  : Demande du client  $i$  (en termes de marchandise).

- $e_{ij}$  : Consommation d'énergie pour parcourir l'arc  $(i, j)$ .
- $r$  : Taux de recharge (en kWh par unité de temps).
- $T$  : Temps maximal autorisé pour une tournée.

Les variables de décision sont définies comme suit :

- $x_{ijk}$  : Variable binaire égale à 1 si le véhicule  $k$  traverse l'arc  $(i, j)$ , et 0 sinon.
- $y_{ik}$  : Quantité de marchandise transportée par le véhicule  $k$  après avoir visité le nœud  $i$ .
- $z_{ik}$  : Niveau de charge de la batterie du véhicule  $k$  après avoir visité le nœud  $i$ .
- $t_{ik}$  : Temps d'arrivée du véhicule  $k$  au nœud  $i$ .

Le problème se modélise comme suit :

$$\min \sum_{k=1}^K \sum_{i=0}^n \sum_{j=0}^n d_{ij} \cdot x_{ijk} \quad (1.26)$$

$$\sum_{k=1}^K \sum_{j=0}^n x_{ijk} = 1 \quad \forall i \in \{1, 2, \dots, n\} \quad (1.27)$$

$$\sum_{j=0}^n x_{ijk} = \sum_{j=0}^n x_{jik} \quad \forall i \in V, \forall k \in \{1, \dots, K\} \quad (1.28)$$

$$\begin{aligned} \sum_{j=1}^n x_{0jk} &= 1 \quad \forall k \in \{1, \dots, K\} \\ \sum_{i=1}^n x_{i0k} &= 1 \quad \forall k \in \{1, \dots, K\} \end{aligned} \quad (1.29)$$

$$z_{jk} \leq z_{ik} - e_{ij} \cdot x_{ijk} + B \cdot (1 - x_{ijk}) \quad \forall i, j \in V, \forall k \in \{1, \dots, K\} \quad (1.30)$$

**Recharge complète :**

$$z_{ik} = B$$

**Sinon :**

$$z_{ik} \leq z_{jk} + e_{ji} \cdot x_{jik} \quad \forall i, j \in V, \forall k \in \{1, \dots, K\} \quad (1.31)$$

$$z_{ik} \geq 0 \quad \forall i \in V, \forall k \in \{1, \dots, K\} \quad (1.32)$$

$$t_{ik} = t_{ik} + \frac{B - z_{ik}}{r} \quad (1.33)$$

$$t_{ik} \leq T \quad \forall i \in V, \forall k \in \{1, \dots, K\} \quad (1.34)$$

La formule (1.26) représente la fonction objectif qui minimise le coût total (distance, énergie et temps). Les contraintes de (1.27) à (1.30) représentent les contraintes du problème classique CVRP :

- (1.27) assure que chaque client est visité une fois.
- (1.28) est la contrainte de capacité des véhicules ; la charge totale ne doit pas dépasser  $Q$ .
- (1.29) garantit l'équilibre de flux.
- (1.30) définit le départ et le retour au dépôt.

Les contraintes de (1.31) à (1.34) sont spécifiques aux véhicules électriques :

- (1.31) représente la consommation d'énergie ; elle garantit qu'un véhicule ne dépasse pas son autonomie énergétique.
- (1.32) assure la recharge de la batterie ; elle garantit que le véhicule ne tombe pas en panne d'énergie pendant la tournée.
- (1.33) modélise le temps nécessaire pour recharger la batterie.
- (1.34) garantit que la tournée respecte la durée maximale autorisée.

## 4.4 Avantages et Inconvénients

Les avantages de l'utilisation des véhicules électriques sont assez nombreux. En termes écologiques, ils contribuent à la lutte contre le changement climatique. En effet, les véhicules électriques ne produisent pas de CO<sub>2</sub>, donc représentent une solution prometteuse pour le transport durable. Ils produisent moins de bruit et reviennent moins cher, car l'électricité est moins coûteuse et la maintenance des moteurs électriques est plus simple et moins chère. En outre, les gouvernements encouragent l'achat de ces véhicules en offrant des aides financières.

Les inconvénients sont principalement dus à l'autonomie limitée du véhicule électrique, qui engendre un temps de recharge considérable et une complexité algorithmique plus élevée. D'autre part, les stations de recharge ne sont pas suffisamment étalées sur toutes les zones, surtout les plus reculées. Un point qu'on néglige souvent est que la production de l'électricité elle-même est dérivée du traitement de l'énergie primaire. La propreté des VE se ferait donc au prix d'une augmentation des émissions de carbone du côté de la

production. La solution est bien évidente : c'est d'utiliser des énergies renouvelables pour la production de l'électricité, mais cela reste toujours limité.

## 4.5 Structure de l'approche hybridée G-RNA pour la Résolution du CE-VRP

La combinaison de l'algorithme génétique et des réseaux de neurones artificiels est une approche hybride qui consiste à combiner les forces de ces deux approches. L'algorithme génétique est capable de parcourir un grand ensemble de solutions grâce à sa population diversifiée. Il évite de tomber dans des optima locaux et s'adapte facilement à des problèmes avec différentes variables, raison pour laquelle nous l'avons choisi pour la résolution du CEVRP. Un autre avantage de cet algorithme est qu'il peut travailler avec des solutions de qualité réduite et les évaluer au cours du processus. Cependant, l'algorithme génétique présente des difficultés en ce qui concerne l'exploitation locale, car il nécessite un temps de calcul important, et peut générer des solutions irréalisables nécessitant des réparations. Les réseaux de neurones viennent combler ces lacunes, car ils se caractérisent par une exploitation locale, une gestion de contraintes précise et un temps de calcul rapide. Les réseaux de neurones utilisés dans cet algorithme sont les réseaux de neurones convolutifs (CNN), un modèle d'intelligence artificielle qui apprend automatiquement à reconnaître des motifs importants dans des données organisées, comme des images ou des séries de chiffres. Il réalise cela en analysant les données progressivement tout en conservant la structure et les relations entre les éléments. La résolution du problème de routage de véhicules électriques avec contraintes capacité (CEVRP) avec cette approche repose sur deux phases importantes : la phase génétique, la phase d'apprentissage.

Phase génétique : la représentation des solutions sous forme de chromosomes où chaque gène correspond à une séquence de clients servis par un véhicule. La population initiale est générée aléatoirement. Le croisement et l'évaluation respectent les contraintes spécifiques du CEVRP.

Phase d'apprentissage : cette phase constitue l'innovation de l'approche hybride proposée. Les réseaux de neurones convolutifs sont entraînés à évaluer la qualité des solutions générées et à apprendre les motifs caractéristiques des bonnes solutions. Les matrices d'adjacence sont utilisées dans cette approche pour représenter les solutions du CEVRP sous une

forme compatible avec un réseau de neurones convolutif. Elles permettent d'encoder la structure des routes (séquences de clients et dépôts), capturer les relations spatiales entre les points de livraison, servir d'entrée au CNN pour évaluer la qualité des solutions.

L'approche génétique hybridée d'un processus de RNA est structurée de ma manière suivante.

### Préparation des données et codage des individus

Pour représenter les données des clients et du dépôt, chaque client est modélisé sous forme d'un dictionnaire contenant le numéro du client, ses coordonnées géographiques  $(x, y)$  ainsi que sa demande. Le dépôt, qui représente le point de départ et d'arrivée, a une demande nulle. Par exemple :

```
depot = {"id": 0, "x": 35, "y": 35, "demand": 0}
client1 = {"id": 1, "x": 41, "y": 49, "demand": 10}
client2 = {"id": 2, "x": 35, "y": 17, "demand": 7}
```

Ensuite, une matrice de distances est construite. Il s'agit d'une matrice carrée, qui stocke les distances euclidiennes entre toutes les paires de points (le dépôt et les clients). Un exemple de matrice de distances est donné ci-dessous :

	Dépôt (0)	Client 1 (1)	Client 2 (2)
Dépôt (0)	0.0	15.23	18.0
Client 1 (1)	15.23	0.0	32.56
Client 2 (2)	18.0	32.56	0.0

Concernant le codage des individus (c'est-à-dire des solutions), chaque solution est représentée sous la forme d'une liste de routes. De la manière suivante :

```
[[0, 12, 5, 0], [0, 8, 3, 0]]
```

Les solutions ainsi générées (sous forme de listes de routes) sont ensuite encodées sous forme de matrices d'adjacence binaires, afin d'être compatibles avec l'entrée d'un réseau de neurones convolutif (CNN). Une normalisation des distances est également appliquée, ramenant les valeurs dans l'intervalle  $[0, 1]$ .

Cette transformation permet au CNN d'extraire à la fois des motifs spatiaux locaux (par exemple, des sous-routes fréquentes) et des motifs

globaux (telle que la structure générale des tournées).

### **Population initiale**

La création d'une population de solutions se fait de manière aléatoire. Dans notre cas, une solution doit respecter la capacité des véhicules, l'autonomie de la batterie. En effet, tant qu'il reste des clients non visités on crée une nouvelle route commençant au dépôt (0) et on initialise la capacité restante et la batterie restante. De nouvelles routes sont générées tant que le véhicule peut ajouter des clients : sélection aléatoire d'un client non visité puis vérification des contraintes de capacité et d'énergie. Si le client est valide, on l'ajoute à la route et on met à jour la capacité restante et l'énergie restante. Si aucun client n'est valide, retourne au dépôt et on termine la route. Ce processus est répété pour générer N solutions aléatoires. Cette génération aléatoire permet de maintenir une diversification dans la population initiale.

### **Échantillonnage**

L'algorithme choisit parmi les solutions initiales un échantillon de 10 solutions d'une manière aléatoire, ce qui peut aider à maintenir la diversité et améliorer la performance. Ces solutions sont converties en matrice d'adjacence pour servir d'entrée pour les CNN.

Par exemple : le véhicule démarre du dépôt au 1<sup>er</sup> client, puis au 3<sup>e</sup>, et enfin revient au dépôt

```
[
  [0, 1, 0, 0, 0], # 0 → 1
  [0, 0, 0, 1, 0], # 1 → 3
  [0, 0, 0, 0, 0],
  [1, 0, 0, 0, 0], # 3 → 0
  [0, 0, 0, 0, 0]
]
```

### **Sélection**

La sélection détermine combien de fois un individu participe à la reproduction. Les individus ayant les meilleures performances sont sélectionnés plus souvent. Pour améliorer cette phase, on associe un processus d'intelligence artificielle : les RNA. Les réseaux de neurones convolutifs sont entraînés à évaluer la qualité des solutions générées et à apprendre les motifs caractéristiques des bonnes solutions. Les matrices d'adjacence sont utilisées dans cette approche pour représenter les solutions du CEVRP sous une forme compatible avec un réseau de neurones convolutif. Elles permettent d'encoder la structure des routes (séquences de clients et dépôts), de capturer les relations spatiales entre les points de livraison, et de servir d'entrée au CNN pour

évaluer la qualité des solutions. Après entraînement, le modèle parvient à estimer la qualité des solutions avec précision, puis effectue une présélection par CNN (rapide), analysant les motifs structurels des solutions, ce qui permet de filtrer les solutions médiocres. La solution prometteuse suit une évaluation exacte, qui repose sur l'ensemble des contraintes opérationnelles.

1. Distance totale (40%) : Selon les travaux de Toth et Vigo (2014)[44], la minimisation de la distance parcourue représente le facteur dominant dans l'optimisation des VRP, justifiant son poids important.
2. Nombre de véhicules (30%) : Comme démontré par Golden et al (2008)[45], la réduction du nombre de véhicules impacte directement les coûts opérationnels, d'où son allocation de 30%.
3. Énergie restante (20%) : Les recherches de Pelletier et al. (2016) [46] sur les VEVRP soulignent l'importance critique de la gestion énergétique, particulièrement pour les flottes électriques.
4. Taux de charge (10%) : L'étude de Laporte (2009) [47] montre qu'une utilisation optimale de la capacité améliore l'efficacité globale sans être aussi déterminante que les autres facteurs.

### Croisement

L'algorithme utilise un croisement client-par-client intelligent spécialement conçu pour le VRP électrique. Cette méthode combine progressivement les solutions parentes en sélectionnant à chaque étape le client valide le plus proche, tout en respectant scrupuleusement les contraintes de capacité et d'autonomie batterie. Contrairement à des méthodes de croisement classiques, elle garantit que chaque solution enfant reste réalisable en vérifiant systématiquement l'énergie nécessaire pour rejoindre le client suivant et retourner au dépôt. L'approche privilégie les connexions courtes entre clients, ce qui favorise naturellement la réduction des distances tout en maintenant une diversité génétique essentielle à l'optimisation.

**Parent1** : [0, 1, 3, 0, 0, 2, 4, 0]

**Parent2** : [0, 3, 1, 0, 0, 4, 2, 0]

#### Déroulement du croisement :

1. Séquence des parents :
  - Parent1 : [1, 3, 2, 4]
  - Parent2 : [3, 1, 4, 2]
2. Construction pas à pas :
  - Position 0 : Compare 1 (P1) et 3 (P2)
  - Supposons que 1 soit valide → Enfant : [1]

- Position 1 : Compare 3 (P1) et 3 (P2)  
3 valide → Enfant : [1, 3]
- Position 2 : Compare 2 (P1) et 1 (P2)  
2 valide mais 1 déjà visité → Enfant : [1, 3, 2]
- Position 3 : Compare 4 (P1) et 4 (P2)  
4 valide → Enfant : [1, 3, 2, 4]

Conversion finale :

Enfant : [0, 1, 3, 2, 4, 0]

### **Évaluation**

Après le croisement, l'algorithme effectue une évaluation en analysant la solution enfant générée. Elle compare ses performances (distance parcourue, nombre de véhicules, consommation énergétique) à celle des solutions parents à l'aide d'un score pondéré. Seul l'enfant démontrant une amélioration significative remplace le parent le moins performant, tandis qu'un mécanisme élitiste conserve la meilleure solution globale. Cette étape rigoureuse assure une optimisation progressive tout en maintenant l'équilibre entre exploitation des bonnes solutions et exploration de nouvelles configurations, garantissant ainsi l'amélioration itérative de la qualité des solutions pour le problème de routage électrique.

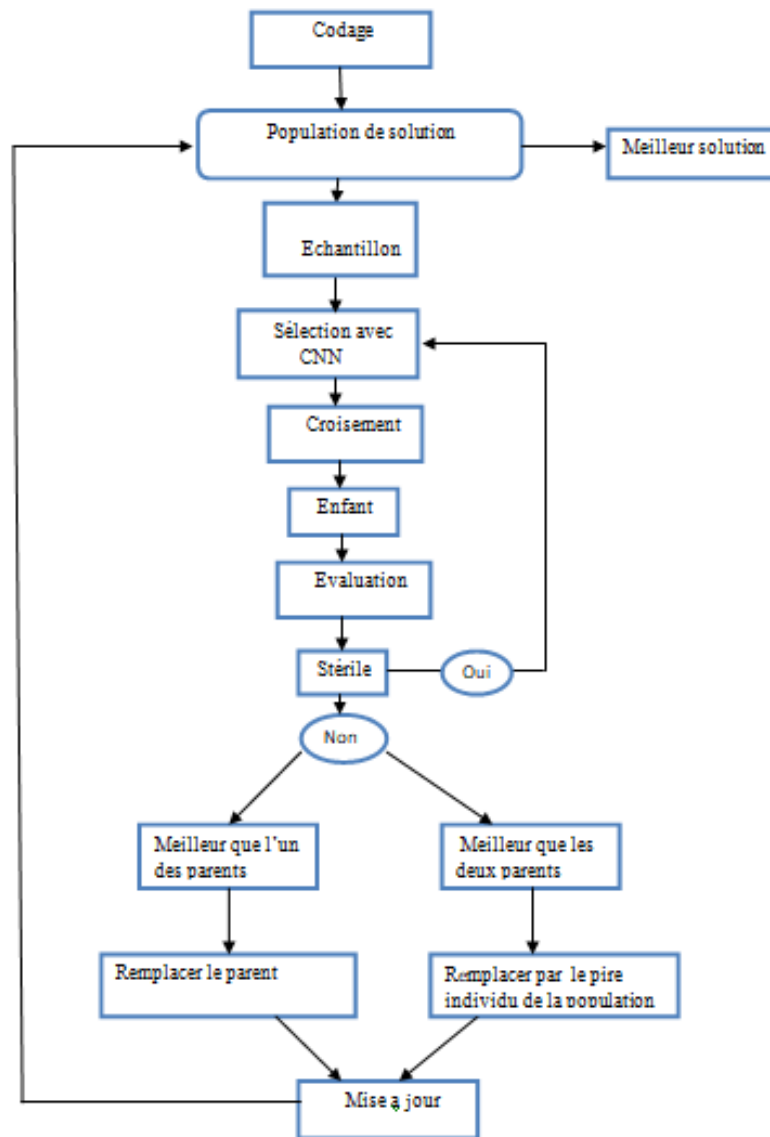


FIGURE 4.1 – Approche génétique hybridée d’un processus RNA

## 4.6 Approche hybridée G-RNA proposée pour la résolution du CE-VRP

L’algorithme proposé combine les avantages d’un algorithme génétique (AG) avec la capacité d’évaluation rapide d’un réseau de neurones convolutif (CNN), appliqué au problème du CEVRP. Il se déroule comme suit :

**Entrée:** Clients, Dépôt, Capacité\_veh, Capacité\_batt, Consommation,  $n$

**Sortie:** Meilleure solution et distance minimale

```

1 Créer une liste SOLUTIONS;
2 for  $i \leftarrow 1$  to  $n$  do
3   | Générer une solution aléatoire et l'ajouter à SOLUTIONS;
4 Entraîner le CNN sur SOLUTIONS;
5 meilleure_solution  $\leftarrow$  vide, meilleure_distance  $\leftarrow$  infini;
6 for génération  $\leftarrow 1$  to 10 do
7   POPULATION  $\leftarrow n$  solutions aléatoires;
8   PARENTS  $\leftarrow$  vide;
9   for  $k \leftarrow 1$  to 2 do
10    | Choisir 10 solutions au hasard;
11    | Garder celle avec le plus bas score CNN comme parent;
12    | L'ajouter à PARENTS;
13 Initialiser enfant vide, position = Dépôt;
14 capacité = Capacité_veh, batterie = Capacité_batt;
15 clients_restants  $\leftarrow$  tous les clients;
16 while clients_restants non vide do
17   | client_choisi  $\leftarrow$  client valide de parent1 ou parent2;
18   | if aucun client valide then
19     | client_choisi  $\leftarrow$  plus proche valide;
20   | if client_choisi then
21     | L'ajouter à l'enfant, mettre à jour état;
22   | else
23     | Retour au dépôt, réinitialiser capacité et batterie;
24 distance  $\leftarrow$  CALCULER_DISTANCE(enfant);
25 if distance < meilleure_distance then
26   | meilleure_distance  $\leftarrow$  distance;
27   | meilleure_solution  $\leftarrow$  enfant;
28 | Afficher génération et distance;
29 Afficher meilleure solution et distance minimale;

```

**Algorithm 2:** Algorithme génétique hybridée d'un processus RNA

## 4.7 Conclusion

Une méthode d'optimisation des tournées de véhicules électriques a été développée en combinant des algorithmes génétiques – capables de générer

une grande variété d'itinéraires – avec des réseaux de neurones, utilisés pour évaluer rapidement la qualité des solutions. Cette approche permet de réduire les temps de calcul tout en respectant les contraintes d'autonomie et de capacité des véhicules. Elle offre également un potentiel d'amélioration continue grâce à l'apprentissage automatique. Le chapitre suivant présentera des tests sur des cas concrets et une comparaison objective avec des méthodes existantes.

# Chapitre 5

## Analyses des Résultats

### 5.1 Introduction

Dans chaque expérience, l'analyse des résultats constitue une étape cruciale. Elle permet d'évaluer l'efficacité de l'approche ainsi que ses limites d'adaptation à son environnement. L'approche proposée est testée sur différentes instances. En effet, les tests ont été menés sur un PC HP équipé d'un processeur Intel Core i3.

Ce chapitre décrit, dans un premier temps, l'environnement dans lequel les expérimentations seront réalisées, puis une présentation des instances du CE-VRP sur lesquelles sera appliquée l'approche proposée. Ensuite, on expose l'analyse des résultats obtenus. Ce chapitre est clos par une synthèse des points clés et des pistes d'amélioration possibles. L'objectif est de donner une vision claire, équilibrée et accessible des performances observées et des perspectives d'évolution.

### 5.2 Environnement d'implémentation de l'approche hybridée G-RNA

Des expériences ont été réalisées sur un ordinateur portable HP avec la configuration suivante : un processeur Intel Core i3-5005U double cœur cadencé à 2,00 GHz. Ce processeur de 5e génération, conçu pour une faible consommation d'énergie, convient aux tâches bureautiques et aux calculs modérés, bien qu'il montre ses limites face aux traitements intensifs ou parallèles. Avec une mémoire vive de 4 Go, cette capacité restreinte impose des contraintes sur la taille des données pouvant être traitées simultanément, et peut allonger les temps d'exécution lors de l'évaluation de solutions com-

plexes ou de l'entraînement de modèles. La carte graphique intégrée Intel HD Graphics 5500, ne disposant pas d'un accélérateur matériel dédié, limite les performances dans les tâches liées à l'apprentissage automatique, en particulier en apprentissage profond. Le stockage SSD/HDD non spécifié offre une capacité suffisante pour héberger les données de test ainsi que les outils nécessaires au développement et à l'exécution des algorithmes. Le système d'exploitation est Windows 10 Professionnel 64 bits, architecture x64. Ce système assure la compatibilité avec les environnements de développement modernes ainsi qu'avec les bibliothèques d'optimisation et d'apprentissage automatique. Malgré une configuration modeste, cette machine permet l'exécution des algorithmes développés sur des instances de taille raisonnable, offrant ainsi un environnement de test réaliste, proche des conditions rencontrées avec du matériel à faible coût, en utilisant l'éditeur visual code et le langage python.

### 5.3 Description des instances

Cette partie présente les différentes instances utilisées pour évaluer l'algorithme conçu. Ces instances correspondent à des cas réels considérés comme instances de référence [50]. Chaque instance est composée des éléments suivants : Nombre de clients, Position géographique des clients et du dépôt, Demande des clients, taille de la flotte de véhicules, Capacité des véhicules, Teneur de la charge électrique des véhicules.

Instance utilisée :

#### **Instance E-n101-k8**

L'instance E-n101-k8 est issue de la littérature classique sur le (CVRP), proposée par Christofides et Eilon. Elle est souvent utilisée comme benchmark dans les études de performance des algorithmes d'optimisation.

Cette instance comporte 101 nœuds : 1 dépôt (point de départ et d'arrivée des véhicules), 100 clients, chacun avec une demande spécifique. Une capacité de 200 unités par véhicules et une demande qui varie entre 1 et 41 unités par clients. L'analyse des coordonnées des 100 clients montre une dispersion sur l'ensemble du plan. Les clients ne sont pas regroupés en clusters clairement définis, mais plutôt répartis de façon relativement homogène sur une large zone. On peut noter : Il n'y a pas de concentration importante de clients dans une région spécifique, comme on le verrait dans une répartition en cluster.

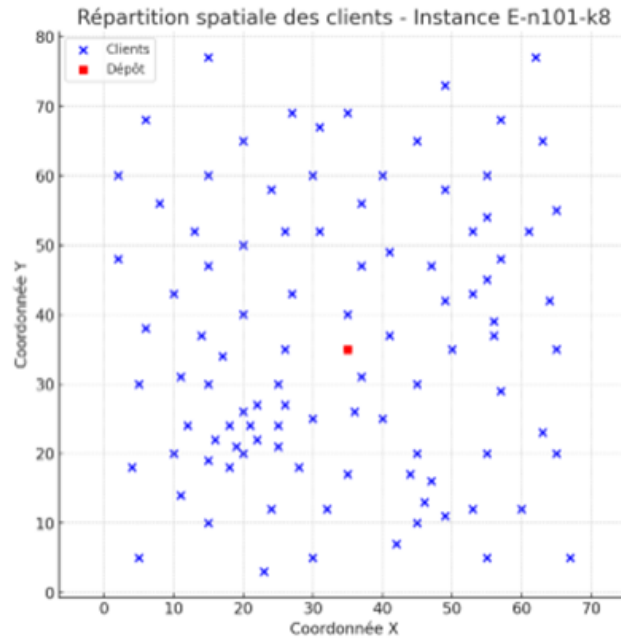


FIGURE 5.1 – instance E-n101-k8

#### Instance P-n101-k4

L'instance P-n101-k4 est une instance classique du problème de tournées de véhicules à capacité limitée (CVRP), issue de la collection d'Augerat et al. Elle est composée de 101 nœuds, dont 1 dépôt et 100 clients répartis dans un espace bidimensionnel. Le dépôt est situé au point de coordonnées (35, 35).

Les clients sont modérément dispersés sur le plan, sans formation claire de clusters. On observe toute fois une légère concentration de points dans la partie centrale et nord-ouest du plan, avec quelques clients isolés. Cette configuration impose une certaine flexibilité dans la construction des tournées, avec des trajets longs pour desservir certains clients éloignés.

La demande des clients varie entre 1 et 41 unités, avec une majorité de valeurs comprises entre 5 et 30 et une capacité de 400 unités par véhicule. Cette hétérogénéité de la demande rend le problème complexe, car elle impose une gestion fine de la capacité disponible dans chaque véhicule pour éviter les dépassements. Le dépôt a une demande nulle.



FIGURE 5.2 – instance P-n101-k4

### Instance M-n101-k10

L'instance M-n101-k10 comprenant 100 clients et un dépôt central. Chaque client a une demande comprise entre 10 et 50 unités, tandis que chaque véhicule dispose d'une capacité maximale de 200 unités. Les clients sont répartis en clusters. Le dépôt est situé au centre de la carte (40, 50), ce qui favorise un certain équilibre pour la distribution vers différentes régions. Les demandes varient fortement, allant de 10 à 50 unités par client.

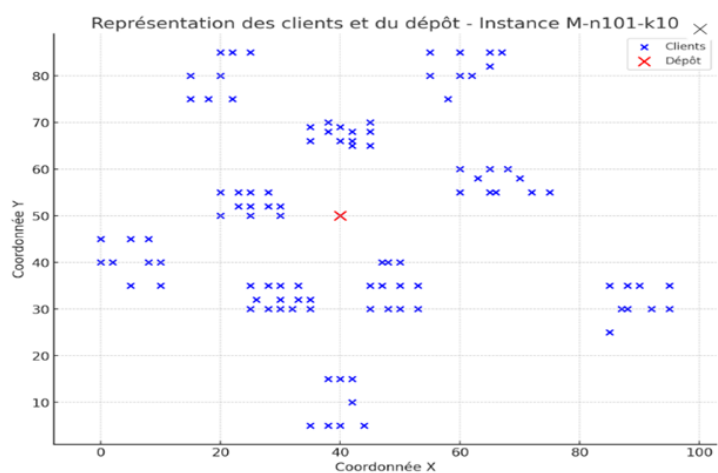


FIGURE 5.3 – L'instance M-n101-k10

### Instance B-n78-k10

comprend 78 nœuds dont un dépôt (nœud 1) et 77 clients avec des demandes variées. La capacité maximale est de 100 unités. Les clients sont

spatialement dispersés, avec des concentrations visibles dans les quadrants nord-est et sud-ouest du plan. Plusieurs zones à forte densité de demande (notamment autour des clients 28, 40 et 54) peuvent rapidement saturer la capacité des véhicules, ce qui nécessitera une répartition fine et optimisée des tournées. Cette configuration accentue les défis logistiques liés à l'équilibrage des charges et à la minimisation des distances totales. La variété des demandes et la répartition géographique rendent cette instance particulièrement intéressante pour évaluer l'efficacité de l'algorithme. Les clients sont regroupés en plusieurs clusters géographiques.

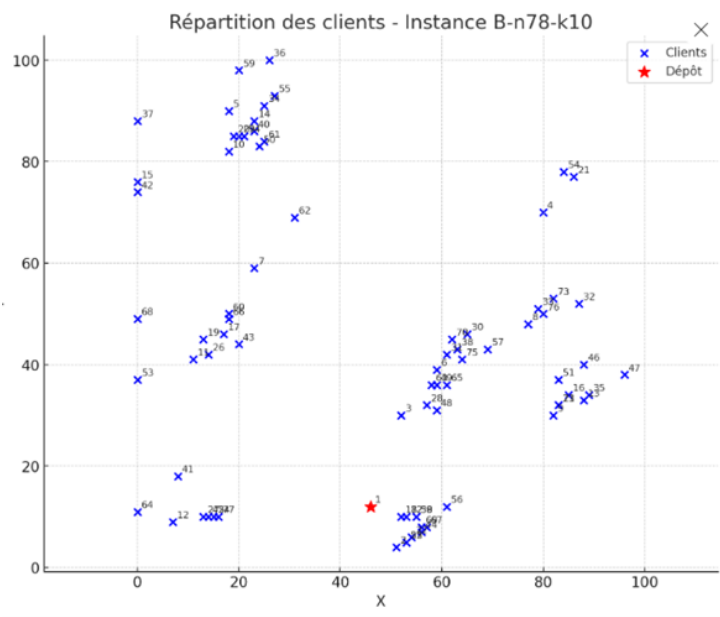


FIGURE 5.4 – L'instance B-n78-k10

## 5.4 Analyse des Résultats

Après la mise en œuvre de notre algorithme d'optimisation pour le CVRP, nous présentons ici l'analyse des performances obtenues sur les instances présentées. Cette section analyse la distance totale parcourue (en kilomètres, km), le taux d'utilisation de l'énergie(kilowattheure (kWh)), le nombre de véhicules mobilisés et la capacité de charge (exprimée en pourcentage). L'analyse est faite en variant la population d'une part et puis l'échantillon de travail.

### Instance E-n101-k8

**Population :** La population de solution varie entre 100 et 1000 avec un échantillon stable de 100 solutions aléatoires.

	100	200	300	400	500	600	700	800	900	1000
Distance	2572	2564	2544	2540	2626	2571	2675	2572	2569	2572
Energie	2439	2493	2439	2341	2369	2409	2384	2239	2372	2490
Capacité	43	38	38	40	40	36.5	36.5	38	38.42	40.6
Nb- véhicule	19	18	18	18	18	18	18	18	17	18

FIGURE 5.5 – Tableau des résultats de la population E-n101-k8

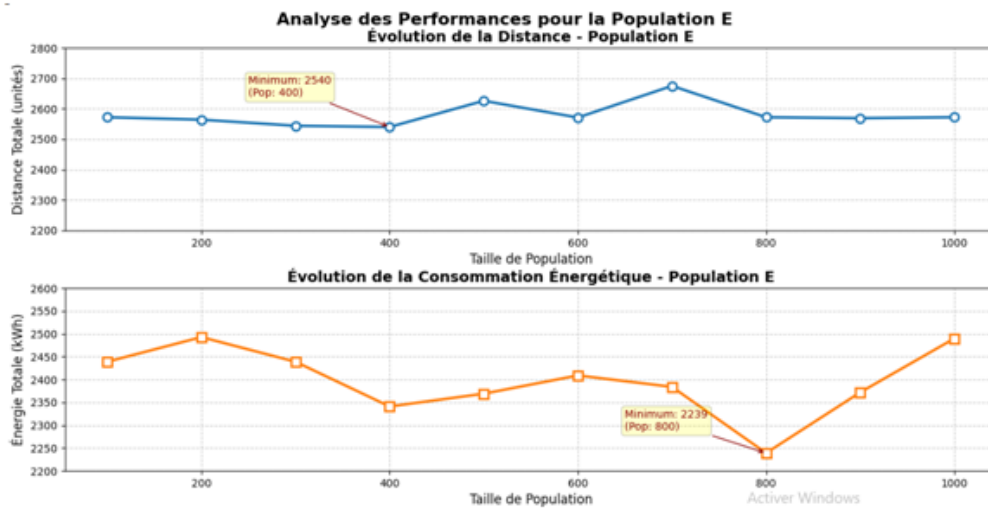


FIGURE 5.6 – Population E-n101-k8

**Echantillon :** Un échantillon de solution varie entre 10 et 100 avec une population stable de 500 solutions aléatoires.

	10	20	30	40	50	60	70	80	90	100
Distance	2579	2561	2560	2483	2479	2492	2312	2522	2428	2562
Energie	2384	2393	2409	2431	2395	2402	2278	2298	2369	2259
Capacité%	38.42	40.52	38.42	38.42	38.42	40.5	40.56	40.58	40.56	38
Nb- véhicules	18	18	18	18	17	17	16	18	17	18

FIGURE 5.7 – Tableau des résultats de variation de l'échantillon population E-n101-k8

#### Instance P-n101-k4

Population :

	100	200	300	400	500	600	700	800	900	1000
Distance	2379	2435	2486	2372	2409	2324	2350	2385	2361	2365
Energie	2481	2425	2448	2464	2525	2448	2389	2320	2361	2425
Capacité	26	24	24	26	26	26	26	26	26	26
Nb- véhicule	17	17	18	17	17	16	17	17	17	17

FIGURE 5.8 – Tableau des résultats de variation de la population P-n101-k4

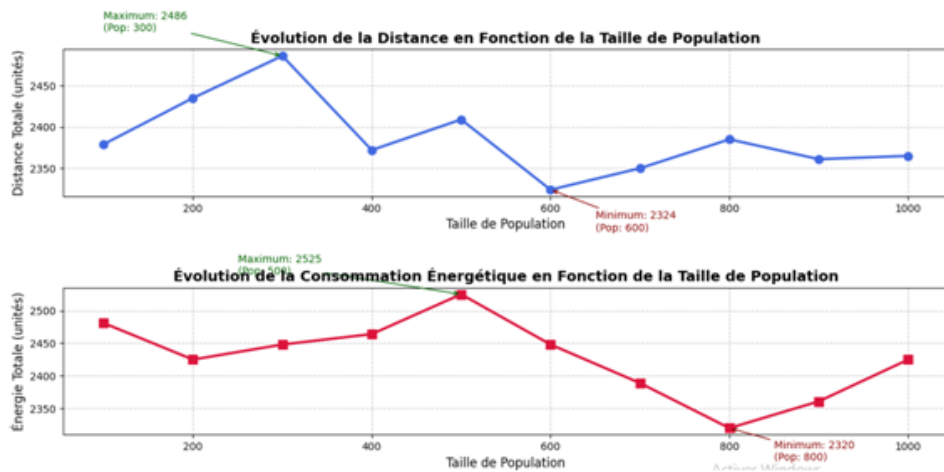


FIGURE 5.9 – La population P-n101-k4

**Echantillon :**

	10	20	30	40	50	60	70	80	90	100
Distance	2579	2561	2560	2483	2479	2492	2312	2522	2428	2562
Energie	2384	2393	2409	2431	2395	2402	2278	2298	2369	2259
Capacité%	38.42	40.52	38.42	38.42	38.42	40.5	40.56	40.58	40.56	38
Nb- véhicules	18	18	18	18	17	17	16	18	17	18

FIGURE 5.10 – Tableau des résultats de variation de l'échantillon la population P-n101-k4

### Instance B-n78-k10

L'algorithme s'arrête sans erreur apparente sur l'instance B-n78-k10; il existe quelques explications théoriques, parmi lesquelles les demandes clients élevées (jusqu'à 26), combinées à la capacité limitée des véhicules (100), créent des configurations complexes où la moindre erreur d'affectation bloque la génération de solutions valides.

Après analyse, on a constaté que le client numéro 35 nécessite 180 unités d'énergie alors que dans notre algorithme, nous avons toujours utilisé 150, et pour régler ce problème, nous avons augmenté la capacité d'énergie à 200 unités.

**Population :**

	100	200	300	400	500	600	700	800	900	1000
Distance	2772	2654	2607	2383	2453	2690	2428	2709	2805	2766
Energie	2538	2561	2431	2586	2667	2639	2400	2414	2611	2595
Capacité	66	67	66.93	66.93	62.5	66.93	67	72	67	67
Nb- véhicule	15	14	14	13	13	14	13	14	15	16

FIGURE 5.11 – Tableau des résultats de variation de la population B-n78-k10

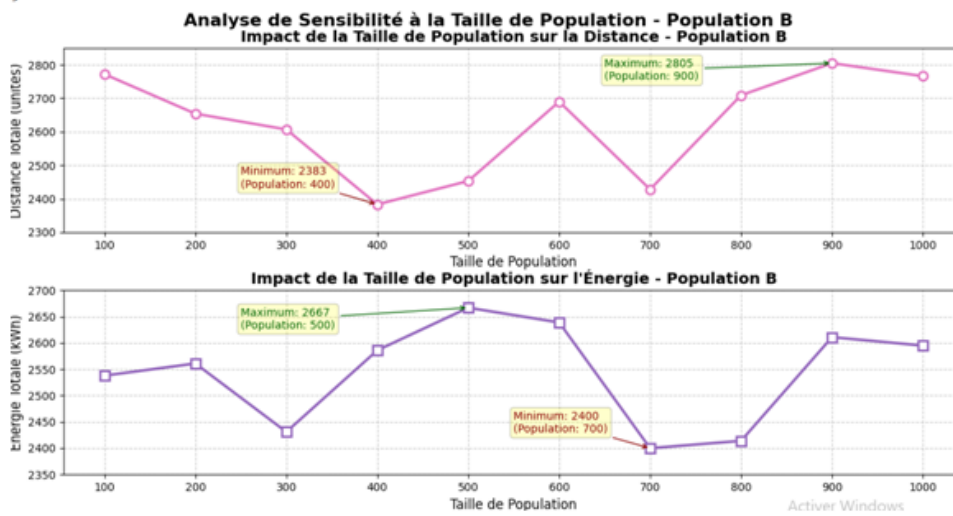


FIGURE 5.12 – Population B-n78-k10

**Echantillon :**

	10	20	30	40	50	60	70	80	90	100
Distance	2779	2735	2693	2452	2671	2466	2552	2630	2452	22478
Energie	2514	2494	2527	2487	2267	2395	2530	2370	2417	2415
Capacité	66.9	66.9	66.9	66	72	66	66	66	72	67
Nb- véhicule	16	15	15	13	14	13	14	14	13	13

FIGURE 5.13 – Tableau des résultats de variation l'échantillon de la population B-n78-k10

### Instance M-n101-k10

#### Population :

	100	200	300	400	500	600	700	800	900	1000
Distance	2557	2538	2523	2554	2303	2502	2517	2570	2530	2557
Energie	2416	2499	2393	2328	2276	2277	2300	2333	2258	2398
Capacité	50.56	53.53	53.53	50.56	50.56	50.56	53.53	53.56	50.56	56.88
Nb- véhicule	18	18	19	18	16	18	18	18	17	18

FIGURE 5.14 – Tableau des résultats de variation de la population M-n101-k10

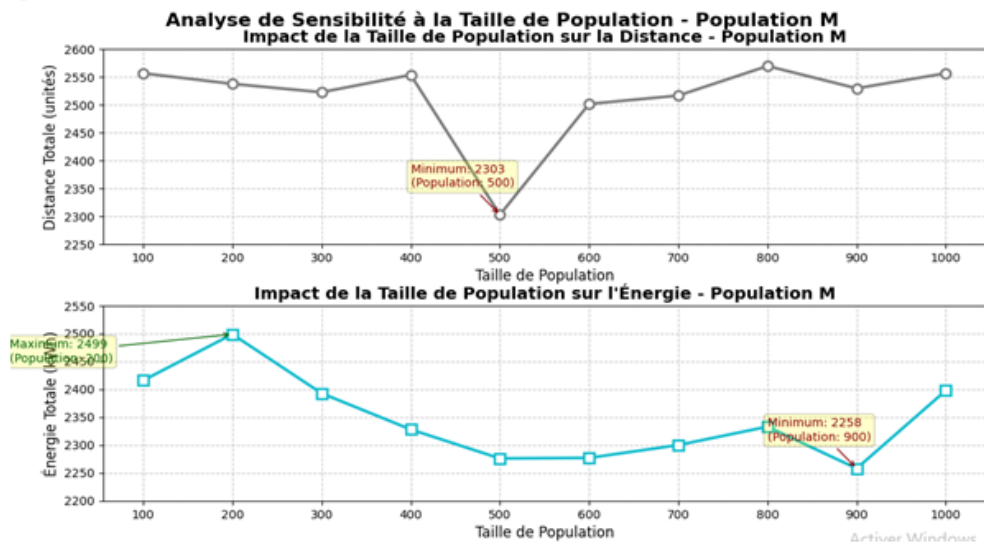


FIGURE 5.15 – Population M-n101-k10

Echantillon :

	10	20	30	40	50	60	70	80	90	100
Distance	2596	2242	2403	2431	2430	2396	2375	2440	2345	2408
Energie	2500	2307	2180	2274	2288	2295	2320	2207	2199	2301
Capacité	53.53	53.53	50.56	53	53	53	53	56	56	57
Nb- véhicule	18	17	17	17	17	17	17	17	17	17

FIGURE 5.16 – Tableau des résultats de variation de l'échantillon la population M-n101-k10

### Analyse

L'étude des différentes instances (E-n101-k8, P-n101-k4, B-n78-k10 et M-n101-k10) montre des tendances claires sur les performances algorithmiques.

On remarque que l'augmentation de la taille de la population améliore systématiquement les résultats, avec un optimum situé entre 300 et 500 solutions. Un échantillon de 50 à 70 solutions suffit pour obtenir une évaluation fiable.

E-n101-k8 et P-n101-k4 partagent les mêmes points géographiques, mais avec des capacités différentes, 200 unités pour E-n101-k8 contre 400 pour P-n101-k4. Cette différence explique la convergence plus rapide de P-n101-k4.

En revanche, les contraintes énergétiques, comme celles rencontrées avec B-n78-k10 où une demande sous-estimée a nécessité une adaptation dynamique des capacités énergétiques, car une mauvaise estimation de la demande peut fortement influencer les résultats.

Chaque instance a son propre comportement propre : E-n101-k8 tire avantage d'une grande population, en effet la distance diminue de 12% lorsque la population passe de 100 à 1000 solutions, montrant l'impact de la diversité initiale, P-n101-k4 se démarque par une convergence rapide, et M-n101-k10 se montre particulièrement robuste avec une faible variabilité des résultats.

En résumé, l'approche reste globalement stable dans son fonctionnement, mais doit être ajustée sur certains cas spécifiques. Les résultats montrent que l'architecture dans son ensemble est solide, tout en donnant des pistes d'amélioration.

## 5.5 Comparaison entre l'approche proposée G-RNA et un AG classique

L'approche proposée utilise les CNN pour sélectionner les parents. Pour mieux évaluer l'efficacité de cette méthode, il est nécessaire de la comparer à une autre approche classique. Dans ce cadre, un autre algorithme a été réalisé, il comprend les mêmes éléments de base, mais les réseaux de neurones ont été éliminés. À la place, une sélection de parents aléatoire a été utilisée.

L'instance P, on a pris une population de 500 solutions avec un échantillon de 100 solutions aléatoires. On remarque un pourcentage d'amélioration de 16,48. Passant de 2798 km pour l'AG classique à 2402 km pour l'algorithme G-RNA.

L'instance B, on a pris une population de 500 solutions avec un échantillon de 100 solutions aléatoires. On observe un pourcentage d'amélioration de 7,7. Allant de 2670 km pour l'AG classique à 2428 km pour l'algorithme

G-RNA.

Pour l'instance M, avec une population de 500 solutions et un échantillon de 100 solutions aléatoires, on note une amélioration de 10,5. Passant de 2645 km pour l'AG classique à 2367 km pour celui G-RNA.

Enfin, pour l'instance E, toujours avec une population de 500 solutions et un échantillon de 100, on constate une amélioration de 11,8. Allant de 2684 km pour l'AG classique à 2369 km pour l'algorithme G-RNA.

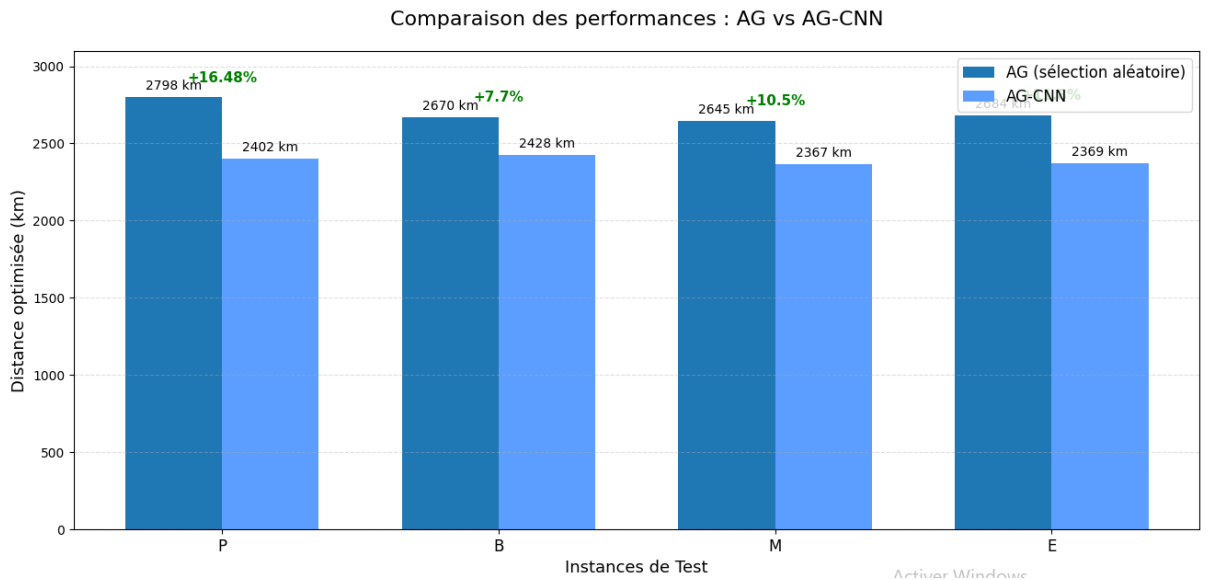


FIGURE 5.17 – Comparaison des distances optimisées : AG vs AG-CNN

## 5.6 Conclusion

On conclut que l'approche génétique hybridée d'un processus RNA assure la résolution de plusieurs instances du CE-VRP. L'hybridation proposée apporte une amélioration de la solution procurée par l'algorithme génétique classique. Il sera intéressant d'insérer le processus de RNA à plusieurs niveaux de la phase de reproduction de l'algorithme génétique classique et de voir l'apport de ces hybridations.

# Conclusion

Le travail réalisé dans ce mémoire porte essentiellement sur le problème de tournée de véhicules. La variante du problème traité comporte des contraintes d'énergie et de capacité de charge. Elle s'intitule problème de tournée de véhicule électrique à contraintes de capacité. Le CE-VRP est un problème central en logistique durable. L'intérêt de cette étude est de proposer une approche innovante pour la résolution d'un CE-VRP intégrant des techniques d'intelligence artificielle.

À travers cinq chapitres, nous avons présenté l'état de l'art du PVC et de ses variantes, en mettant en lumière l'évolution des modèles et des contraintes introduites pour refléter des situations réelles. Par la suite, nous avons étudié les méthodes de résolution des problèmes complexes, incluant les approches exactes, heuristiques, méta-heuristiques et hybrides. Dans le troisième chapitre, nous avons abordé les réseaux de neurones, tout en décrivant leur structure et leur architecture.

La description de l'approche proposée, génétique hybridée d'un processus RNA pour la résolution du CE-VRP, a fait l'objet du quatrième chapitre. En effet, l'approche proposée combine l'algorithme génétique et les réseaux de neurones, précisément les CNN. Ces derniers ont été mis en place dans la phase de sélection des solutions parents à partir d'un échantillon de solutions issues d'une population de solutions aléatoires. Cette méthode permet d'assurer une intensification des solutions tout en maintenant la diversification.

Les résultats expérimentaux de cette approche montrent une réduction significative de l'écart optimal théorique par rapport à une méthode classique d'algorithme génétique reposant sur une sélection aléatoire des solutions parents, ce qui valide l'efficacité de cette stratégie. Cette approche ouvre la voie à une nouvelle génération d'algorithmes intelligents et auto-adaptatifs.

Pour finir, nous suggérons une amélioration de cette approche compte

tenu des stations de recharge et des contraintes de temps pour simuler des scénarios de trafic en temps réel, ainsi que l'usage des techniques de l'intelligence artificielle dans les différentes phases de reproduction. L'expérimentation sur des instances à grande échelle est également envisagée.

# Bibliographie

- [1] M. Soh, N. Tsofack, and C.T. Djamegni. *Approche Heuristique Multi Colonie Des Fourmis Pour La Résolution du Problème de Voyageur de Commerce*. In Colloque Africain sur la Recherche en Informatique et en Mathématiques Appliquées, 2020
- [2] E. Vercesi et A. Buchanan. *The Dantzig–Fulkerson–Johnson TSP formulation is easy to solve for few subtour constraints*. Optimization Online, 2024. Disponible en libre accès.
- [3] Z. Zhang, H. Liu, M. Zhou, et J. Wang. *Solving dynamic traveling salesman problems with deep reinforcement learning*. IEEE Transactions on Neural Networks and Learning Systems, 34(4), 2119–2132, 2021.
- [4] C. Archetti, D. Feillet, A. Mor, et M. G. Speranza. *Dynamic traveling salesman problem with stochastic release dates*. European Journal of Operational Research, 280(3), 832–844, 2020.
- [5] C. Hansknecht, I. Joormann, et S. Stiller. *Dynamic shortest paths methods for the time-dependent TSP*. Algorithms, 14(1), 21, 2021.
- [6] F. Focacci, A. Lodi, et M. Milano. *A hybrid exact algorithm for the TSPTW*. INFORMS Journal on Computing, 14(4), 403–417, 2002.
- [7] S. Frizzi, R. Kaabi, M. Bouchouicha, J.-M. Ginoux, F. Fnaiech, et E. Moreau. *Détection de la fumée et du feu par réseau de neurones convolutifs*. Conférence Nationale sur les Applications Pratiques de l’Intelligence Artificielle, 2017.
- [8] P. Agarwal, D. Bagchi, T. Rambha, et V. Pandey. *A Bi-criterion Steiner Traveling Salesperson Problem with Time Windows for Last-Mile Electric Vehicle Logistics*. arXiv preprint arXiv :2409.14848, 2024.
- [9] M. Y. Khachai et E. D. Neznakhina. *Approximation schemes for the generalized traveling salesman problem*. Proceedings of the Steklov Institute of Mathematics, 299, 97–105, 2017.
- [10] M. de Berg, J. Gudmundsson, M. J. Katz, C. Levcopoulos, M. H. Overmars, et A. F. Van Der Stappen. *TSP with neighborhoods of varying size*. Journal of Algorithms, 57(1), 22–36, 2005.

- [11] N. Agatz, P. Bouman, et M. Schmidt. *Optimization approaches for the traveling salesman problem with drone*. *Transportation Science*, 52(4), 965–981, 2018.
- [12] A. K. Mandal et P. K. D. Sarma. *Novel applications of ant colony optimization with the traveling salesman problem in DNA sequence optimization*. In *IEEE 2nd Int. Symp. on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)*, pp. 1–6, 2022.
- [13] R. F. Carpio, J. Maiolini, C. Potena, E. Garone, G. Ulivi, et A. Gasparn. *Mp-stsp : A multi-platform steiner traveling salesman problem formulation for precision agriculture in orchards*. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 2345–2351, 2021.
- [14] S. Doherty, R. Wehbe, K. Plain, R. Harik, et A. Halbritter. *Off-part motion optimization for an automated fiber placement machine using travelling salesman problem*. *Comput Aided Des Appl*, 19(2), 220–237, 2022.
- [15] F. Alesiani, G. Ermis, et K. Gkiotsalitis. *Constrained clustering for the capacitated vehicle routing problem (CC-CVRP)*. *Applied Artificial Intelligence*, 36(1), 1995658, 2022.
- [16] U. I. Fadila, S. Wahyuningsih, et D. Satyananda. *Study of the General Variable Neighborhood Search (GVNS) algorithm and Variable Neighborhood Search (VNS) algorithm in CVRPTW*. In *AIP Conference Proceedings*, Vol. 2639, No. 1, 2022.
- [17] A. Froger, O. Jabali, J. E. Mendoza, et G. Laporte. *The electric vehicle routing problem with capacitated charging stations*. *Transportation Science*, 56(2), 460–482, 2022.
- [18] S. S. Khairunisa. *Penentuan Rute Distribusi Bantuan Bencana Banjir di Kota Bekasi Menggunakan Heterogeneous Fleet Vehicle Routing Problem (HFVRP)*. Thèse de doctorat, 2022.
- [19] D. Barbucha. *Multi-agent approach to the DVRP with GLS improvement procedure*. In *Int. Conf. on Intelligent Decision Technologies*, pp. 117–126. Springer, 2020.
- [20] W. Yang, D. Wang, W. Pang, A. H. Tan, et Y. Zhou. *Marchandises consommées pendant le transit dans les problèmes d’acheminement des véhicules de livraison fractionnés : modélisation et solution*. *IEEE Access*, 8, 110336–110350, 2020.
- [21] H. Zhang, Z. Wang, M. Tang, X. Lv, H. Luo, et Y. Liu. *Dynamic memory memetic algorithm for VRPPD with multiple arrival time and traffic congestion constraints*. *IEEE Access*, 8, 167537–167554, 2020.

- [22] T. Adamo, M. Gendreau, G. Ghiani, et E. Guerriero. *Un examen des progrès récents en matière d'itinéraires de véhicules en fonction du temps*. Revue européenne de recherche opérationnelle, 2024.
- [23] SIMON, Herbert A. ; NEWELL, Allen. *Heuristic Problem Solving : The Next Advance in Operations Research*.
- [24] E. T. Bogue, H. S. Ferreira, T. F. Noronha, et C. Prins. *A column generation and a post optimization VNS heuristic for the vehicle routing problem with multiple time windows*. Optimization Letters, pp. 1–17, 2022.
- [25] T. Chen, F. Chu, J. Zhang, et J. Sun. *Sustainable collaborative strategy in pharmaceutical refrigerated logistics routing problem*. International Journal of Production Research, 62(14), 5036–5060, 2024.
- [26] L. D. P. Pugliese, D. Ferone, P. Festa, F. Guerriero, et G. Macrina. *Combining variable neighborhood search and machine learning to solve the vehicle routing problem with crowd-shipping*. Optimization Letters, pp. 1–23, 2023.
- [27] M. Asghari et S. M. J. M. Al-e. *Green vehicle routing problem : A state-of-the-art review*. International Journal of Production Economics, 231, 107899, 2021.
- [28] M. Akli. *Problème de tournées de véhicules avec contraintes et fenêtre de temps*. Thèse de magister, 2013.
- [29] S. A. Cook. *The complexity of theorem-proving procedures*. Proceedings of the Third Annual ACM Symposium on Theory of Computing, pp. 151–158, 1971.
- [30] Michel Sakarovitch. *Optimisation combinatoire : Programmation discrète*. Hermann, 1984.
- [31] Th. Paschos (dir.), *Optimisation combinatoire 1 : Concepts fondamentaux*, Lavoisier, 2005
- [32] M. Sakarovitch. *Optimisation combinatoire : Graphes et programmation linéaire*. Hermann, 1984.
- [33] D. R. Morrison, S. H. Jacobson, J. J. Sauppe, and E. C. Sewell, “Branch-and-bound algorithms : A survey of recent advances in searching, branching, and pruning,” *Discrete Optimization*, vol. 19, pp. 79–102, 2016.
- [34] K. Ghedira. *Optimisation combinatoire par métaheuristiques : Origines, concepts et éléments de base, algorithmes canoniques et étendus*, Éditions Technip, 2007.
- [35] M. Haj-Rachid, C. Bloch, W. Ramdane-Cherif, et P. Chatonnay. *Différents opérateurs évolutionnaires de permutation : sélections, croisements et mutations*. LIFC, Université de Franche-Comté, Thème 4 OMNI, juillet 2010.

- [36] B. Widrow and M. Lehr. *30 years of adaptive neural networks : Perceptron, Madaline, and backpropagation*. Proceedings of the IEEE, vol. 78, pp. 1415–1442, 1990. doi :10.1109/5.58323.
- [37] W. R. Schwartz, J. Rouat, and J. L. Dussault. *Réseaux de neurones : De la modélisation au traitement du signal*. Presses internationales Polytechnique, 2001.
- [38] K. Ghedira. *Optimisation combinatoire par métaheuristiques : Origines, concepts et éléments de base, algorithmes canoniques et étendus*, Éditions Technip, 2007.
- [39] D. Rothman. *Transformers for Natural Language Processing*, 2<sup>e</sup> édition, Packt Publishing, 2021.
- [40] J.-S. Dessureault. *Infrastructure logicielle d'apprentissage automatique à base de méthodes de traitement du langage naturel*. Thèse de doctorat, Université du Québec à Trois-Rivières, 2024.
- [41] L. Mohammadpour, T. C. Ling, C. S. Liew, et A. Aryanfar. *A survey of CNN-based network intrusion detection*. Applied Sciences, vol. 12, no. 16, p. 8162, 2022.
- [42] Y. Li, C. Xue, F. Zargari, et Y. R. Li. *From graph theory to graph neural networks (GNNs) : The opportunities of GNNs in power electronics*. IEEE Access, vol. 11, pp. 145067–145084, 2023.
- [43] H. Schättler, U. Ledzewicz. *Geometric Optimal control : theory, methods and examples*, Interdisciplinary Applied Mathematics 38, Springer 2012.
- [44] P. Toth, D. Vigo. *Vehicle Routing : Problems, Methods, and Applications*, SIAM, 2014.
- [45] B. Golden et al. *The Vehicle Routing Problem : Latest Advances and Challenges*, Springer, 2008.
- [46] S. Pelletier et al. *Transportation Research Part D : Transport and Environment*, Elsevier, 2016.
- [47] G. Laporte. *Operations Research Perspectives*, Elsevier, 2009.
- [48] M.Y. AMMAR. *Mise en œuvre de réseaux de neurones pour la modélisation de cinétiques réactionnelles en vue de la transposition Batch/Continu*. thèse de doctorat, Institut National Polytechnique de Toulouse, 2007.
- [49] A. Montoya. *Méthodes de résolution pour le problème de tournées de véhicules avec fenêtres de temps et contraintes énergétiques*. Thèse de doctorat, Université d'Angers, École doctorale STIM, Laboratoire LARIS, 9 décembre 2016.

- [50] GALGOS Research Group. *Vehicle Routing Problem Research Platform*. Pontifical Catholic University of Rio de Janeiro, <http://vrp.galgos.inf.puc-rio.br/>, 2005-2023 [Consulté le JJ/MM/AAAA].