

FACULTE: Mathématique et informatique

DOMAINE: Mathématique et informatique

DEPARTEMENT: Informatique

FILIERE: Informatique

N°:.....

OPTION: Intelligence artificielle



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

**Par: Bouzidi Hamza**

Djaidja Mazin

**Intitulé**

**Contribution à la résolution du problème VRP  
multi-chemins avec incertitude**

**Soutenu devant le jury composé de:**

Mr. Mouhoub Nacer Eddine

Université de M'sila

Président

Mr. Sayad Lamri

Université de M'sila

Rapporteur

Mr. Dabba Ali

Université de M'sila

Examineur

**Année universitaire:2023/2024**

## **Remerciement**

Je remercie en premier lieu mon Dieu de nous donner la santé et la patience pour avoir terminé ce travail.

Ainsi je tiens également à exprimer mes vifs remerciements à notre encadreur Pr. Sayad Lamri pour avoir d'abord proposé ce thème. Pour suivi continuel tout long de la réalisation de ce mémoire et qui n'a pas cessé de me donner ses conseils et remarques.

Je remercie également tous les membres du jury d'avoir accepté d'examiner ce travail.

Ainsi que tous mes enseignants du Département mathématique et informatique.

Je veux également remercier ma famille pour leur soutien moral particulièrement mes parents pour leur soutien inconditionnel.

Je tiens également à remercier tous mes collègues de l'étude et particulièrement mon frère Hamza Bouzidi.

Enfin, j'adresse mes plus sincères remerciement à tous mes proches et amis qui m'ont toujours soutenue et encouragé au cours de la réalisation de ce mémoire.

## **DEDICACE**

**Je dédie ce modeste travail :**

A ma mère.

A mon père.

A Madiha.

A Dr. Billal Djaidja

A mes frères.

A tout la famille.

A tous mes amis.

A tous les enseignants de mathématique et informatique.

**Résumé:**

Le problème de la planification des itinéraires de véhicules avec des chemins multiples et des incertitudes est un défi majeur auquel de nombreuses entreprises de logistique et organisations sont confrontées dans la gestion de la distribution et du transport des marchandises. Ce problème concerne les difficultés liées à la détermination des meilleurs itinéraires pour les véhicules entre plusieurs emplacements tout en tenant compte de facteurs variables. Cette recherche a contribué à fournir des solutions efficaces au problème de la planification des itinéraires de véhicules avec des chemins multiples et des incertitudes. Un modèle de planification a été développé basé sur un algorithme génétique pour déterminer les chemins optimaux. Un site web a été développé pour fournir l'interface visuelle de ce modèle. Avec ce modèle et le site web associé, les entreprises de logistique peuvent améliorer l'efficacité de leurs processus opérationnels et réduire les coûts résultant des retards de livraison et de la sélection de routes inefficaces. Cette recherche représente une étape importante vers le développement de solutions innovantes dans le domaine de la gestion de la distribution, en s'appuyant sur des techniques modernes d'analyse de données et en améliorant la prise de décision logistique.

**Mots clés :** Problème de tournées de véhicules, itinéraires, chemins, distribution, modèle, algorithme génétique.

مشكلة توجيه المركبات بطرق متعددة وعدم اليقين هي تحدٍ كبير تواجهه العديد من شركات اللوجستيات والمنظمات في إدارة توزيع ونقل البضائع. تتعلق هذه المشكلة بالتحديات الناشئة عن تحديد أفضل الطرق للمركبات بين مواقع متعددة مع مراعاة العوامل المتغيرة. ساهم هذا البحث في توفير حلول فعالة لمشكلة توجيه المركبات بطرق متعددة. تم تطوير نموذج توجيه يعتمد على خوارزمية جينية لتحديد المسارات المثلى. تم تطوير موقع ويب لتوفير الواجهة البصرية لهذا النموذج. مع هذا النموذج والموقع الإلكتروني المرتبط به، يمكن لشركات اللوجستيات تحسين كفاءة عملياتها التشغيلية وتقليل التكاليف الناتجة عن تأخير التسليم واختيار الطرق غير الفعالة. يمثل هذا البحث خطوة مهمة نحو تطوير حلول مبتكرة في مجال إدارة التوزيع تعتمد على تقنيات تحليل البيانات الحديثة وتحسين اتخاذ القرارات اللوجستية.

**الكلمات الرئيسية:** مشكلة توجيه المركبات، طرق، مسارات، توزيع، نموذج، جينية.

## Table de matières

<b><u>TABLE DE MATIERES</u></b> .....	<b><u>5</u></b>
<b><u>LISTE DES FIGURES</u></b> .....	<b><u>8</u></b>
<b><u>LISTE DES TABLEAUX</u></b> .....	<b><u>10</u></b>
<b><u>INTRODUCTION GENERALE</u></b> .....	<b><u>11</u></b>
<b>CHAPITRE 01 : <u>PRÉSENTATION DU PROBLÈME DE TOURNÉE DE VÉHICULES (VRP)</u></b> .....	<b>12</b>
<b><u>1.1.INTRODUCTION :</u></b> .....	<b>13</b>
<b><u>1.2.DEFINITION DE PROBLEME DE TOURNEES DE VEHICULES (VRP) :</u></b> .....	<b>13</b>
<b><u>1.3.CHAMPS D'APPLICATION (VRP) :</u></b> .....	<b>13</b>
<b><u>1.4.PARAMETRES DU VRP :</u></b> .....	<b>14</b>
<b><u>1.4.1.RÉSEAU :</u></b> .....	<b>14</b>
<b><u>1.4.2.CLIENTÈLE :</u></b> .....	<b>14</b>
<b><u>1.4.3.FLOTTE DE VEHICULES :</u></b> .....	<b>14</b>
<b><u>1.4.4.FONCTION OBJECTIF :</u></b> .....	<b>15</b>
<b><u>1.5.VARIANTES DU VRP :</u></b> .....	<b>15</b>
<b><u>1.5.1.VRP A CONTRAINTES LIEES A LA FLOTTE DE VEHICULES :</u></b> .....	<b>15</b>
<b><u>1.5.1.1.VRP-C (Capacitated Vehicle Routing Problem) :</u></b> .....	<b>15</b>
<b><u>1.5.1.2.VRP-FL (Problème de Routage de Véhicules avec Chargement Complet) :</u></b> .....	<b>15</b>
<b><u>1.5.1.3.VRP-HF (PROBLEME DE ROUTAGE DE VEHICULES AVEC FLOTTE HETEROGENE) :</u></b> .....	<b>15</b>
<b><u>1.5.1.4.VRP-O (Problème de Routage de Véhicules Ouvert) :</u></b> .....	<b>15</b>
<b><u>1.5.1.5.VRP-B (Problème de Routage de Véhicules avec Retour) :</u></b> .....	<b>15</b>
<b><u>1.5.2.VRP A CONTRAINTES LIEES A LA DEMANDE DES CLIENTS :</u></b> .....	<b>16</b>
<b><u>1.5.2.1.VRP à Demande Déterministe :</u></b> .....	<b>16</b>
<b><u>1.5.2.2.VRP à Demande Stochastique :</u></b> .....	<b>16</b>
<b><u>1.5.3.VRP A CONTRAINTES LIEES AUX DEPOTS :</u></b> .....	<b>16</b>
<b><u>1.5.3.1.VRP-MD (Problème de Routage de Véhicules Multi-Dépôts) :</u></b> .....	<b>16</b>
<b><u>1.5.3.2.VRPP (Problème de Routage de Véhicules) :</u></b> .....	<b>16</b>
<b><u>1.5.4.VRP A CONTRAINTES LIEES AUX PRODUITS :</u></b> .....	<b>16</b>
<b><u>1.5.4.1.MP-VRP (Problème de Tournées de Véhicules à Produits Multiples) :</u></b> .....	<b>16</b>
<b><u>1.5.4.2.VRP (Problème de Tournées de Véhicules à un Seul Produit) :</u></b> .....	<b>16</b>
<b><u>1.5.5.VRP A CONTRAINTES LIEES AU TEMPS :</u></b> .....	<b>16</b>
<b><u>1.5.5.1.PVRP (Problème de Tournées de Véhicules Périodiques) :</u></b> .....	<b>16</b>
<b><u>1.5.5.2.VRP à Temps de Service Déterministe :</u></b> .....	<b>16</b>
<b><u>1.5.5.3.VRP à Temps de Service Stochastique :</u></b> .....	<b>16</b>
<b><u>1.5.5.4.VRP-TW (Problème de Tournées de Véhicules avec Fenêtres de Temps) :</u></b> .....	<b>16</b>
<b><u>1.5.6.PROBLEMES DE TOURNEES DE VEHICULES FREQUENTS :</u></b> .....	<b>17</b>
<b><u>1.5.6.1.VRP Statique :</u></b> .....	<b>17</b>
<b><u>1.5.6.2.DVRP (Problème de Routage de Véhicules Dynamique) :</u></b> .....	<b>17</b>
<b><u>1.5.6.3.SVRP (Problème de Routage de Véhicules Stochastique) :</u></b> .....	<b>17</b>

<b><u>1.6.FORMULATION MATHÉMATIQUE DU VRP :</u></b>	<b>18</b>
<b><u>1.6.1.FORMULATION MATHÉMATIQUE :</u></b>	<b>18</b>
<b><u>1.6.2.FORMULATION 1 :</u></b>	<b>19</b>
<b><u>1.6.3.FORMULATION 2 :</u></b>	<b>20</b>
<b><u>1.7.COMPLEXITE ALGORITHMIQUE :</u></b>	<b>22</b>
<b><u>1.8.CONCLUSION:</u></b>	<b>22</b>
<b>CHAPITRE 02:<u>LES MÉTHODES DE RÉOLUTION ET MÉTAHEURISTIQUE</u></b>	<b>23</b>
<b><u>2.1.INTRODUCTION:</u></b>	<b>24</b>
<b><u>2.2.LES METHODS EXACTES :</u></b>	<b>24</b>
<b><u>2.2.1.LA METHODE DE BRANCH AND BOUND :</u></b>	<b>25</b>
<b><u>2.2.2.LA METHODE DE BRANCH AND CUT :</u></b>	<b>25</b>
<b><u>2.2.3.LA PROGRAMMATION DYNAMIQUE:</u></b>	<b>25</b>
<b><u>2.2.4.AUTRES METHODS EXACTES:</u></b>	<b>25</b>
<b><u>2.3.LES METHODS APPROCHEES :</u></b>	<b>26</b>
<b><u>2.3.1.LES HEURISTIQUES :</u></b>	<b>26</b>
<b><u>2.3.1.1.L'heuristique « groupe en premier, route en second » :</u></b>	<b>26</b>
<b><u>2.3.1.2.L'heuristique « route en premier, groupe en second »:</u></b>	<b>26</b>
<b><u>2.3.1.3.L'algorithme en pétale :</u></b>	<b>26</b>
<b><u>2.3.2.MÉTAHEURISTIQUES :</u></b>	<b>27</b>
<b><u>2.3.2.1.Le recuit simulé (simulated annealing):</u></b>	<b>27</b>
<b><u>2.3.2.2.La recherche Tabou (Tabu Search):</u></b>	<b>29</b>
<b><u>2.3.2.3.Les algorithmes génétiques:</u></b>	<b>30</b>
<b><u>2.3.2.4.Les algorithmes de colonies de fourmis:</u></b>	<b>30</b>
<b><u>2.3.2.4.1.Principe de l'algorithme:</u></b>	<b>32</b>
<b><u>2.3.2.4.2.Mise à jour de phéromones:</u></b>	<b>32</b>
<b><u>2.3.2.5.L'optimisation par essaim de particules:</u></b>	<b>33</b>
<b><u>2.3.2.5.1.Principe général:</u></b>	<b>33</b>
<b><u>2.3.2.5.2.Algorithme de base:</u></b>	<b>34</b>
<b><u>2.3.2.6.La recherche dispersée:</u></b>	<b>35</b>
<b><u>2.3.2.7.Description générale de la recherche dispersée:</u></b>	<b>35</b>
<b><u>2.4.CONCLUSION :</u></b>	<b>36</b>
<b>CHAPITRE 03:<u>UN ALGORITHME DE GÉNÉTIQUE APPLIQUÉES AU PROBLÈME DU VRP</u></b>	<b>37</b>
<b><u>3.1.INTRODUCTION:</u></b>	<b>38</b>
<b><u>3.2.RESOLUTION INTERACTIVE DE PROBLÈMES D'OPTIMISATION (VRP):</u></b>	<b>38</b>
<b><u>3.3.UN ALGORITHME GÉNÉTIQUE POUR LE VRP:</u></b>	<b>38</b>
<b><u>3.3.1.CODAGE:</u></b>	<b>39</b>
<b><u>3.3.2.OPÉRATEURS GÉNÉTIQUES:</u></b>	<b>41</b>
<b><u>3.3.2.1.Procédure de sélection:</u></b>	<b>41</b>

<a href="#"><u>3.3.2.1.1.La sélection par roulette:</u></a>	41
<a href="#"><u>3.2.1.2.La sélection par rang:</u></a>	41
<a href="#"><u>3.3.2.2.Opérateur de croisement :</u></a>	42
<a href="#"><u>3.3.2.3.Opérateur de mutation :</u></a>	46
<a href="#"><u>3.3.2.4.Population initiale:</u></a>	46
<a href="#"><u>3.3.2.5.Fitness:</u></a>	46
<a href="#"><u>3.4.CONCLUSION :</u></a>	48
<b>CHAPITRE 04:<a href="#"><u>VISUALISATION DES RESULTATS DE</u></a> ALGORITHME GENETIQUE.</b>	<b>49</b>
<a href="#"><u>4.1.INTRODUCTION</u></a>	50
<a href="#"><u>4.2.UML DIAGRAMMES DE NOTRE APPLICATION WEB :</u></a>	50
<a href="#"><u>4.3.OUTILS ET TECHNOLOGIES UTILISES POUR NOTRE IMPLEMENTATION:</u></a>	51
<a href="#"><u>4.3.1.L'API GOOGLE MAPS :</u></a>	51
<a href="#"><u>4.3.2.JQUERY:</u></a>	52
<a href="#"><u>4.3.3.SERVLET :</u></a>	53
<a href="#"><u>4.3.4.MY SQL:</u></a>	54
<a href="#"><u>4.3.5.WAMP:</u></a>	54
<a href="#"><u>4.3.6.NETBEANS IDE:</u></a>	54
<a href="#"><u>4.4.PRESENTATION DU SYSTEME:</u></a>	55
<a href="#"><u>4.4.1.APPLICATION WEB:</u></a>	55
<a href="#"><u>4.4.1.1.Pour le client (Client):</u></a>	55
<a href="#"><u>4.4.1.2.Pour l'administrateur :</u></a>	57
<a href="#"><u>4.5.CONCLUSION:</u></a>	58
<a href="#"><u>CONCLUSION GENERALE:</u></a>	59

## Liste des Figures

<b>Figure</b>	<b>Titre</b>	<b>Page</b>
<b>1.</b>	problème de tournée de véhicule	<b>3</b>
<b>2.</b>	Variantes du VRP	<b>8</b>
<b>3.</b>	Fonctionnement de l'algorithme de recuit simulé	<b>20</b>
<b>4.</b>	Fonctionnement d'un algorithme génétique	<b>22</b>
<b>5.</b>	L'expérience du pont à double branche	<b>23</b>
<b>6.</b>	Analysis of Particle Trajectories	<b>26</b>
<b>7.</b>	Vrp ayant n=20 clients	<b>31</b>
<b>8.</b>	Deux solutions utilisées comme parents pour illustrer l'opérateur de croisement	<b>35</b>
<b>9.</b>	croisement de Baker and Ayechew	<b>35</b>
<b>10.</b>	Enfant E2 généré par le croisement des parents de la figure 10	<b>36</b>
<b>11.</b>	croisement des chromosomes 1	<b>36</b>
<b>12.</b>	croisement des chromosomes 2	<b>37</b>
<b>13.</b>	croisement des chromosomes 3	<b>37</b>
<b>14.</b>	Diagramme de cas d'utilisation	<b>42</b>
<b>15.</b>	Diagramme de classe	<b>43</b>
<b>16.</b>	le logo de google maps	<b>43</b>
<b>17.</b>	le logo jquery	<b>44</b>
<b>18.</b>	Explication de servlet	<b>45</b>
<b>19.</b>	le logo MY SQL	<b>46</b>
<b>20.</b>	le logo eclipse	<b>47</b>
<b>21.</b>	interface de netbeans	<b>48</b>
<b>22.</b>	Interface de site web	<b>49</b>
<b>23.</b>	Inscription et connexion	<b>49</b>
<b>24.</b>	La barre de menu	<b>49</b>
<b>25.</b>	Réglages de Mes Boutiques("My shops")	<b>50</b>
<b>26.</b>	Formulaire d'ajout de magasin("Add shop")	<b>50</b>

<b>27.</b>	Réglages de reservation	<b>51</b>
<b>28.</b>	Interface du panneau de contrôleadministrateur	<b>51</b>
<b>29.</b>	Barre de menu du panneau de contrôle	<b>52</b>
<b>30.</b>	Exemple de livraison	<b>52</b>

## **Liste des tableaux**

<b>Tableau</b>	<b>Titre</b>	<b>Page</b>
<b>1.</b>	tableau de chromosome initial	<b>32</b>
<b>2.</b>	tableau de deuxième chromosome	<b>32</b>
<b>3.</b>	tableau de troisième chromosome	<b>33</b>
<b>4.</b>	tableau de quatrième chromosome	<b>36</b>

Le problème de tournées de véhicules (VRP : Vehicle Routing Problem ) a été le sujet d'une recherche intensive durant plus de cinquante ans, liée à son importance dans le domaine de la logistique, et à sa grande difficulté. Le VRP implique la planification de routes de livraison à moindre coût, afin de servir un ensemble de clients dispersés géographiquement, tout en respectant les contraintes de capacité des véhicules. Ce problème d'optimisation combinatoire présente à la fois des caractéristiques d'allocation de ressources (répartition des charges dans les camions: bin packing ), et de construction de séquence (problème du voyageur de commerce). Les applications pratiques, sont souvent de grande taille et doivent actuellement être résolues heuristiquement.

Dans un monde où les grands acteurs du secteur industriel gèrent des systèmes de création de valeur toujours plus grands, la qualité de la prise de décision est largement liée à l'habileté à résoudre des problèmes combinés complexes. Cette considération mène à des problèmes de VRP agrémentés de caractéristiques ou contraintes supplémentaires, dédiées à des choix tactiques comme les problèmes de routage sur un horizon temporel étendu ou à une plus grande fidélité à la réalité contraintes législatives sur les temps de conduite, congestion des axes de transport, ou enfin placement explicite des objets dans le chargement. Ces caractéristiques combinées au problème de VRP, parfois appelées attributs, forment un vaste domaine de la littérature, comprenant des centaines voire des milliers de contributions spécialisées à des attributs particuliers.

Un grand nombre d'heuristiques ont ainsi été développées pour ces divers problèmes, mais cependant, très peu de méthodes sont aujourd'hui à même de résoudre de façon généraliste un ensemble relativement large de variantes du VRP. Cette littérature, qui comprend des problèmes et méthodes de résolution hétéroclites, est particulièrement difficile à classifier. La contribution de cette revue est de proposer une analyse objective des méthodes heuristiques, concepts à succès et défis pour les problèmes de VRP multi-attributs. Devant l'ampleur du domaine, une démarche exhaustive est impensable aujourd'hui, si bien qu'une analyse synthétique est nécessaire.

Nous avons relevé et classifié les variantes du VRP, reconnues pour être le support d'une recherche soutenue, et comprenant un nombre conséquent de méthodes heuristiques, des jeux de tests communs, et éventuellement des revues de littérature spécialisées.

Ce mémoire constitue des étapes importantes pour résoudre problème de tournées de véhicules. La structure de cette mémoire est la suivante: nous introduirons dans un premier temps en chapitre 1 le VRP "classique".

Le chapitre 2 introduit les méthodes de résolution et Metaheuristiques. Dans le chapitre 03, nous détaillons un algorithme génétique pour résoudre le problème en fin le chapitre 04 présente implémentation de l'algorithme génétique et afficher des résultats. Ainsi, que des conclusions.

## **CHAPITRE 01**

# **PRÉSENTATION DU PROBLÈME DE TOURNÉE DE VÉHICULES (VRP)**

## 1.1.Introduction :

La plupart des entreprises de distribution sont confrontées au défi de la livraison et de la collecte des colis pour satisfaire leurs clients. Cette activité engendre un problème plus important : comment distribuer les véhicules de manière à livrer les marchandises aux clients tout en collectant d'autres et en les stockant dans les entrepôts, tout en tenant compte du volume et du temps nécessaires à cet effet. Dans ce chapitre, nous aborderons le problème de la distribution des véhicules, en exposant brièvement ses tenants et aboutissants, ainsi qu'en fournissant une description précise de ses composantes et de certaines variations entre ses formulations mathématiques. Enfin, nous présenterons la complexité algorithmique des problèmes de distribution de véhicules classiques (VRP) à la fin du chapitre.

## 1.2.Définition de Problème de Tournées de Véhicules (VRP) :

Selon le problème de tournées de véhicules (Véhicule Routing Problém) est une extension du Problème du Voyageur de Commerce (TSP) où  $K$  véhicules ayant une capacité  $Q$  sont disponibles au point de départ. Chaque véhicule doit réaliser une tournée viable en quittant le point de départ, en visitant les clients une fois, avec une somme des demandes ne dépassant pas la capacité  $Q$ , avant de retourner au point de départ. L'objectif est de trouver des tournées de coût minimal permettant de livrer tous les clients tout en respectant les contraintes de capacité des véhicules [1].

La planification des routes de livraison à moindre coût, telle que le VRP, est une démarche visant à desservir efficacement un ensemble de clients dispersés géographiquement tout en tenant compte des limitations de capacité des véhicules. Différentes variantes des problèmes de tournées sont développées pour répondre aux objectifs spécifiques et aux contraintes opérationnelles des applications concernées[2].

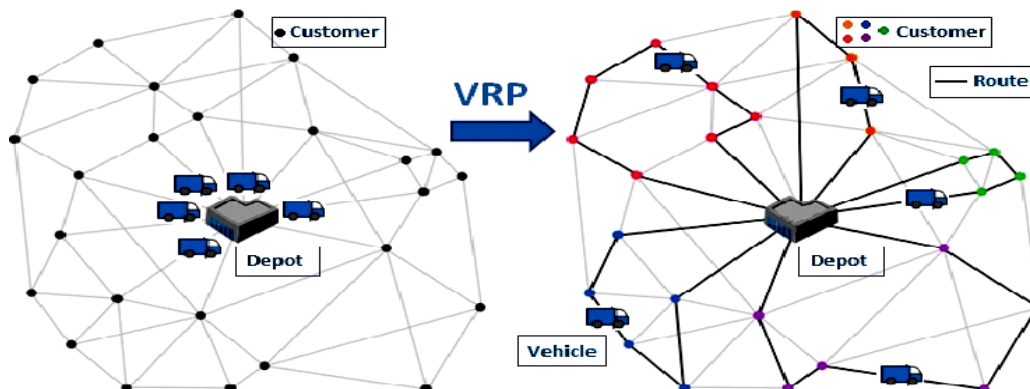


Figure.1:problème de tournée de véhicule [3].

## 1.3.Champs d'Application (VRP) :

Chaque entreprise industrielle s'efforce d'améliorer la qualité de ses services en résolvant le problème de la distribution des véhicules, afin de produire des biens et de fournir ses services au coût le plus bas possible et de garantir un flux fluide de ses marchandises. Ce problème occupe une place importante dans notre vie quotidienne, que ce soit pour le ramassage des élèves, le transport du personnel, la collecte des déchets ménagers, la distribution du courrier et des produits alimentaires, ainsi que pour la fourniture de services

médicaux mobiles et la distribution de médicaments. L'objectif n'est pas de réduire la distance parcourue, mais de servir un maximum de clients.

Dans le domaine de l'informatique et des robots, ce problème est largement utilisé pour le routage des informations à distance ou via des câbles, ainsi que dans la planification des vols aériens.

Malgré l'intérêt que l'homme porte à ce problème, il reste encore incapable de résoudre avec précision certaines de ses instances, en raison de la complexité de certaines situations, voire de l'impossibilité de les résoudre même avec l'aide d'un ordinateur, quel que soit sa puissance.

#### **1.4. Paramètres du VRP :**

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

##### **1.4.1. Réseau :**

Le réseau de transport permet le flux des individus, des marchandises et des informations, et représente l'infrastructure de communication de base dans le système. Il peut être représenté graphiquement sous forme de nœuds représentant les clients et d'arêtes représentant les chemins entre eux, parfois orientés.

##### **1.4.2. Clientèle :**

- ❖ **Demande:** Le client peut demander un service ou des produits, qui peuvent être de différents types.
- ❖ **Capacité maximale de la commande:** La somme totale des demandes des clients dans une tournée ne doit pas dépasser la capacité du véhicule.
- ❖ **Position géographique:** chaque client a une position géographique unique, ce qui influence la planification des tournées et le choix des itinéraires.
- ❖ **Demande déterminée ou incertaine:** La demande du client peut être soit spécifique avec une quantité fixe et connue, soit incertaine et sujette à des variations aléatoires.

##### **1.4.3. Flotte de véhicules :**

- ✓ **Taille:** Le nombre de véhicules dans la flotte.
- ✓ **Homogénéité :** Si les véhicules ont la même capacité de chargement et le même coût de transport.
- ✓ **Hétérogénéité :** Si les véhicules ont des capacités de chargement et/ou des coûts de transport différents.

#### **1.4.4. Fonction objectif :**

Principaux objectifs de la planification des tournées :

- Réduire le nombre de véhicules ou la distance totale.
- Réduire le temps total de la tournée et les retards.
- Minimiser les coûts totaux.
- Maximiser le profit et la qualité du service.
- Optimiser le chargement des véhicules.

#### **Remarque (1.1):**

L'objectif en matière de planification des tournées peut varier entre l'orientation des ressources telles que la réduction du nombre de véhicules ou l'utilisation maximale de la capacité des véhicules, et l'orientation des tournées elles-mêmes en réduisant leur durée, leur temps ou leur coût. Il peut également être axé sur l'amélioration des performances des arcs et des nœuds en réduisant le temps d'attente ou en augmentant la satisfaction client. Dans la pratique, le problème du VRP est plus complexe car il faut souvent tenir compte de plusieurs objectifs contradictoires. Par exemple, concilier la réduction du nombre de véhicules avec l'augmentation de la distance totale parcourue peut être un défi.

#### **1.5. Variantes du VRP :**

La variété des variantes du Problème de Routage de Véhicules (VRP) qui peuvent être définies en modifiant, supprimant ou ajoutant des contraintes au problème classique. Ces variantes sont ensuite classées selon le type de contraintes qui les caractérisent, notamment les contraintes de capacité, de temps, géographiques, de ressources, environnementales, ainsi que d'autres variantes telles que le VRP stochastique, le VRP avec collecte et livraison, et le VRP avec tournées périodiques.

##### **1.5.1. VRP à Contraintes Liées à la Flotte de Véhicules :**

###### **1.5.1.1. VRP-C (Capacitated Vehicle Routing Problem) :**

###### **VRP avec contraintes de capacité :**

Les véhicules sont tenus de respecter les contraintes de capacité de charge, où la somme des demandes dans une tournée ne doit pas dépasser la capacité du véhicule.

###### **1.5.1.2. VRP-FL (Problème de Routage de Véhicules avec Chargement Complet) :**

Visé à utiliser les camions à pleine capacité de charge.

###### **1.5.1.3. VRP-HF (Problème de Routage de Véhicules avec Flotte Hétérogène) :**

Composé d'une flotte de véhicules variés en termes de capacité, de puissance et de coût de transport.

###### **1.5.1.4. VRP-O (Problème de Routage de Véhicules Ouvert) :**

Permet au véhicule de rejoindre ou non le dépôt après la fin de la tournée, et s'il choisit de le rejoindre, il doit reprendre la tournée dans le sens inverse.

###### **1.5.1.5. VRP-B (Problème de Routage de Véhicules avec Retour) :**

Exige que le véhicule retourne au dépôt dès que le dernier client est servi, sans reprendre la tournée.

## **1.5.2. VRP à Contraintes Liées à la Demande des Clients :**

### **1.5.2.1. VRP à Demande Déterministe :**

Un problème courant pour les entreprises qui offrent des services de livraison à la demande, où le chauffeur connaît à l'avance la quantité de livraison pour chaque client.

### **1.5.2.2. VRP à Demande Stochastique :**

Dans ce problème, le chauffeur ne connaît pas la quantité de livraison pour chaque client avant de le servir, et il estime approximativement la demande en utilisant une fonction stochastique.

**1.5.2.3. S-VRP (Problème de Routage de Véhicules avec Livraisons Fractionnées) :** Ce problème implique de visiter un client plusieurs fois pour répondre pleinement à sa demande, avec la possibilité que la demande du client dépasse la capacité du véhicule.

**1.5.2.4. VRP-PD (Problème de Routage de Véhicules avec Prélèvement et Livraisons) :** Il s'agit d'un problème de tournées de véhicules impliquant à la fois des opérations de prélèvement et de livraison. La durée de service est comptée deux fois en raison de la nécessité de collecter et de livrer.

## **1.5.3. VRP à Contraintes Liées aux Dépôts :**

### **1.5.3.1. VRP-MD (Problème de Routage de Véhicules Multi-Dépôts) :**

Les véhicules peuvent se ravitailler à partir de plusieurs dépôts différents.

### **1.5.3.2. VRPP (Problème de Routage de Véhicules) :**

Les véhicules doivent se ravitailler à partir d'un seul dépôt.

Il est possible de passer du VRP-MD au VRPP ou inversement en centralisant ou en divisant les dépôts respectivement.

## **1.5.4. VRP à Contraintes Liées aux Produits :**

### **1.5.4.1. MP-VRP (Problème de Tournées de Véhicules à Produits Multiples) :**

Les véhicules doivent livrer une variété de produits différents à divers clients au cours d'une seule tournée.

### **1.5.4.2. VRP (Problème de Tournées de Véhicules à un Seul Produit) :**

Chaque véhicule doit livrer un seul produit à différents clients au cours d'une seule tournée.

## **1.5.5. VRP à Contraintes Liées au Temps :**

### **1.5.5.1. PVRP (Problème de Tournées de Véhicules Périodiques) :**

Chaque client est visité périodiquement selon une planification prédéfinie.

### **1.5.5.2. VRP à Temps de Service Déterministe :**

La durée du service est connue par le livreur avant de commencer la tournée.

### **1.5.5.3. VRP à Temps de Service Stochastique :**

Le livreur ne connaît pas la durée du service des clients à l'avance, mais l'estime au moment de les servir en utilisant une fonction stochastique.

### **1.5.5.4. VRP-TW (Problème de Tournées de Véhicules avec Fenêtres de Temps) :**

Chaque client doit être servi dans une fenêtre de temps prédéfinie, et tout écart par rapport à cette contrainte peut entraîner une pénalité. On distingue entre le VRPTW rigide, où le respect de la fenêtre de temps est impératif, et le VRPTW relâché, où le retard ou l'avance ne génère qu'une pénalité.

### **1.5.6. Problèmes de Tournées de Véhicules Fréquents :**

#### **1.5.6.1. VRP Statique :**

Toutes les composantes du problème sont fixes et connues avant le début de toute tournée. Cela peut être dû à une étude rigoureuse des paramètres ou parce que le problème modélise un phénomène statique à l'origine.

#### **1.5.6.2. DVRP (Problème de Routage de Véhicules Dynamique) :**

Contrairement au VRP statique, le DVRP comporte au moins un élément dynamique ou changeant pendant son exécution. Cela peut inclure des variations dans la demande des clients, les fenêtres de temps de service ou le nombre de clients à desservir, ce qui rend le problème plus complexe.

#### **1.5.6.3. SVRP (Problème de Routage de Véhicules Stochastique) :**

Le SVRP est une sous-catégorie des problèmes dynamiques où les variations suivent une loi probabiliste. Cela permet une modélisation mathématique de la variation, facilitant ainsi sa résolution.

#### **Remarque (1.2) :**

En réalité, un problème peut combiner plusieurs contraintes. Par exemple, une contrainte de capacité et de fenêtre de temps, ou encore une contrainte de multiples dépôts avec des opérations de collecte et de livraison. Cette combinaison de contraintes rend le problème plus représentatif de la réalité mais également plus difficile à résoudre.

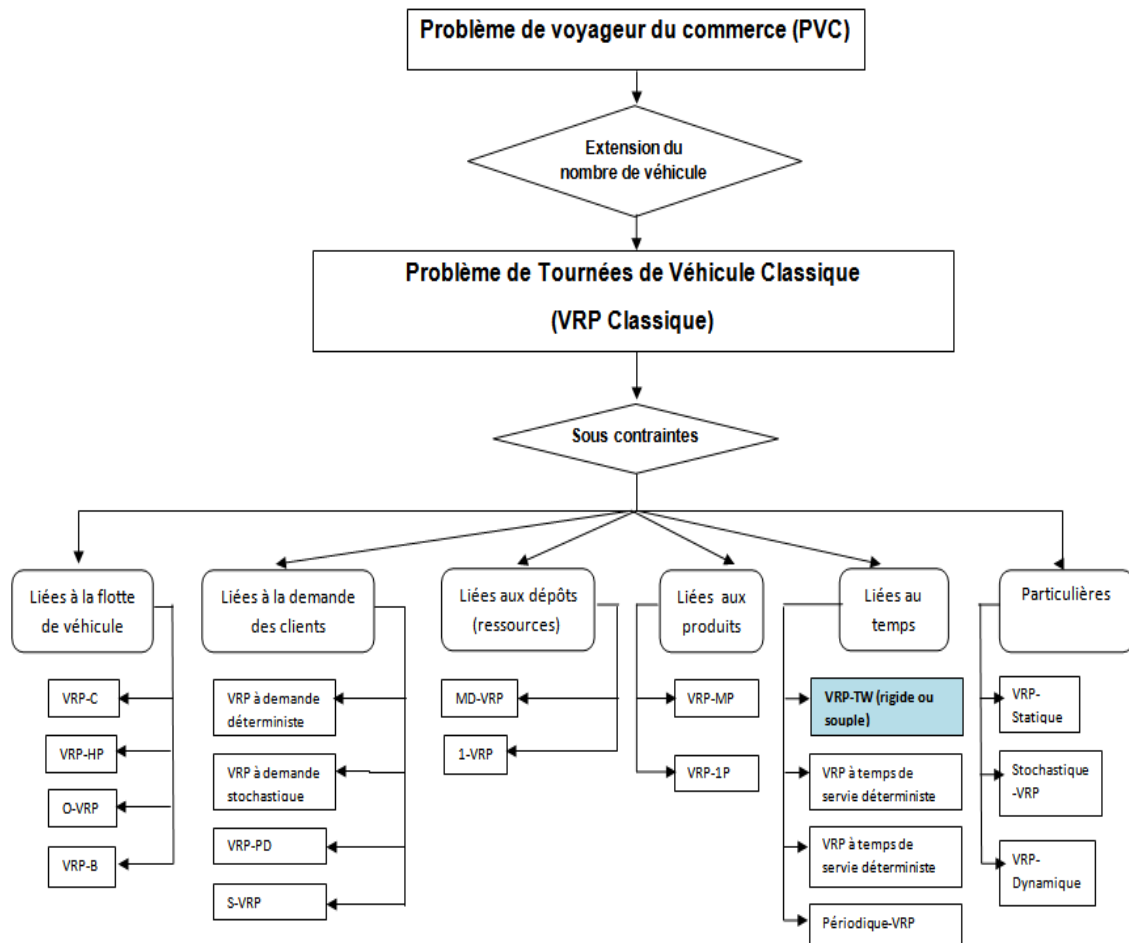


Figure.2 : Variantes du VRP .

## 1.6. Formulation mathématique du VRP :

La littérature scientifique regorge de nombreuses formulations du VRP. Ce qui les lie toutes, c'est la représentation du problème de tournées de véhicules sous forme d'un graphe orienté

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

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

### 1.6.1. Formulation mathématique :

Dans la littérature, on trouve de nombreuses formulations mathématiques du VRP classique. Parmi celles-ci, les deux formulations suivantes sont parmi les plus courantes.

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

$n$ : nombre de clients.

$m$ : nombre de véhicules ( $m \leq n$ ).

$Q_k$ : capacité du  $k^{\text{ème}}$  véhicule.

$\mathcal{T}_k$ : temps maximal de la tournée du  $k^{\text{ème}}$  véhicule.

$as_j$ : nombre d'arcs sortants du sommet  $j$ .

$ae_j$ : nombre d'arcs entrants du sommet  $j$ .

$q_j$ : demande du  $j^{\text{ème}}$  client  $d_o = 0$ .

$s_k^j$ : durée du service du  $j^{\text{ème}}$  client par le  $k^{\text{ème}}$  véhicule ( $s_o^k = 0$ ).

$t_{ij}^k$ : temps nécessaire pour passer du  $i^{\text{ème}}$  client au  $j^{\text{ème}}$  client ( $t_{jj}^k = +\infty$ ).

$d_{ij}$ : distance séparant  $i^{\text{ème}}$  client du  $j^{\text{ème}}$  client ( $d_{jj}^k = +\infty$ ).

$C_{ij}$ : le coût du déplacement du  $i^{\text{ème}}$  client au  $j^{\text{ème}}$  client.

### 1.6.2. Formulation 1 :

La formulation mathématique du VRP classique la plus communément utilisée dans la littérature est celle adoptée par Laporte en 1992, Rego and Roucairol en 1994, Toth and Vigo a 2001, Crainic and Semet a 2006. En effet, elle nécessite la définition de  $n \times n$  variables de décision de type binaire, à trois indices, suivantes :

$$x_{ij}^k \begin{cases} 1 \text{ si l'arc } (i, j) \text{ est parcouru par } k^{\text{ème}} \text{ véhicule} \\ 0 \text{ sinon} \end{cases} \quad (1.1)$$

Autrement dit :

$$x_{ij}^k \in \{0,1\} ; i = 0, \dots, n ; j = 0, \dots, n ; k = 0, \dots, m \quad (1.2)$$

Le problème se modélise comme suit :

$$\text{Minimiser } \sum_{i=1}^n \sum_{j=1}^n C_{ij} \sum_{k=1}^m x_{ij}^k \quad (1.3)$$

Sous

$$\sum_{i=1}^n \sum_{k=1}^m x_{ij}^k = 1 ; j = 2, \dots, n \quad (1.4)$$

$$\sum_{j=1}^n \sum_{k=1}^m x_{ij}^k = 1; i = 1, \dots, n \quad (1.5)$$

$$\sum_{i=1}^n x_{ip}^k - \sum_{j=1}^n x_{pj}^k = 0; k = 1, \dots, m; p = 1, \dots, n \quad (1.6)$$

$$\sum_{j=1}^n q_i \left( \sum_{k=1}^m x_{ij}^k \right) \leq Q_k; k = 1, \dots, m \quad (1.7)$$

$$\sum_{i=1}^n s_i^k x_{ij}^k + \sum_{i=1}^n \sum_{j=1}^n t_{ij}^k x_{ij}^k \leq T_k; k = 1, \dots, m \quad (1.8)$$

$$\sum_{j=1}^n x_{0j}^k \leq 1; k = 1, \dots, m \quad (1.9)$$

$$\sum_{i=1}^n x_{i0}^k \leq 1; k = 1, \dots, m \quad (1.10)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij}^k \leq |S| - 1; \text{pour tout } S \in N; \text{et } 2 \leq |S| \leq (n - 1). \quad (1.11)$$

La formule (1.3) représente la fonction à optimiser ou la fonction objectif du VRP classique. Généralement, l'objectif est de trouver le minimum du coût global des tournées.

Les formules de (1.4) à (1.18) représentent les contraintes du problème : Les contraintes (1.4) et (1.5) assurent que chaque client n'est servi qu'une et seule fois et par un et un seul véhicule. La contrainte (1.6) assure la continuité de la tournée : non seulement un véhicule doit passer une et une seule fois chez un client (1.4) et (1.5)

Mais il doit impérativement le quitter une fois le service est achevé. La contrainte (1.7) assure que la capacité du véhicule ne sera pas dépassée. La contrainte (1.8) assure que la durée totale d'une tournée ne dépassera jamais sa durée totale maximale. Les contraintes (1.9) et (1.10) assurent le non dépassement de la disponibilité de chacun des véhicules, un véhicule ne sort du dépôt et n'y revient qu'une seule fois. La dernière contrainte (1.11) garantit l'élimination de sous-tours[4].

### 1.6.3. Formulation 2 :

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

:

$$x_{ij}^k \begin{cases} 1 \text{ si l'arc } (ij) \text{ est parcouru par } k^{\text{eme}} \text{ véhicule} \\ 0 \text{ sinon} \end{cases} \quad (1.12)$$

$$y_{ij}^k \begin{cases} 1 \text{ si } i^{\text{eme}} \text{ client est servi par } k^{\text{eme}} \text{ véhicule} \\ 0 \text{ sinon} \end{cases} \quad (1.13)$$

La formule précédente est équivalente à :

$$y_i^k \in \{0, 1\}; i = 1, \dots, n; k = 1, \dots, m \quad (1.14)$$

Le problème se modélise comme suit :

$$\text{Minimiser } \sum_{i=1}^n \sum_{j=1}^n C_{ij} \sum_{k=1}^m k_{ij} \quad (1.15)$$

Sous :

$$\sum_{k=1}^n y_{ik} = 1; i = 1, \dots, n \quad (1.16)$$

$$\sum_{k=1}^m y_{0k} = m \quad (1.17)$$

$$\sum_{i=1}^n q_i y_{ik} \leq Q_k; k = 1, \dots, m \quad (1.18)$$

$$\sum_{j=1}^n x_{ij}^k = \sum_{k=1}^m x_{ij}^k = y_{ik}; k = 1, \dots, m; i = 1, \dots, n \quad (1.19)$$

$$\sum_{i \in S} \sum_{i \in S} x_{ij}^k \leq |S| - 1; S \in \{2, \dots, n\} \text{ et } j = 1, \dots, m \quad (1.20)$$

La formule en (1.15) est équivalente à la formule (1.10), elle représente la fonction à optimiser. Les formules de (1.16) à (1.20) représentent les contraintes du problème. La contrainte (1.16) est équivalente aux contraintes (1.11) elle assure que chaque client n'est servi qu'une et seule fois et par un et un seul véhicule. La contrainte (1.17) assure que tous les véhicules se trouvent au dépôt. La contrainte (1.18) assure que la capacité de chacun des véhicules ne soit pas dépassée. La contrainte (1.19) permet de lier les deux types de

contraintes d'une part, et d'autre part, elle assure que le véhicule en quitte le client aussitôt que ce dernier ait été servi. La dernière contrainte (1.20) garantit l'élimination de sous-tour[4].

**Remarque (1.3) :**

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

**1.7. Complexité algorithmique :**

La complexité d'un problème se réfère à l'estimation, dans le pire des cas, du nombre d'instructions nécessaires pour résoudre une instance du problème. La complexité algorithmique mesure l'efficacité d'un algorithme de résolution d'un problème, où plus le nombre d'instructions est élevé, plus le problème est complexe. Il existe plusieurs types de complexité, y compris la complexité constante ( $o(1)$ ), logarithmique ( $o(\log(n))$ ), linéaire ( $o(n)$ ), quadratique ( $o(n^2)$ ), cubique ( $o(n^3)$ ), polynomiale ( $o(n^p)$ ), exponentielle ( $o(\exp(n))$ ), et factorielle ( $o(n!)$ ).

En fonction du critère de complexité, quatre classes de problèmes principales sont distinguées :

- **Classe P** : Problèmes pouvant être résolus en temps polynomial par une machine de Turing. Ils sont appelés problèmes polynomiaux ou Classe P.
- **Classe P-Complet** : Ensemble des problèmes polynomiaux et de certains problèmes non polynomiaux qui peuvent être réduits en temps poly-logarithmique avec ou sans l'utilisation d'une machine de calcul déterministe.
- **Classe NP** : Ensemble des problèmes pouvant être résolus en temps polynomial sur une machine de Turing.
- **Classe NP-DUR** : Aussi appelée NP-Difficile. Elle regroupe les problèmes dont la complexité est exponentielle ou doublement exponentielle, et pour lesquels la résolution optimale ou exacte des grandes instances est difficile en temps polynomial. Les solutions de qualité peuvent être trouvées en temps polynomial.

Le problème de tournées de véhicules (VRP) est classé comme NP-DUR. Sa résolution exacte est inappropriée pour les grandes instances, et il est généralement résolu par des approches heuristiques qui fournissent des solutions réalisables en un temps raisonnable.

**1.8. Conclusion:**

Dans ce chapitre, nous avons introduit le Problème de Tournées de Véhicules (VRP), en commençant par présenter ses domaines d'application et sa formulation mathématique, puis en énumérant ses variantes. Cependant, les problèmes de tournées de véhicules appartiennent à la classe NP difficile. Leur résolution ne peut être réalisée que de manière approximative, car à ce jour, il n'existe aucun algorithme permettant de les résoudre en temps polynomial. Les approches heuristiques qui seront développées dans le prochain chapitre constituent une alternative pour la résolution de ce type de problème.

## **CHAPITRE02**

# **LES MÉTHODES DE RÉOLUTION ET MÉTAHEURISTIQUE**

## **2.1.Introduction:**

Les méthodes exactes garantissent une solution optimale dans certains cas, mais parfois, on privilégie des solutions de bonne qualité avec un temps de calcul réduit. C'est là qu'interviennent les méta-heuristiques, adaptées à chaque problème mais sans garantie de qualité de solution. Elles exploitent des processus aléatoires pour explorer l'espace de recherche, évitant ainsi l'explosion combinatoire. Les méta-heuristiques sont souvent itératives, répétant le même processus pour résoudre le problème. Leur principal avantage est d'éviter les minima locaux, même au prix d'une dégradation de la fonction objectif.

L'optimisation combinatoire est cruciale en recherche opérationnelle et en informatique. De nombreux problèmes peuvent être modélisés sous forme d'optimisation combinatoire, comme le problème du voyageur de commerce, l'ordonnancement de tâches et la coloration de graphes. Ces problèmes impliquent un ensemble fini de solutions, chaque solution devant satisfaire des contraintes spécifiques et être évaluée par une fonction objectif. La solution optimale est celle qui minimise (ou maximise) cette fonction objectif parmi toutes les solutions possibles. Cela fait de l'optimisation combinatoire un domaine crucial pour de nombreuses applications.

La résolution des problèmes d'optimisation combinatoire est assez délicate car le nombre fini de solutions réalisables augmente généralement avec la taille et la complexité du problème. Cela a conduit les chercheurs à développer de nombreuses méthodes de résolution en recherche opérationnelle et en intelligence artificielle. Ces approches de résolution peuvent être classées en deux catégories : les méthodes exactes et les méthodes approchées.

Les techniques exactes sont efficaces pour résoudre des problèmes de taille raisonnable, mais elles rencontrent souvent des difficultés avec des problèmes de grande taille. En revanche, les méthodes d'approximation ne garantissent pas une solution exacte, mais seulement une approximation.

## **2.2. Les méthodes exactes :**

Les méthodes exactes reposent sur l'utilisation d'algorithmes qui mènent de façon sûre vers la solution optimale. Le principe essentiel de ces méthodes est d'énumérer de manière implicite l'ensemble des solutions de l'espace de recherche. Malgré l'important temps de Calcul que nécessitent, généralement, ces approches, plusieurs méthodes ont été développées.

Elles permettent de résoudre efficacement des problèmes allant jusqu'à 50 clients. En 20 Une méthode résolvant un problème contenant 100 clients a été proposée dans [5] .

Parmi ces méthodes on peut citer :

### **2.2.1. La méthode de Branch and Bound :**

La méthode de branch and bound consiste à planifier intelligemment les solutions, en éliminant les solutions partielles qui ne mènent pas à la solution recherchée. Son exécution est représentée par un arbre, où la racine représente l'ensemble de toutes les solutions possibles. Elle commence par évaluer le problème initial et appliquer des bornes inférieures et supérieures à la racine. Si une solution optimale est trouvée, le processus s'arrête. Sinon, l'ensemble des solutions est divisé en sous-ensembles.

La méthode est ensuite appliquée de manière récursive à ces sous-problèmes, ce qui crée une arborescence. Si une solution optimale est trouvée pour un sous-problème, elle est réalisable mais pas nécessairement optimale pour le problème de départ. Comme elle est réalisable, elle peut être utilisée pour éliminer toute sa descendance : si la borne inférieure d'un nœud dépasse la valeur d'une solution déjà connue, alors on peut affirmer que la solution optimale globale ne peut être contenue dans le sous-ensemble de solution représenté par ce nœud. La recherche se poursuit jusqu'à ce que tous les nœuds soient explorés ou éliminés.

Cette technique donne de bons résultats pour les problèmes d'ordonnancement de petite taille, mais dans le cas contraire, elle risque de générer des branches très étendues.

### **2.2.2. La méthode de branch and cut :**

La méthode est connue sous le nom de méthode de programmation en nombres entiers. Comme toute méthode énumérative implicite, l'algorithme construit une arborescence appelée l'arbre du "branch and cut", où les sous-problèmes sont des nœuds. Il existe trois types de nœuds dans l'arbre "branch and cut" : le nœud courant en cours de traitement, les nœuds actifs en attente de résolution, et les nœuds inactifs qui ont été élagués pendant l'exécution de l'algorithme. Le principe est de partir d'une solution entière admissible du problème et, à l'aide d'algorithmes tels que le simplexe, d'atteindre progressivement une autre solution entière admissible jusqu'à l'optimum.

### **2.2.3. La programmation dynamique:**

Cette méthode se base sur le principe de Bellman : « Si C'est un point qui appartient au Chemin optimal entre A et B, alors la portion de ce même chemin allant de A à C est le Chemin optimal entre A et C.» [6] Pour obtenir le chemin optimal du problème, il suffit de construire les différents sous chemins optimaux. Cette méthode a été utilisée dans [7] pour la résolution de problème allant de 10 à 25 clients.

### **2.2.4. Autres methods exactes:**

En dehors des méthodes d'énumération (branch and bound et branch and cut) et la programmation dynamique décrites précédemment, de nombreuses approches exactes différentes existent, et utilisent les spécificités du problème traité pour résoudre le problème d'optimisation.

D'autres approches exactes sont entièrement spécifiques au problème traité, comme l'algorithme de Johnson pour l'ordonnancement [9]. Elles sont généralement appliquées aux problèmes difficiles, même si l'on en rencontre aussi pour la résolution de problèmes NP-difficiles.

### **2.3. Les méthodes approchées :**

Les méthodes de résolution approchée sont utilisées lorsque les méthodes exactes échouent, car la résolution exacte nécessite d'explorer l'ensemble de l'espace de recherche, ce qui devient irréalisable pour les gros problèmes. Dans ces cas, une exécution partielle de l'algorithme exact ne donne généralement pas une solution de bonne qualité. Les méthodes de résolution approchée ont été développées pour fournir rapidement des solutions de bonne qualité mais non optimales.

#### **2.3.1. Les heuristiques :**

Les heuristiques sont des algorithmes qui fournissent rapidement des solutions réalisables pour des problèmes d'optimisation difficiles. Contrairement aux méthodes exactes, elles ne garantissent pas l'optimalité mais offrent des temps de calcul bien plus courts. Leur utilisation s'avère particulièrement efficace pour résoudre des problèmes de grande taille, permettant ainsi de trouver des solutions acceptables dans des délais raisonnables.

Parmi les heuristiques qui traitent le VRP, nous citons :

- le groupe en premier, la route en second ;
- La route en premier, le groupe en second ;
- L'algorithme en Pétale.

##### **2.3.1.1. L'heuristique « groupe en premier, route en second » :**

C'est une des heuristiques les plus connues. Elle se base sur l'aspect géométrique de Problème. Elle consiste à grouper les nœuds qui sont géographiquement voisins, et chaque groupe est assigné à un véhicule, ensuite pour chaque véhicule le TSP correspondant est Résolu [10].

##### **2.3.1.2. L'heuristique « route en premier, groupe en second »:**

Le principe de cette heuristique est de construire des tournées comportant un grand Nombre de clients, qui sont réellement non réalisables, puis de les subdiviser en de petites Tournées pour obtenir des solutions acceptables pour le VRP [11] et [12].

##### **2.3.1.3. L'algorithme en pétale :**

Cet algorithme de balayage génère plusieurs routes, appelées pétales, pour ensuite effectuer une sélection en résolvant un problème de partitionnement. Il est important de noter que si les routes correspondent à des secteurs continus des arcs, alors le problème peut être résolu en temps polynomial.

### **2.3.2. Métaheuristiques :**

Face aux difficultés rencontrées par les heuristiques pour obtenir une solution réalisable de bonne qualité pour des problèmes d'optimisation difficiles, les métaheuristiques ont émergé. Ces algorithmes sont plus complets et complexes qu'une simple heuristique, et ils permettent généralement d'obtenir des solutions de très bonne qualité pour des problèmes provenant des domaines de la recherche opérationnelle ou de l'ingénierie pour lesquels il n'existe pas de méthodes efficaces connues ou lorsque la résolution du problème nécessite beaucoup de temps ou de mémoire de stockage.

Le rapport entre le temps d'exécution et la qualité de la solution trouvée par une métaheuristique reste très intéressant dans la plupart des cas par rapport aux différents types d'approches de résolution. La plupart des métaheuristiques utilisent des processus aléatoires et itératifs pour rassembler des informations, explorer l'espace de recherche et faire face à des problèmes tels que l'explosion combinatoire. Une métaheuristique peut être adaptée à différents types de problèmes, tandis qu'une heuristique est utilisée pour un problème donné. Plusieurs d'entre elles sont souvent inspirées par des systèmes naturels dans de nombreux domaines tels que la biologie (algorithmes évolutionnaires et génétiques), la physique (recuit simulé) et l'éthologie (algorithmes de colonies de fourmis).

Un enjeu majeur de la conception des métaheuristiques est donc de faciliter le choix d'une méthode et le réglage des paramètres pour les adapter à un problème donné. Les métaheuristiques peuvent être classées de nombreuses façons. On peut distinguer celles qui travaillent avec une population de solutions de celles qui ne manipulent qu'une seule solution à la fois. Les méthodes qui tentent itérativement d'améliorer une solution sont appelées méthodes de recherche locale ou méthodes de trajectoire. Ces méthodes construisent une trajectoire dans l'espace des solutions en tentant de se diriger vers des solutions optimales. Les exemples les plus connus de ces méthodes sont la recherche tabou et le recuit simulé. Les algorithmes génétiques, l'optimisation par essaim de particules et les algorithmes de colonies de fourmis sont les exemples les plus connus des méthodes qui travaillent avec une population

#### **2.3.2.1. Le recuit simulé (simulated annealing):**

Le recuit simulé a été introduit par [13] et [14] comme une méthode de recherche locale normale, utilisant une stratégie pour éviter les minima locaux. Cette métaheuristique est basée sur

une technique utilisée depuis longtemps par les métallurgistes qui, pour obtenir un alliage sans défaut, faisant alterner les cycles de réchauffage (ou de recuit) et de refroidissement lent des métaux. Le recuit simulé s'appuie sur des travaux faites par [15], qui ont pu décrire l'évolution d'un système en thermodynamique.

Le principe du recuit simulé est de parcourir de manière itérative l'espace des solutions. On part avec une solution notée  $s_0$  initialement générée de manière aléatoire dont correspondent une énergie initiale  $E_0$ , et une température initiale  $T_0$  généralement élevée.

A chaque itération de l'algorithme, un changement élémentaire est effectué sur la solution, cette modification fait varier l'énergie du système  $\Delta E$ . Si cette variation est négative (la nouvelle solution améliore la fonction objective, et permet de diminuer l'énergie du système), elle est acceptée. Si la solution trouvée est moins bonne que la précédente alors elle sera acceptée avec une probabilité  $P$  calculée suivant la distribution de Boltzmann suivante :

$$P(E, T) = \exp^{-\frac{\Delta E}{T}} \quad (2.1)$$

En fonction du critère de Metropolis, un nombre  $e \in [0, 1]$  est comparé à la probabilité  $P = \exp^{-\frac{\Delta E}{T}}$ . Si  $p \leq e$  la nouvelle solution est acceptée.

Le fonctionnement du critère de Metropolis est interprété par

- Si  $\Delta E = f(s') - f(s) < 0$  alors  $e^{-\frac{\Delta E}{T}} > 1$ , donc  $e$  est toujours inférieur à cette valeur, et on accepte la solution  $s'$
- Si  $\Delta > 0$
- T est très grande, alors  $e^{-\frac{\Delta E}{T}} \cong 1$ , tout voisin est systématiquement accepté.
- T est très petite, alors  $e^{-\frac{\Delta E}{T}} \cong 0$ , une dégradation a peu de chances d'être acceptée

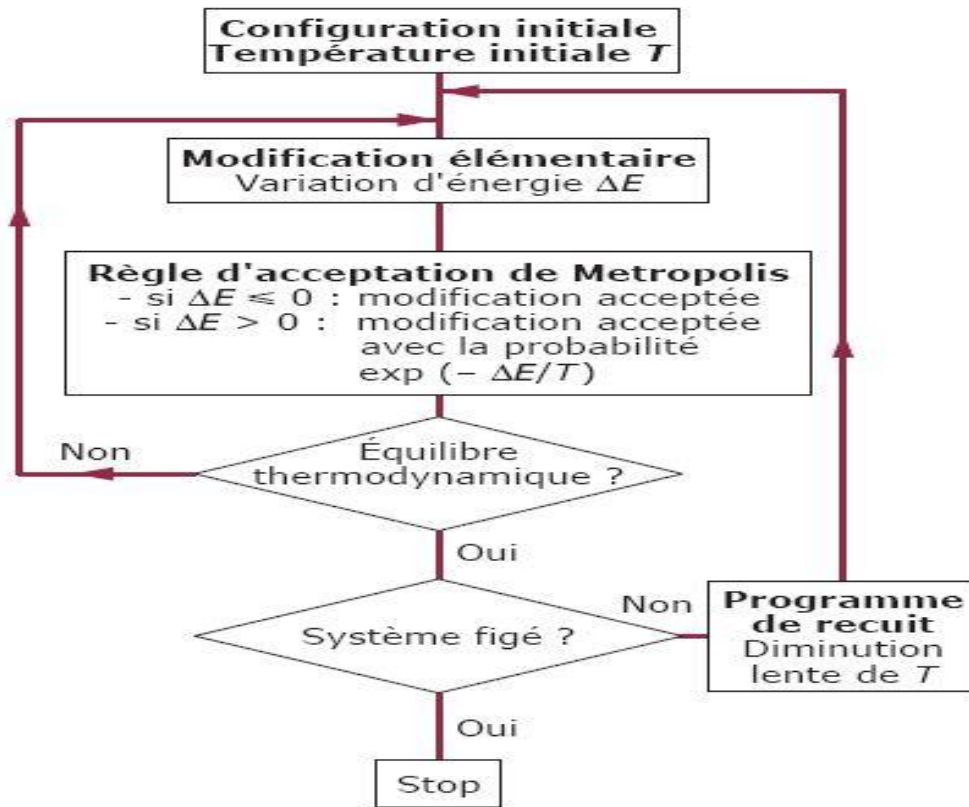


Figure .3: – Fonctionnement de l’algorithme de recuit simulé [16].

### 2.3.2.2. La recherche Tabou (Tabu Search):

La recherche tabou (TS) est une méthode de recherche locale combinée avec un ensemble de techniques permettant d’éviter d’être piégé dans un minimum local ou la répétition d’un cycle. La recherche tabou est introduite principalement par Glover [17], Hansen [18], Glover et Laguna dans [19]. Cette méthode a montré une grande efficacité pour la résolution des problèmes d’optimisation difficiles. En effet, à partir d’une solution initiales dans un ensemble de solutions local  $S$ , des sous-ensembles de solution  $N(s)$  appartenant au voisinage  $S$  sont générés. Par l’intermédiaire de la fonction d’évaluation nous retenons la solution qui améliore la valeur de  $f$ , choisie parmi l’ensemble de solutions voisines  $N(s)$ .

L’algorithme accepte parfois des solutions qui n’améliorent pas toujours la solution courante. Nous mettons en œuvre une liste tabou (tabulist)  $T$  de longueur  $k$  contenant les  $k$  dernières solutions visitées, ce qui ne donne pas la possibilité à une solution déjà trouvée d’être acceptée et stockée dans la liste tabou. Alors le choix de la prochaine solution est effectué sur un ensemble des solutions voisines en dehors des éléments de cette liste tabou. Quand le nombre  $k$  est atteint, chaque nouvelle solution sélectionnée remplace la plus ancienne dans la liste. La construction de la liste tabou est basée sur le principe FIFO, c’est-à-dire le premier entré est le premier sorti. Comme critère d’arrêt on peut par exemple fixer un nombre maximum

d'itérations sans amélioration de  $s^*$ , ou bien fixer un temps limite après lequel la recherche doit s'arrêter.

### 2.3.2.3. Les algorithmes génétiques:

Les algorithmes génétiques (AG) sont des algorithmes d'optimisation stochastique fondés sur les mécanismes de la sélection naturelle et de la génétique. Ils ont été adaptés à l'optimisation par John Holland [20], également les travaux de David Goldberg ont largement contribué à les enrichir [21],[22].

Le vocabulaire utilisé est le même que celui de la théorie de l'évolution et de la génétique, on emploie le terme individu (solution potentielle), population (ensemble de solutions), génotype (une représentation de la solution), gène (une partie du génotype), parent, enfant, reproduction, croisement, mutation, génération, etc.

Leur fonctionnement est extrêmement simple, on part d'une population de solutions potentielles (chromosomes) initiales, arbitrairement choisies. On évalue leur performance (Fitness) relative. Sur la base de ces performances on crée une nouvelle population de solutions potentielles en utilisant des opérateurs évolutionnaires simples : la sélection, le croisement et la mutation. Quelques individus se reproduisent, d'autres disparaissent et seuls les individus les mieux adaptés sont supposés survivre. On recommence ce cycle jusqu'à ce qu'on trouve une solution satisfaisante. En effet, l'héritage génétique à travers les générations permet à la population d'être adaptée et donc répondre au critère d'optimisation, illustre les principales étapes d'un algorithme génétique.

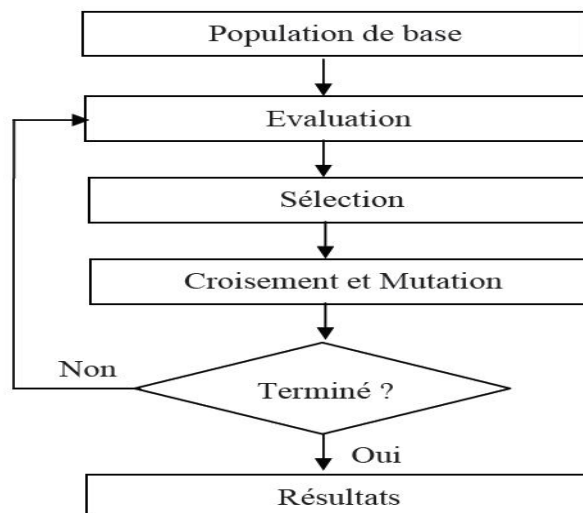


Figure 4: – Fonctionnement d'un algorithme génétique [16].

### 2.3.2.4. Les algorithmes de colonies de fourmis:

Les algorithmes de colonies de fourmis ont été proposés par Colorni, Dorigo et Maniezzo en 1992 [23] et appliquées la première fois au problème du voyageur de commerce. Ce sont des

algorithmes itératifs à population où tous les individus partagent un savoir commun qui leur permet d'orienter leurs futurs choix et d'indiquer aux autres individus des choix à suivre ou à éviter. Le principe de cette métaheuristique repose sur le comportement particulier des fourmis, elles utilisent pour communiquer une substance chimique volatile particulière appelée phéromone grâce à une glande située dans leur abdomen. En quittant leur nid pour explorer leur environnement à la recherche de la nourriture, les fourmis arrivent à élaborer des chemins qui s'avèrent fréquemment être les plus courts pour aller du nid vers une source de nourriture. Chaque fourmi dépose alors une quantité de phéromones sur ces pistes qui deviendront un moyen de communication avec leurs congénères, les fourmis choisissent ainsi avec une probabilité élevée les chemins contenant les plus fortes concentrations de phéromones à l'aide des récepteurs situés dans leurs antennes.

La figure 5 illustre et confirme ce constat, une expérience a été faite par gauss et de neubourg en 1989 appelée expérience du pont à double branche, Les fourmis qui retournent au nid rapidement, après avoir visité la source de nourriture, sont celles qui ont choisi la branche courte et les fourmis empruntant cette branche faisant plus d'aller-retour, et par conséquent la quantité de phéromones déposée sur la plus courte branche est relativement supérieure que celle présente sur la plus longue branche. Puisque les fourmis sont attirées plus vers les pistes de plus grande concentration en phéromones, alors la branche courte sera la plus empruntée par la majorité des fourmis. Cette métaheuristique a permis de résoudre différents problèmes d'optimisation combinatoire à forte complexité, comme le problème du voyageur de commerce [24], le problème de coloration de graphe [25], le problème d'affectation quadratique [26] ou le problème de routage de véhicules [27], etc.

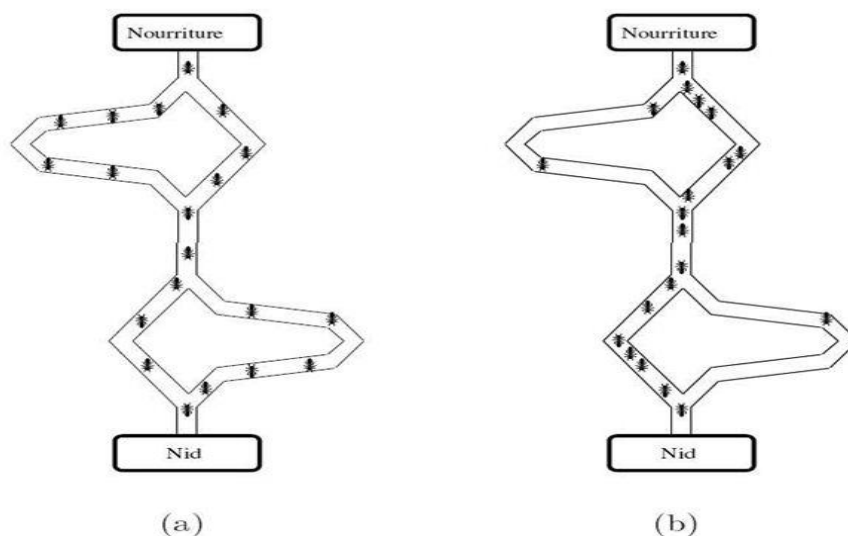


Figure .5: – L'expérience du pont à double branche [16].

### 2.3.2.4.1. Principe de l'algorithme:

Dans ce paragraphe on décrit l'implémentation d'un algorithme de colonie de fourmis original dit 'Ant System' (AS) (Colomi et al. 1992a) [28], orienté pour résoudre le problème de voyageur de commerce (TSP), ce problème consiste à trouver le plus court cycle hamiltonien dans un graphe, où chaque sommet du graphe représente une ville. La distance entre les villes  $i$  et  $j$  est représentée par  $d_{ij}$ , et le couple  $(i, j)$  représente l'arête entre ces deux villes. Nous initialisons d'abord la quantité de phéromone sur les arêtes à  $\tau_{init} > 0$ , chaque fourmi parcourt le graphe et construit un trajet complet (une solution). A chaque étape de la construction des solutions, la fourmi doit décider à quel sommet elle va se déplacer, cette décision est prise d'une manière probabiliste fondée sur les valeurs de phéromone et d'une information statistique qui permet notamment de trouver une bonne solution.

La probabilité pour qu'une fourmi  $k$  se déplace du sommet  $i$  au sommet  $j$ , qui appartient à un ensemble de sommets qui ne sont pas encore visités par la fourmi  $k$  noté  $S_i^k$ , est :

$$P_{ij}^k(t) = \frac{(\tau_{ij}(t))^\alpha (n_{ij})^\beta}{\sum_{l \in S_i^k} (\tau_{il}(t))^\alpha (n_{il})^\beta} \quad (1.3)$$

$\alpha$  et  $\beta$  sont deux paramètres qui influencent sur l'importance de l'intensité de phéromone,  $\tau_{ij}$  et l'information statistique dite visibilité  $n_{ij}$ . Cette valeur guide le choix des fourmis vers des villes proches, et éviter le plus possible celles trop lointaines ( $n_{ij} = \frac{1}{d_{ij}}$ ). Pour  $\alpha = 0$ , on prend en compte juste la visibilité c'est-à-dire que le choix sera tombé à chaque fois sur la ville la plus proche. Si  $\beta = 0$ , seules les pistes de phéromones jouent sur le choix. Pour éviter une sélection trop rapide d'un chemin, un compromis convenable entre ces deux paramètres est obligatoire.

### 2.3.2.4.2. Mise à jour de phéromones:

Lorsque toutes les fourmis ont construit une solution, une quantité de phéromones  $\Delta\tau_k$  est déposée par chaque fourmi  $k$  sur son trajet.

Pour toute itération  $t$ , si le chemin  $(i, j)$  est dans la tournée de la fourmi  $k$  la quantité de phéromones déposée sur ce trajet est :

$$\Delta\tau_{ij}^k(t) = \frac{Q}{L^k(t)}$$

Où  $L^k(t)$  est la longueur totale de la tournée de la fourmi  $k$ , et  $Q$  est une constant.

Donc l'ajout de la quantité de phéromones dépend certainement de la qualité de la solution obtenue c'est-à-dire plus la tournée parcourue est petite plus l'ajout de la quantité de phéromones est important.

Dans le but de ne pas négliger toutes les mauvaises solutions obtenues, et ainsi éviter la convergence vers des optima locaux de mauvaise qualité, le concept d'évaporation des pistes de phéromones est simulé à travers un paramètre  $\rho$  appelé le taux d'évaporation ( $0 < \rho < 1$ ) comme suit :

$$\tau_{ij}(t + 1) \leftarrow (1 - \rho) \cdot \tau_{ij}(t) + \Delta \tau_{ij}(t) \quad (1.4)$$

Où  $\Delta \tau_{ij}(t) = \sum_{k=0}^m \Delta \tau_{ij}^k(t)$   $t$  représente une itération donnée et  $m$  le nombre de fourmis.

### 2.3.2.5. L'optimisation par essaim de particules:

#### 2.3.2.5.1. Principe général:

L'optimisation par Essaim Particulaire a été proposée par Kennedy et Eberhart [29]. Cette méthode est inspirée du comportement social des animaux évoluant en essaim.

Au départ, ils cherchaient à simuler la capacité des oiseaux à voler de façon synchrone et leur aptitude à changer brusquement de direction tout en restant en une formation optimale.

Les particules sont les individus et elles se déplacent dans l'hyperespace de recherche en se basant sur des informations limitées :

- ✓ Chaque particule est dotée d'une mémoire qui lui permet de mémoriser le meilleur point par lequel elle est déjà passée et elle a tendance à retourner vers ce point.
- ✓ Chaque particule est informée du meilleur point connu au sein de son voisinage et elle va tendre à aller vers ce point.

Chaque individu utilise donc, non seulement, sa propre mémoire, mais aussi l'information locale sur ses plus proches voisins pour décider de son propre déplacement. Des règles simples, telles que " aller à la même vitesse que les autres ", " se déplacer dans la même direction " ou encore " rester proche de ses voisins " sont des exemples de comportements qui suffisent à maintenir la cohésion de l'essaim.

Le déplacement d'une particule est influencé par les trois types de comportement :

- Une composante physique : la particule tend à suivre sa propre voie ;
- Une composante cognitive : la particule tend à revenir vers le meilleur site par lequel elle est déjà passée ;
- Une composante sociale : la particule tend à se diriger vers le meilleur site déjà atteint par ses voisins.

**2.3.2.5.2. Algorithme de base:**

Un essaim de particules survole l'espace de recherche, en quête de l'optimum global.

Dans  $R_n$ , la particule  $i$  (solution potentielle) de l'essaim est modélisée par son vecteur position  $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$  et par son vecteur vitesse  $v_i = (v_{i1}, v_{i2}, \dots, v_{in})$ .

Cette particule garde en mémoire la meilleure position par laquelle elle est déjà passée, que l'on note  $p_i = (p_{i1}, p_{i2}, \dots, p_{in})$ . La meilleure position atteinte par toutes les particules de l'essaim est notée  $p_g = (p_{g1}, p_{g2}, \dots, p_{gn})$ .

Au temps  $t + 1$ , le vecteur vitesse est calculé à partir de 1.5

$$v_{ij}(t + 1) = v_{ij}(t) + c_1 r_1 (p_{ij}(t) - x_{ij}(t)) + c_2 r_2 (p_{gj}(t) - x_{ij}(t)) \quad (1.5)$$

$c_1$  et  $c_2$  sont deux constantes, appelées coefficients d'accélération ;  $r_1$  et  $r_2$  sont deux nombres aléatoires tirés uniformément dans  $[0, 1]$ .

- $v_{ij}(t)$  correspond à la composante physique du déplacement.
- $c_1 r_1 (p_{ij}(t) - x_{ij}(t))$  correspond à la composante cognitive du déplacement.  $c_1$  pondère les tendances de la particule à vouloir suivre son instinct de conservation et à aller vers sa meilleure position connue.
- $c_2 r_2 (p_{gj}(t) - x_{ij}(t))$  correspond à la composante sociale du déplacement.  $c_2$  contrôle l'aptitude sociale de la particule en se rapprochant plus de la meilleure position de ses informatrices.

La position de la particule  $i$  est alors définie par :

$$x_i(t + 1) = x_i(t) + v_i(t + 1)$$

Il est à noter que le terme "vitesse" est ici abusif car les vecteurs  $v_i$  ne sont pas homogènes à une vitesse. Il serait plus approprié de parler de "direction de déplacement".

La stratégie de déplacement d'une particule est illustrée dans la figure 2.7

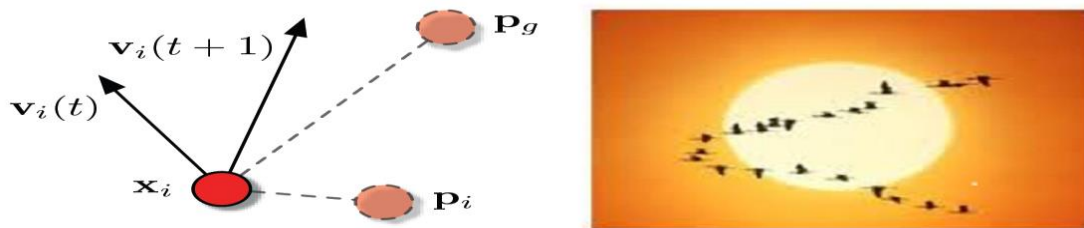


Figure .6: – Analysis of Particle Trajectories [16].

Le pseudo code pour la version la plus générale de l'algorithme .

Le critère d'arrêt peut être différent suivant le problème posé et les exigences de l'utilisateur. Si l'optimum global est connu a priori, on peut définir une "erreur acceptable" comme critère

d'arrêt. Sinon, on peut fixer nombre maximum d'itérations ou un nombre maximum d'évaluations de la fonction objectif.

#### **2.3.2.6. La recherche dispersée:**

La recherche dispersée est une méthode d'évolution qui a été proposée par (Glover et Laguna 1997) [19]. Elle a été proposée dans le cadre de la résolution des problèmes combinatoire.

Cette métaheuristique se distingue des algorithmes évolutionnistes classiques par l'utilisation d'une procédure de recherche locale et par la généralisation de l'opérateur de croisement. Tout comme les algorithmes génétiques, elle est basée sur une population de solutions qui évolue dans le temps à l'aide à la fois d'un opérateur de sélection, de la combinaison linéaire de solutions de la population pour créer une nouvelle solution provisoire (non forcément entière ou admissible), d'un opérateur de projection permettant de rendre la solution provisoire admissible et d'opérateurs d'élimination.

#### **2.3.2.7. Description générale de la recherche dispersée:**

Le concept général de la recherche dispersée qui a été donné par (Glover et al. 1998) [30] se base sur les trois fondations suivantes :

- L'information utile sur la solution optimale est typiquement contenue dans une collection diverse de solutions élites.
- Les stratégies de combinaison des solutions contiennent des mécanismes qui incorporent la diversification et l'intensification dans la génération des nouvelles solutions.
- L'utilisation de plusieurs combinaisons de solutions de référence augmente l'opportunité d'exploiter l'information contenue dans l'union des solutions élites.

Le modèle de l'algorithme de la recherche dispersée repose sur la définition des cinq composants majeurs (dits méthodes) suivants :

- ✓ Méthode de génération de diversification : Consiste à générer une collection de diverses solutions de départ.
- ✓ Méthode d'amélioration : Les points sont optimisés à l'aide d'une recherche locale.
- ✓ Méthode de mise à jour de l'ensemble de référence  $R_i$  : construit et maintient un ensemble de référence obtenu en sélectionnant les  $b$  meilleures solutions trouvées.
- ✓ Méthode de génération des sous-ensembles : génère des sous-ensembles  $D_i$  à partir de l'ensemble de référence.
- ✓ Méthode de combinaison de solutions : génère des solutions en combinant les solutions de chaque sous-ensemble  $D_i$  pour produire l'ensemble  $C_i$ .

## **2.4.Conclusion :**

Nous pouvons conclure que les métaheuristiques présentent une classe de méthodes approchées adaptables à une grande variété de problèmes d'optimisation combinatoire et mènent à des résultats pertinents. Mais il existe assez peu de contributions permettant de comprendre la raison de cette efficacité, et aucune preuve ne peut montrer qu'une métaheuristique sera plus efficace qu'une autre sur un problème donné. Certaines métaheuristiques présentent l'avantage d'être simples à mettre en œuvre, comme le cas du recuit simulé, d'autres sont bien adaptées à la résolution de certaines classes de problème, très contraints, comme le système de colonies de fourmis. La qualité des solutions trouvées par les métaheuristiques dépend de leur paramétrage, et de l'équilibre entre un balayage de tout l'espace des solutions (diversification) et une exploration locale (l'intensification).

## **CHAPITRE 03**

# **UN ALGORITHME DE GÉNÉTIQUE APPLIQUÉES AU PROBLÈME DU VRP**

### 3.1. Introduction:

Le Problème de Tournées de Véhicules (VRP) est un problème majeur dans l'optimisation combinatoire. Il a été l'objet de nombreuses études en raison de sa difficulté de résolution et de ses nombreuses applications pratiques dans la logistique. Avec la montée en puissance des métaheuristiques telles que les algorithmes génétiques, de nombreuses instances de problèmes ont été résolues de manière efficace, voire optimale. Cependant, les méthodes de résolution utilisées sont souvent entièrement automatiques et ne font intervenir les humains que dans la phase de modélisation du problème. Elles ne permettent donc pas d'exploiter toute la richesse des connaissances et de l'expertise que les opérateurs humains peuvent apporter. De plus, leur fonctionnement en "boîte noire" ne répond pas au besoin de ces opérateurs de comprendre le déroulement du processus d'optimisation.

### 3.2. Résolution interactive de problèmes d'optimisation (VRP):

Des travaux ont été menés récemment pour sortir de ce paradigme de type « client » ou « oracle » (l'humain soumet une question au système, qui lui fournit une réponse) et aller vers une résolution interactive. Dans l'approche HuGS l'humain attribue à chaque client du VRP une valeur de *mobilité* (faible, moyenne, forte) pour contraindre un algorithme de recherche locale dans son exploration du voisinage de la solution courante. Gacias et al. permettent à l'humain de construire une solution partielle d'un problème de tournées avec fenêtres temporelles (VRPTW) et de demander au système de la compléter.

Dans la famille des méthodes d'optimisation à base de population, les algorithmes génétiques ont eux aussi été adaptés à une résolution interactive : la machine génère des solutions, et l'humain les évalue. Cette approche est adaptée aux problèmes dont la fonction objectif n'est pas formulable mathématiquement tels que les problèmes artistiques, et présente un inconvénient majeur lié à la fatigue de l'utilisateur.

### 3.3. Un algorithme génétique pour le VRP:

Les algorithmes génétiques ont été appliqués avec succès à la résolution du problème du voyageur de commerce, et ont donc naturellement été adaptés aussi à la résolution du problème de routage de véhicules dans ses variantes courantes.

Ces algorithmes sont tout particulièrement performants sur le VRPTW, et ont produit certaines des meilleures solutions connues sur les problèmes tests [33].

Les adaptations des algorithmes génétiques au VRP sont tellement nombreuses que l'état de l'art dressé par Haj-Rachid et al.[31] dénombre 5 types de codage, 2 méthodes de création de la population initiale, 3 méthodes d'évaluation des individus, 5 types de sélection des parents, 22 types de croisements, 6 types de mutations, 4 stratégies de construction de la génération suivante, et 3 types de critères d'arrêt.

Notre but n'étant pas de détailler exhaustivement toutes les adaptations possibles, nous nous contentons de deux parmi les meilleures d'entre elles [32] : il s'agit des adaptations de Baker and Ayechev [33] et de Prins[34] appliquées au VRP avec contraintes de capacités (CVRP). Pour les illustrer, nous utiliserons l'exemple de la figure 7 (VRP ayant  $n = 20$  clients).

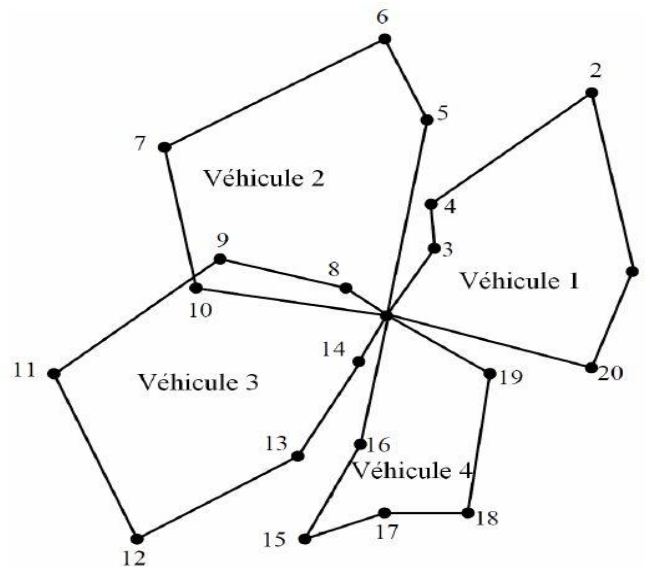


Figure .7: -Vrp ayant n=20 clients [35] .

### 3.3.1. Codage:

Un exemple de codage direct d'une solution de VRP consiste à représenter celle-ci sous forme de plusieurs séquences de clients dont chacune correspond à une tournée.

Des délimiteurs, appelés séparateurs de tournées, peuvent être insérés entre les séquences en question pour séparer les tournées.

À titre d'exemple, la solution de la figure 7 peut être représentée par le chromosome suivant:

0	20	1	2	4	3	0	5	6	7	10	0	8	9	11	12	13	14	0	16	15	17	18	19	0
---	----	---	---	---	---	---	---	---	---	----	---	---	---	----	----	----	----	---	----	----	----	----	----	---

Tableau 1:- tableau de chromosome initiale.

Le séparateur de tournée utilisé est le symbole **0**(le dépôt) et permet de distinguer le début et la fin de chaque tournée.

L'ordre des clients est important car il détermine le sens dans lequel sont effectuées les tournées.

En supprimant le dépôt de ce chromosome, on obtient un deuxième chromosome correspondant à un codage indirect de la même solution :

20	1	2	4	3	5	6	7	10	8	9	11	12	13	14	16	15	17	18	19
----	---	---	---	---	---	---	---	----	---	---	----	----	----	----	----	----	----	----	----

Tableau 2:- tableau de deuxième chromosome.

En effet, cette représentation peut être vue comme une seule méta-tournée d'un voyageur du commerce. Elle indique certes l'ordre des clients dans les tournées ,mais ne permet pas de connaître les débuts et les fins de celles-ci. Une procédure supplémentaire de découpage est donc utilisée. Elle se base généralement sur les contraintes du problème telles que la capacité des véhicules ou les fenêtres temporelles des clients.

Prins [34] utilise ce deuxième type codage car les codages avec séparateurs de tournées ne se sont pas avérés très efficaces à cause des procédures de réparation auxquelles elles doivent recourir.

La procédure de découpage (appelée Split) appliquée à un chromosome consiste à résoudre un problème de plus court chemin Haj-Rachid et al.[31].

Ces deux premiers codages sous forme de cycles correspondent à une approche de type route first-cluster second. Une approche de type cluster first-route second donnerait lieu à un codage de type affectation [33] : le chromosome indique pour chaque client le n° du véhicule qui le dessert.

Il s'agit ainsi d'un codage indirect puisqu'il ne donne pas d'information sur l'ordre des tournées au sein des partitions de clients.

Baker and Ayechev [33] utilisent ce codage, et représentent la solution de la figure 7 avec le chromosome suivant :

1	1	1	1	2	2	2	3	3	2	3	3	3	3	4	4	4	4	4	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Tableau 3:- tableau de troisième chromosome.

### 3.3.2. Opérateurs génétiques:

#### 3.3.2.1. Procédure de sélection:

##### 3.3.2.1.1. La sélection par roulette:

La sélection par roulette consiste à associer à chaque individu un segment dont la longueur est proportionnelle à sa fitness. Ces segments sont ensuite concaténés sur un axe que l'on normalise entre 0 et 1. On tire alors un nombre aléatoire de distribution uniforme entre 0 et 1, puis on regarde quel est le segment sélectionné.

Si la population d'individus est de taille égale à N, alors la probabilité de sélection d'un individu  $x_i$  notée  $p(x_i)$  est égale à :

$$p(x_i) = \frac{f(x_i)}{\sum_{k=1}^N f(x_k)}$$

En pratique, on calcule pour chaque individu  $p(x_i)$  sa probabilité cumulée  $q_i = \sum_{j=1}^i p(x_j)$  et on choisit aléatoirement un nombre r compris entre 0 et 1.

L'individu retenu est  $x_1$  si  $q_1 \geq r$  ou  $(2 \leq i \leq N)$  si  $q_{i-1} < r \leq q_i$

Ce processus est répété N fois. Avec une telle sélection, un individu fort peut être choisi plusieurs fois. Par contre, un individu faible a moins de chance d'être sélectionné.

##### 3.2.1.2. La sélection par rang:

La sélection par rang trie d'abord la population par fitness. Chaque chromosome se voit associé un rang en fonction de sa position. Le plus mauvais chromosome aura le rang 1, le suivant 2, et ainsi de suite jusqu'au meilleur chromosome qui aura le rang N (pour une population de N chromosomes). La sélection par rang d'un chromosome est la même que par roulette, mais les proportions sont en relation avec le rang plutôt qu'avec la valeur de l'évaluation. Avec cette méthode de sélection, tous les chromosomes ont une chance d'être sélectionnés. Cependant, elle conduit à une convergence plus lente vers la bonne solution.

Ceci est dû au fait que les meilleurs chromosomes ne diffèrent pas énormément des plus mauvais.

### 3.3.2.1.3 La sélection par tournoi :

Elle consiste à choisir aléatoirement deux ou plusieurs individus et à sélectionner le plus fort. Ce processus est répété plusieurs fois jusqu'à l'obtention de N individus. L'avantage d'une telle sélection est d'éviter qu'un individu très fort soit sélectionné plusieurs fois.

### 3.3.2.2. Opérateur de croisement :

Pour illustrer les opérateurs de croisement, nous utiliserons la solution de la figure 7 comme premier parent (**P1**), et une autre solution comme deuxième parent **P2** (problème à  $n = 20$  clients et résolu avec  $m = 4$  véhicules).

Les opérateurs s'effectuent à deux points, entre les positions 6-7 et 15-16 du chromosome.

Baker and Ayechev [33] utilisent un opérateur classique de croisement à deux points, identique à celui de la figure 8. Le croisement entre les deux solutions de la figure 9 s'effectue de cette manière :

Les chromosomes des deux solutions enfants héritent chacun d'une partie de ceux des parents. Cet héritage s'effectue comme suit : pour l'enfant **E1** (respectivement **E2**), les éléments situés entre les points de croisement sont hérités du parent **P1** (respectivement **P2**) et les éléments restants **P2** (respectivement **P1**).

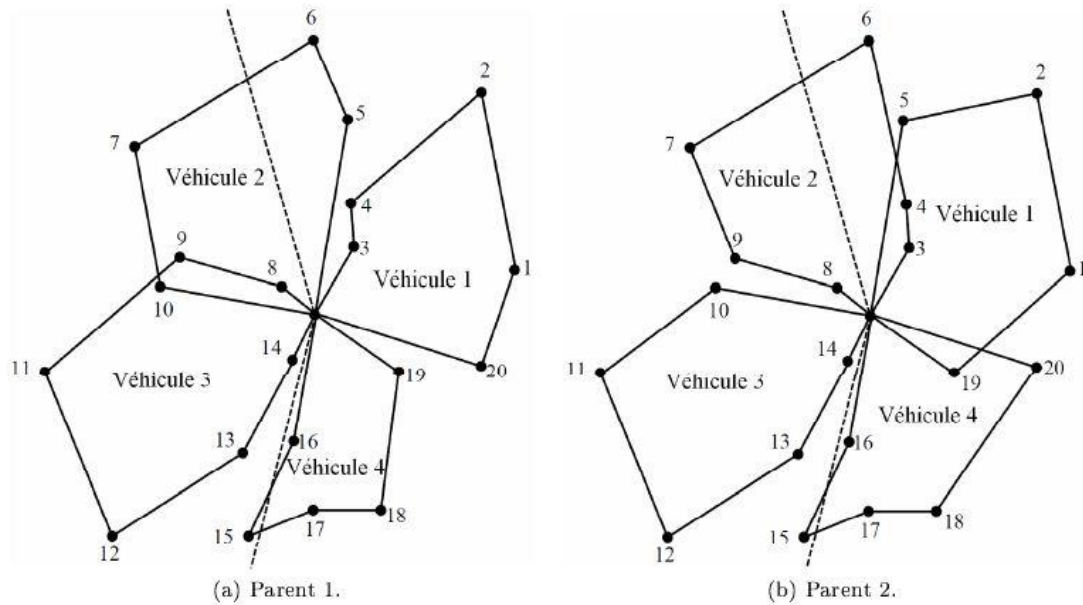


Figure .8: - Deux solutions utilisées comme parents pour illustrer l'opérateur de croisement de Baker and Ayechev [33].

P2 (respectivement P1) :

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1 :	1	1	1	1	2	2	2	3	3	2	3	3	3	3	4	4	4	4	4	1
P2 :	1	1	2	2	1	2	2	2	2	3	3	3	3	3	4	4	4	4	1	4

↓

E1 :	2	3	3	2	3	3	3	3	4											
E2 :	2	2	2	3	3	3	3	3	4											

↓

E1 :	1	1	2	2	1	2	2	3	3	2	3	3	3	3	4	4	4	4	1	4
E2 :	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	1

Figure 9: - croisement de Baker and Ayechev [33].

D'après le chromosome de la solution **E2**, on sait que le véhicule 1 (respectivement 2,3, 4) assure la tournée entre les clients {1, 2, 3, 4, 20} (respectivement {5, 6, 7, 8, 9}, {10, 11, 12, 13, 14}, {15, 16, 17, 18, 19}). L'information manquante est l'ordre dans lequel s'effectuent ces tournées : un TSP est alors résolu pour chaque tournée. Les tournées résultantes sont dessinées dans la figure 10.

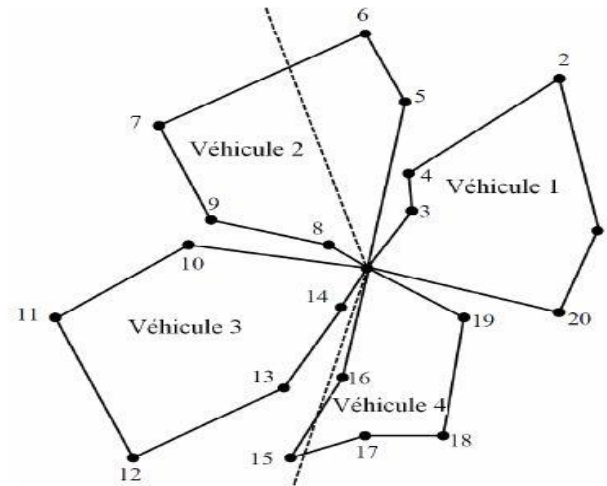


Figure 10- Enfant E2 généré par le croisement des parents[33].

Nous utiliserons une autre solution parent **P2** pour illustrer l'opérateur de croisement de Prins on chromosome est le suivant :

17	18	19	20	1	3	4	2	6	5	8	7	9	10	11	14	13	12	15	16
----	----	----	----	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----

Tableau 4:- tableau de quatrième chromosome .

Le codage de Prins étant proche de celui du codage d'un TSP, alors l'opérateur de croisement utilisé est l'opérateur OrdreCrossover (OX), un opérateur classique pour le TSP, dont le principe est le suivant :

1. Pour l'enfant **E1** (respectivement **E2**), les éléments situés entre les points de croisement sont hérités du parent **P1** (respectivement **P2**),
2. Les éléments restants de l'enfant **E1** (respectivement **E2**) sont hérités aussi du parent **P1** (respectivement **P2**) mais ne sont pas recopiés tels quels (sinon **E1** (resp. **E2**) serait identique à **P1** (resp. **P2**)). Ils sont recopiés selon leur ordre d'apparition dans l'autre parent **P2** (resp. **P1**), à partir de la position située juste après le deuxième point de croisement.

Dans l'exemple, les éléments entre les points de croisement dans **E2** sont mis en évidence en gris clair. Les éléments restants sont {20, 1, 2, 4, 3, 5, 16, 15, 17, 18, 19} et sont mis en évidence en gris foncé. Ces éléments sont soulignés dans le deuxième parent **P2**. À

partir de la 16<sup>ème</sup> position (position situées juste après le deuxième point de croisement), ces éléments apparaissent dans **P2** selon l'ordre

{15, 16, 17, 18, 19, 20, 1, 3, 4, 2, 5} et sont rajoutés à **E1** dans ce même ordre d'apparition :

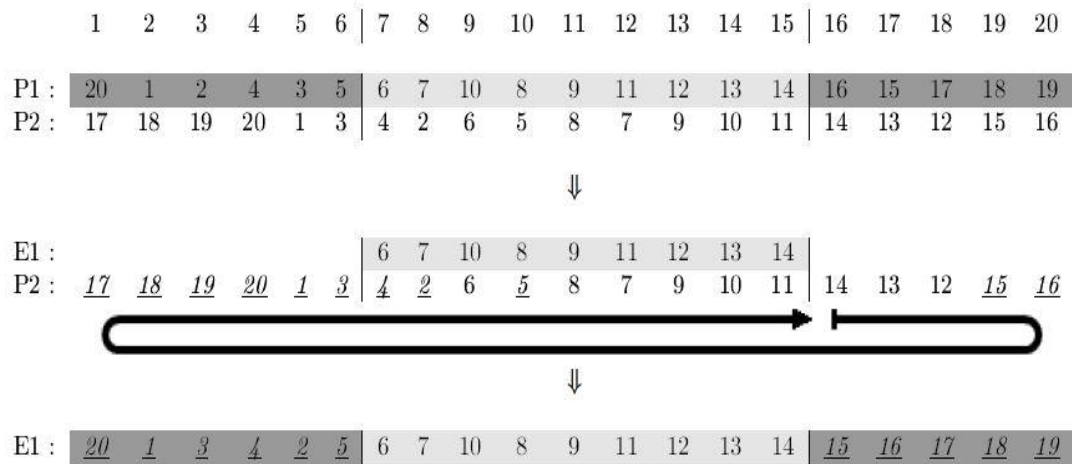


Figure 11 – croisement des chromosomes 1 [33].

De manière similaire, les éléments rajoutés à l'enfant **E2** sont

{15, 16, 17, 18, 19, 20, 1, 3, 4, 2, 5}, le chromosome obtenu est alors :

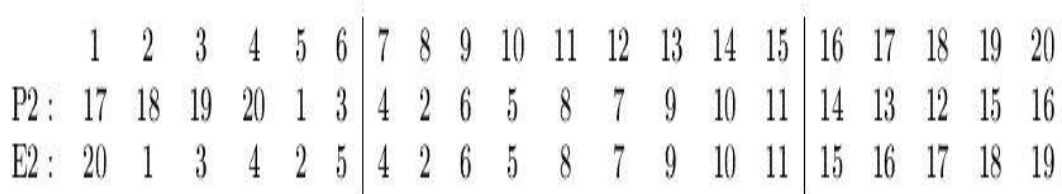


Figure 12:– croisement des chromosomes 2 [33] .

Notons qu'un autre opérateur classique, dit Linear Order Crossover (LOX), opère de manière similaire à OX : après la copie des éléments entre les deux points de croisement, la copie des éléments restants commence à la 16<sup>ème</sup> position (juste après le deuxième point de croisement) s'effectue selon leur ordre d'apparition dans l'autre parent. Cependant, dans LOX, cet ordre est calculé à partir de la première position, et non pas la 16<sup>ème</sup>.

L'application de LOX aux deux parents traités précédemment donne que l'enfant E1, les éléments restants sont copiés dans l'ordre {17, 18, 19, 20, 1, 3, 4, 2, 5, 15, 16}.

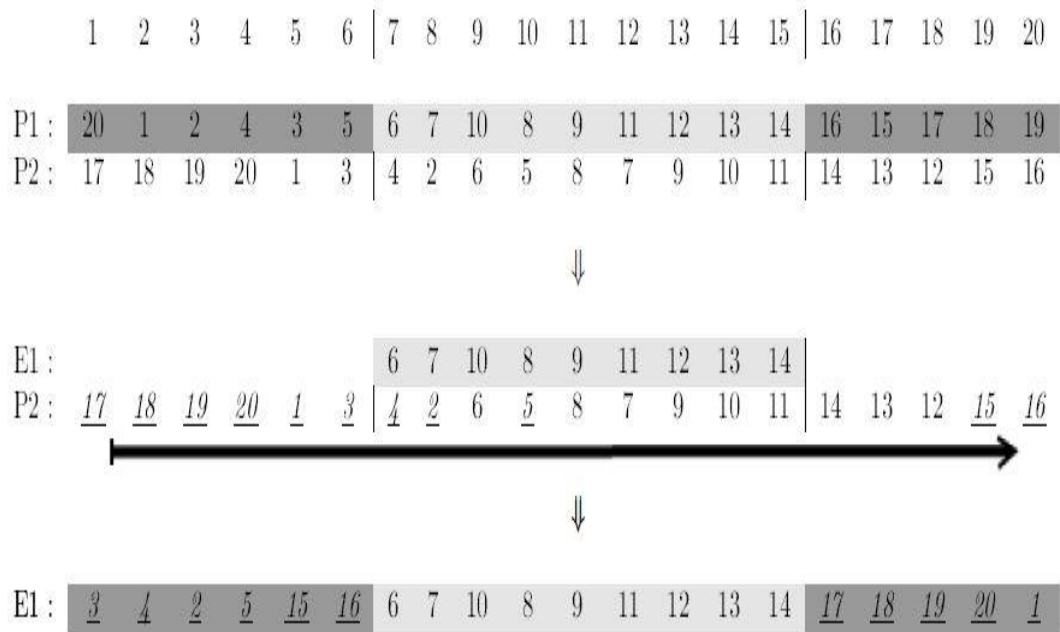


Figure 12: – croisement des chromosomes 3 [33].

En dépit de leur principe simple, les deux opérateurs de croisement OX et LOX sont les plus utilisés dans la littérature [31]. Prins [34] a choisi OX car il s'est avéré plus efficace que LOX lors des expérimentations.

### 3.3.2.3. Opérateur de mutation :

Prins n'applique pas un vrai opérateur de croisement mais le remplace par une procédure d'amélioration basée sur une recherche locale de type 2-opt

Voilà pourquoi son algorithme génétique est qualifié d'hybride. Baker and Ayechev utilise un opérateur de mutation classique basé sur l'échange aléatoire de deux clients entre deux tournées.

### 3.3.2.4. Population initiale:

Les deux approches utilisent des heuristiques simples pour générer la population initiale ; telles que la méthode des économies et la méthode de balayage. Haj-Rachid et al. [31] identifient deux autres moyens d'initialiser les algorithmes génétiques : générer la population de façon totalement aléatoire, ou combiner la génération aléatoire et le recours aux heuristiques.

### 3.3.2.5. Fitness:

Le plus souvent, la fonction d'évaluation (fitness) est calculée à partir d'une somme pondérée de plusieurs objectifs (méthode d'agrégation).

Berger et al. [35] ont par exemple proposé la formule

$$\text{Fitness} = \mathbf{R} - \mathbf{R}_{\min} + \frac{\min(D, 2D_{\min})}{D_{\min}}$$

Où :

- $\mathbf{R}$  est le nombre de tournées dans la solution à évaluer,
- $\mathbf{R}_{\min}$  est le nombre de tournées dans la meilleure solution dans la population courante,
- $D$  est la distance totale parcourue dans la solution à évaluer
- $D_{\min}$  est la distance totale parcourue dans la meilleure solution de la population courante.

Cette formulation de la fonction objectif implique que les meilleures solutions recherchées ont peu de tournées et des distances totales parcourues petites.

La formulation de la fitness proposée par Sushil et Rilun [36] permet de prendre en compte les fenêtres temporelles

$$\text{Fitness} = w_1 * D + w_2 * T + w_3 * R_1 + w_4 * R_2 + w_5 * O$$

Où :

- $D$  est la distance totale parcourue par les véhicules,
- $T$  est la durée totale des différentes tournées,
- $R_1$  est le retard total des véhicules chez les clients,
- $R_2$  est le retard total des véhicules au dépôt,
- $O$  est la surcharge totale de capacité des véhicules et les  $w_i$  sont les poids 10 (coefficients) de pondération de ces objectifs.

Housroum [37], qui a repris cette formulation générale, a comparé les formulations de 17 publications différentes traitant du VRPTW. Il en conclut que les objectifs les plus utilisés sont le nombre de véhicules utilisés et la distance totale parcourue.

### **3.4. Conclusion :**

Nous remarquerons que en termes de qualité des solutions obtenues, les métaheuristiques sont généralement meilleures que les heuristiques et que en termes de temps d'exécution la comparaison est inversée. Ce résultat est prévisible car plusieurs métaheuristiques appliquent elles-mêmes des heuristiques d'amélioration sur leurs solutions (hybridation des méthodes). De plus, les algorithmes génétiques sont considérés comme étant parmi les métaheuristiques les plus efficaces sur les principaux modèles de problèmes de distribution de marchandises. Des études montrent que les algorithmes génétiques surpassent les heuristiques dans la résolution du problème de distribution de marchandises avec des fenêtres temporelles. Par exemple, l'efficacité des algorithmes génétiques a été confirmée dans les modèles étudiés par Baker et Ayechev et Prins .

## **CHAPITRE 04**

# **VISUALISATION DES RESULTATS DE ALGORITHME GENETIQUE**

### 4.1.Introduction

Dans ce dernier chapitre, nous vous présentons une application Web de géolocalisation qui vise à simplifier la résolution du Problème de Tournées de Véhicules (VRP) grâce à une interface utilisateur en ligne conviviale. Cette plateforme offre une manière accessible d'entrer les données du problème et d'afficher les solutions de manière claire. Les utilisateurs peuvent saisir des informations telles que les emplacements, les distances entre eux et le nombre de véhicules disponibles. De plus, les solutions sont présentées de manière interactive et visuelle, ce qui permet d'afficher les tournées proposées sur la carte et de fournir des résumés des distances parcourues. Des outils d'analyse sont également disponibles pour une meilleure compréhension des solutions proposées, ainsi que la possibilité de personnaliser l'affichage en ajustant les paramètres ou en examinant les solutions sous différents angles. Enfin, l'application s'engage à garantir la sécurité et la confidentialité des données, assurant ainsi une utilisation fiable et efficace.

### 4.2. UML Diagrammes de notre application web :

Dans la section suivante, nous présenterons les diagrammes UML essentiels de notre application web. Cette application est créée pour administrer notre solution, où le gestionnaire peut prendre note des demandes des clients et lancer notre solution pour les satisfaire.

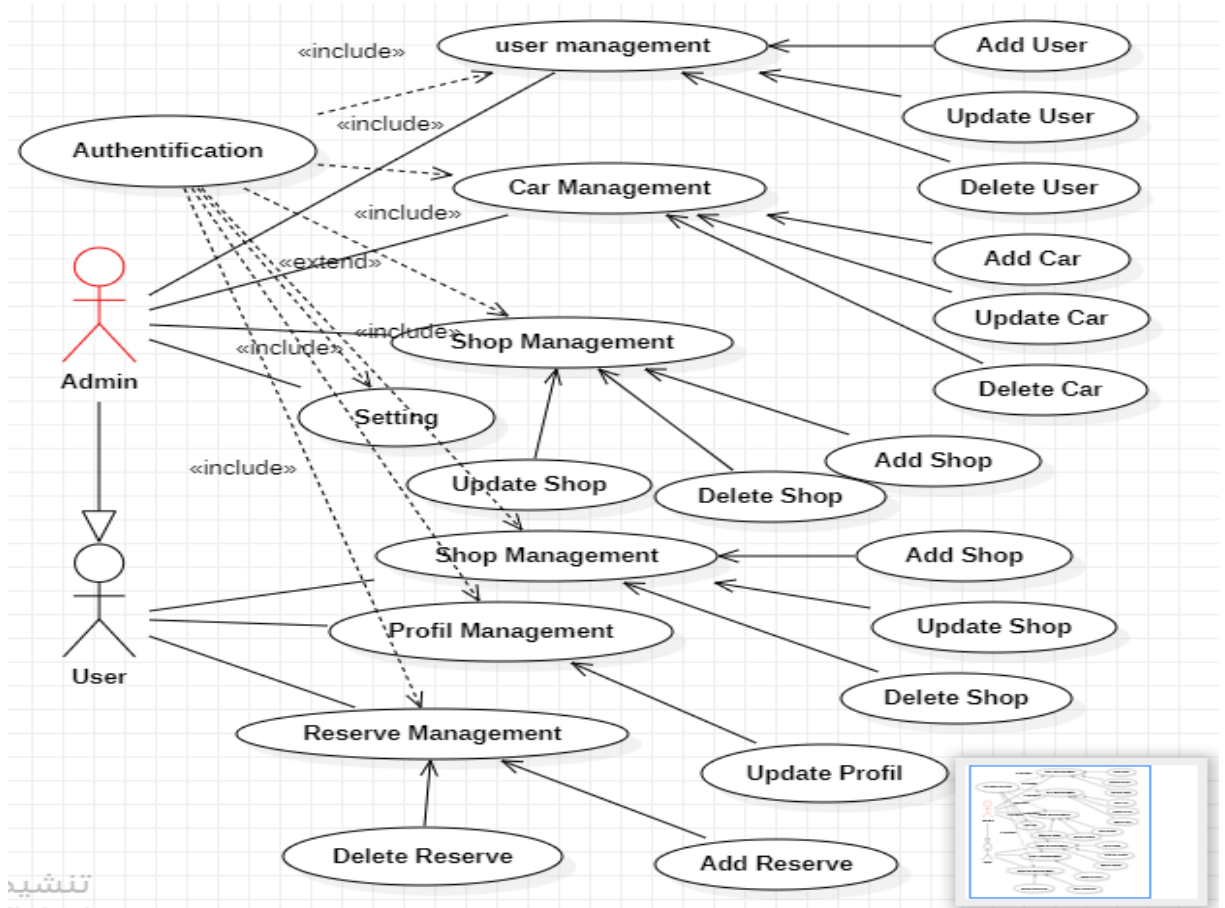


Figure 13: Diagramme de cas d'utilisation.

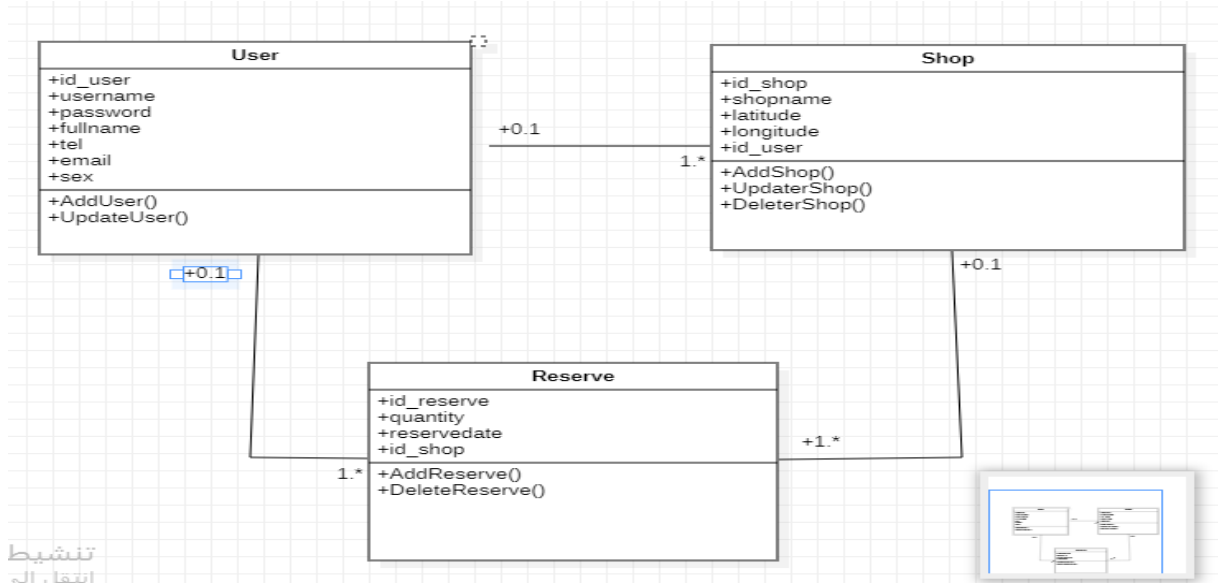


Figure 14:Diagramme de classe.

### 4.3. Outils et technologies utilisés pour notre implémentation:

Notre solution VRP est implémentée sous forme d'une application Web de géo localisation basée sur l'API Google et les technologies Web telles que : HTML, PHP, CSS3, en plus de MySQL qui sont utilisées pour la conception et le développement de l'application Web. Étant donné que notre application Web repose sur une approche client/serveur, Java Servlet est le meilleur choix pour implémenter notre application. D'autres technologies sont également

utilisées, telles que JQuery et JSON. Nous utilisons JQuery pour bénéficier de la mise à jour des informations sur une page Web en temps réel, sans rechargement de la page.

#### 4.3.1. L'API Google Maps :



Figure 15: le logo de google maps.

L'API Google Maps permet d'intégrer des cartes Google sur les pages Web des développeurs externes, en utilisant une interface JavaScript simple ou une interface Flash. Elle est conçue pour fonctionner aussi bien sur les appareils mobiles que sur les applications

de navigateur de bureau traditionnelles. L'API inclut la localisation linguistique pour plus de 50 langues, la localisation régionale et la géo codification, et dispose de mécanismes pour les développeurs d'entreprise souhaitant utiliser l'API Google Maps au sein d'un intranet. Les services HTTP de l'API peuvent être accessibles via une connexion sécurisée (HTTPS) par les clients de l'API Google Maps Premier.

#### 4.3.2. Jquery:



Figure 16: le logo jquery.

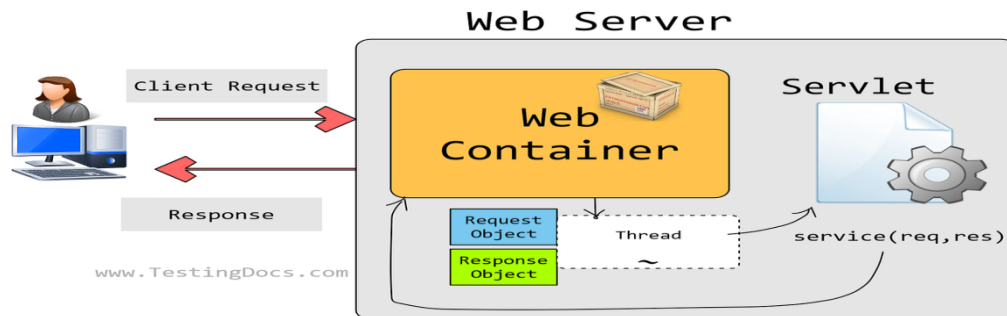
jQuery est une bibliothèque JavaScript qui permet aux développeurs web d'ajouter des fonctionnalités supplémentaires à leurs sites web. Elle est open source et fournie gratuitement sous la licence MIT. Au cours des dernières années, jQuery est devenue la bibliothèque JavaScript la plus populaire utilisée dans le développement web. Pour implémenter jQuery, un développeur web doit simplement faire référence au fichier JavaScript jQuery dans le HTML d'une page web. Certains sites hébergent leur propre copie locale de jQuery, tandis que d'autres font simplement référence à la bibliothèque hébergée par Google ou le serveur jQuery. Par exemple, une page web peut charger la bibliothèque jQuery en utilisant la ligne suivante dans la section <head> du HTML:

```
<script type="text/javascript" src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
```

Une fois que la bibliothèque jQuery est chargée, une page Web peut appeler n'importe quelle fonction jQuery prise en charge par la bibliothèque. Des exemples courants incluent la modification de texte, le traitement des données de formulaire, le déplacement d'éléments sur une page et l'exécution d'animations. Comme jQuery s'exécute côté client (plutôt que sur le serveur Web), il peut mettre à jour les informations sur une page Web en temps réel, sans rechargement de la page. Un exemple courant est "l'autocomplétion", où un formulaire de recherche affiche automatiquement les recherches courantes pendant que vous tapez votre requête. En fait, c'est ainsi que TechTerms.com fournit des suggestions de recherche lorsque vous tapez dans la zone de recherche. Outre sa licence gratuite, l'autre raison principale de la popularité de jQuery est sa compatibilité multi-navigateur. Étant donné que chaque navigateur

rend HTML, CSS et JavaScript différemment, il peut être difficile pour un développeur web de faire apparaître un site Web de la même manière sur tous les navigateurs. Au lieu de devoir écrire des fonctions personnalisées pour chaque navigateur, un développeur web peut utiliser une seule fonction jQuery qui fonctionnera dans Chrome, Safari, Firefox et Internet Explorer. Ce support multi-navigateur a incité de nombreux développeurs à passer du JavaScript standard à jQuery, car cela simplifie grandement le processus de codage.

### 4.3.3 Servlet :



**Figure 17:** Explication de servlet.

Un servlet est une classe Java qui permet de créer dynamiquement des données au sein d'un serveur HTTP. Ces données sont le plus souvent présentées au format HTML, mais elles peuvent également être au format XML ou tout autre format destiné aux navigateurs web. Les servlets utilisent l'API Java Servlet (package javax.servlet).

Un servlet s'exécute dynamiquement sur le serveur web et permet l'extension des fonctions de ce dernier, notamment l'accès aux bases de données, les transactions de commerce électronique, etc. Un servlet peut être chargé automatiquement lors du démarrage du serveur web ou à la première requête du client. Une fois chargés, les servlets restent actifs en attente d'autres requêtes clients.

L'utilisation de Servlet se fait via un conteneur de servlet (frame work) côté serveur. Il s'agit de l'environnement d'exécution du servlet qui lui permet de persister entre les requêtes des clients. L'API définit les relations entre le conteneur et le servlet. Le conteneur reçoit la requête du client et sélectionne le servlet qui devra la traiter. Le conteneur fournit également une gamme de services standard pour simplifier la gestion des requêtes et des sessions.

Un servlet, ou plusieurs servlets, sont des applications web Java. Leur principe de fonctionnement (architecture, fonctionnalités, configuration, déploiement) est décrit suivant une spécification officielle, dirigée par Sun Microsystems et à laquelle toute personne peut contribuer via les Requêtes de Spécification Java (Java Community Process). La version actuelle de la spécification des servlets est la 3.1.

#### 4.3.4. *My SQL:*



**Figure 18:** le logo MY SQL.

MySQL est un système de gestion de base de données relationnelle open source. Il est basé sur le langage de requête structuré (SQL), qui est utilisé pour ajouter, supprimer et modifier des informations dans la base de données. Les commandes SQL standard, telles que ADD, DROP, INSERT et UPDATE, peuvent être utilisées avec MySQL.

MySQL peut être utilisé pour une variété d'applications, mais on le trouve le plus souvent sur les serveurs Web. Un site Web qui utilise MySQL peut inclure des pages Web qui accèdent aux informations d'une base de données. Ces pages sont souvent appelées "dynamiques", ce qui signifie que le contenu de chaque page est généré à partir d'une base de données lorsque la page se charge. Les sites Web qui utilisent des pages Web dynamiques sont souvent appelés des sites Web pilotés par base de données. De nombreux sites Web pilotés par base de données qui utilisent .

MySQL utilisent également un langage de script Web comme PHP pour accéder aux informations de la base de données. Les commandes MySQL peuvent être incorporées dans le code PHP, permettant à tout ou partie d'une page Web d'être générée à partir des informations de la base de données. Étant donné que MySQL et PHP sont tous deux open source (ce qui signifie qu'ils sont gratuits à télécharger et à utiliser), la combinaison PHP/MySQL est devenue un choix populaire pour les sites Web pilotés par base de données.

#### 4.3.5. **WAMP:**

WAMP est l'acronyme de "Windows, Apache, MySQL et PHP". Il s'agit d'une variation de LAMP destinée aux systèmes Windows et est souvent installée sous forme de bundle logiciel (Apache, MySQL et PHP). Elle est fréquemment utilisée pour le développement web et les tests internes, mais peut également servir à héberger des sites web en direct.

La partie la plus importante du package WAMP est Apache (ou "Apache HTTP Server"), qui est utilisé pour exécuter le serveur web sous Windows. En exécutant un serveur web Apache local sur une machine Windows, un développeur web peut tester des pages web dans un navigateur web sans les publier en direct sur Internet.

#### 4.3.6. **NetBeans IDE:**

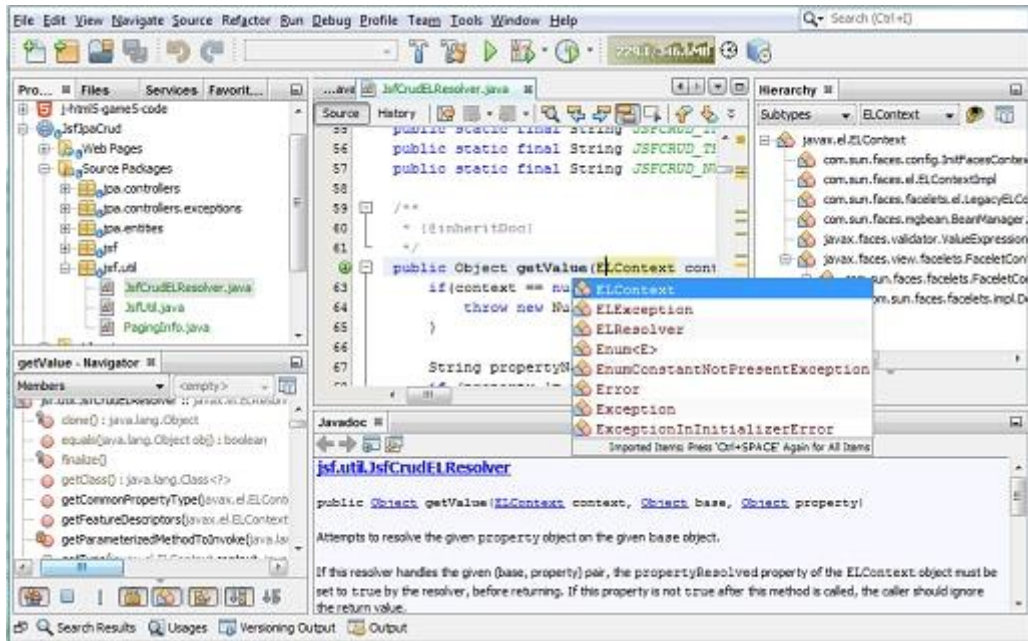


Figure 20: interface de netbeans.

NetBeans IDE est un environnement de développement intégré gratuit et open source pour le développement d'applications sur les systèmes d'exploitation Windows, Mac, Linux et Solaris.

L'IDE simplifie le développement d'applications web, d'entreprise, de bureau et mobiles utilisant les plates-formes Java et HTML5. L'IDE offre également une prise en charge du développement d'applications PHP et C/C++.

#### 4.4. Présentation du système:

Dans la section suivante, nous présenterons des captures d'écran des pages les plus importantes de notre application web de gestion

##### 4.4.1. Application web:

##### 4.4.1.1. Pour le client (Client):

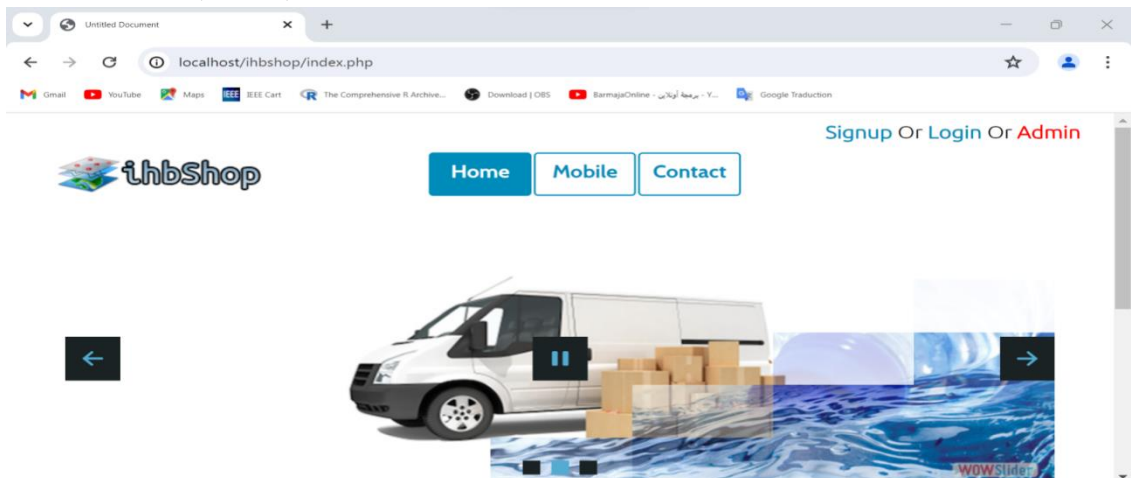


Figure 21: Interface de site web.

Signup Or Login Or Admin

Figure 22: Inscription et connexion.

Si vous êtes un nouvel utilisateur, vous devez vous inscrire en cliquant sur "Signup". Si vous êtes déjà inscrit, cliquez sur "Login" pour accéder. Si vous êtes un administrateur, cliquez sur "Admin" pour accéder au panneau de contrôle du site.



Figure 23: La barre de menu.

Profile : Pour voir et modifier les informations personnelles lorsque nécessaire

My shops : Pour voir et modifier les informations des boutiques lorsque nécessaire .

Réserve : Pour voir et modifier les réserves lorsque nécessaire.

Lorsque nous choisissons "My shops", nous obtenons:

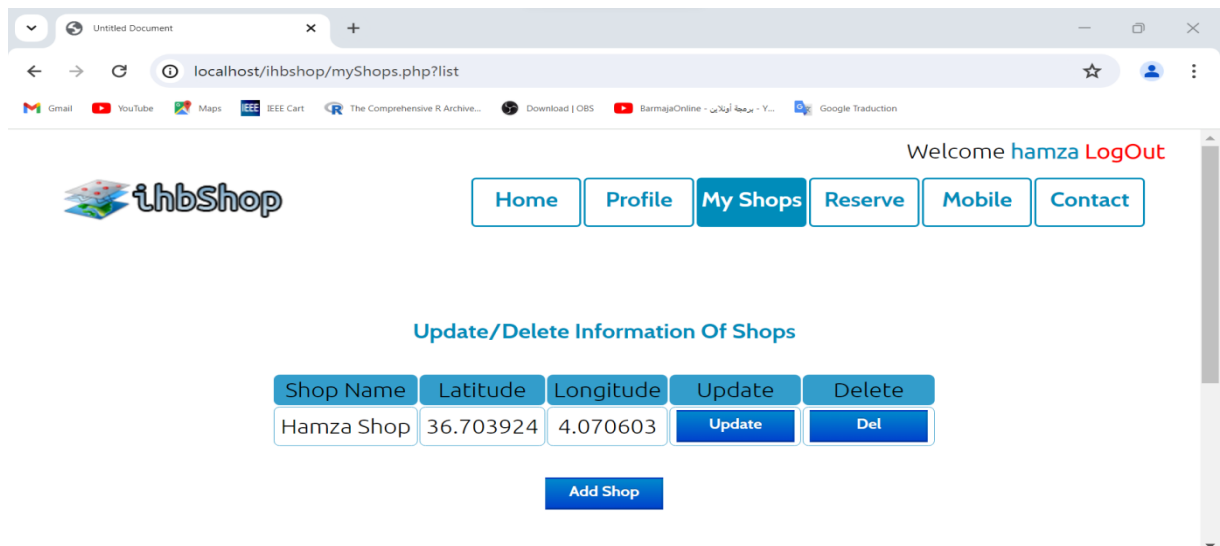


Figure 25: Réglages de Mes Boutiques ("My shops").

Lorsque nous cliquons sur le bouton "Add shop", nous obtenons:

The image shows a web form titled "Add Shop". It contains three text input fields stacked vertically, labeled "Shop Name", "Latitude", and "Longitude". Below these fields are two blue buttons: "Add" and "Reset".

**Figure 26:**Formulaire d'ajout de magasin("Add shop").

Pour ajouter un nouveau magasin, vous devez entrer le nom du magasin et les informations de localisation du magasin. et cliquer le bouton "Add".

Quand nous cliquons sur Réserver dans le menu du bar :

The image shows a web browser window displaying the "Reserve Form" page. The page has a navigation bar with buttons for "Home", "Profile", "My Shops", "Reserve", "Mobile", and "Contact". The main content area includes a "Quantity" input field, a "Select Shop To Reserve" dropdown menu with "Hamza Shop" selected, and a "Reserve" button. Below the form is a "My Reserve" link.

**Figure27:**Réglages de réservation.

#### 4.4.1.2. Pour l'administrateur :

The image shows a web browser window displaying the administrator control panel. The page has a navigation bar with a "LogOut" button. Below the navigation bar are five circular icons representing different settings: "User Setting", "Car Setting", "Shop Setting", "Setting", and "Delivery Setting". On the left side, there is a sidebar with a blue header "Outils" and three options: "Add User", "Update User", and "Delete User". The main content area has a large white box with the text "Select Outils To Show Settings ..".

**Figure 28:**Interface du panneau de contrôle administrateur.



Figure 29: Barre de menu du panneau de contrôle.

Après la réservation des clients et l'application de notre système basé sur GA, nous obtiendrons le résultat. Voici le résultat (ici, j'ai utilisé un exemple simple) :

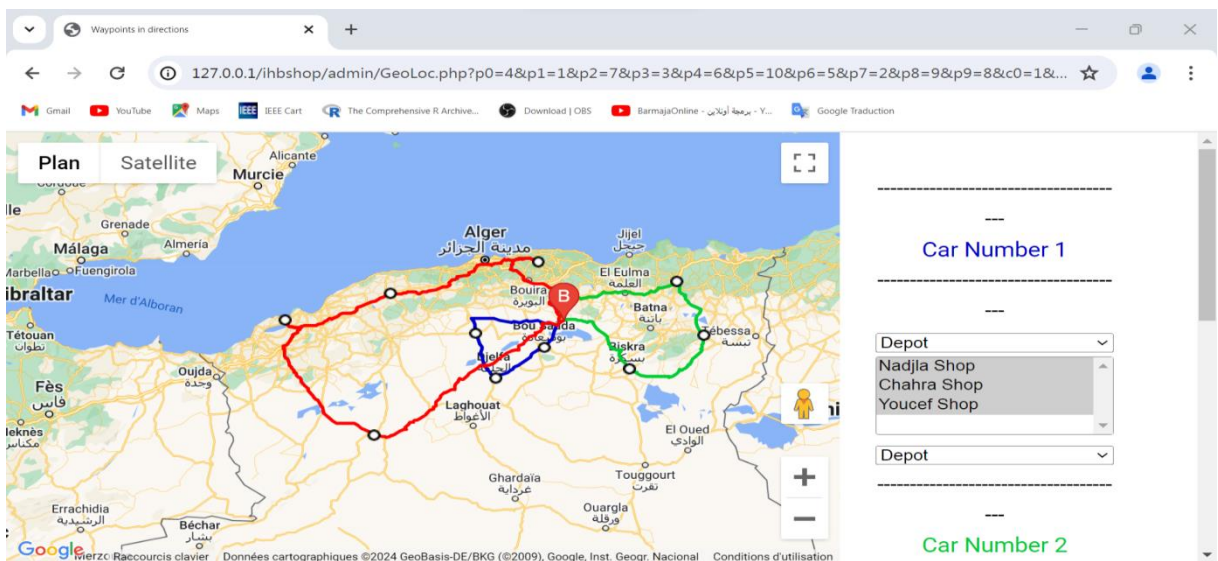


Figure 30: Exemple de livraison.

#### 4.5. Conclusion:

Dans ce dernier chapitre, nous avons présenté notre application, qui est une solution au problème de la distribution des véhicules. Notre approche est basée sur des algorithmes heuristiques, inspirés des algorithmes génétiques. Notre objectif est d'optimiser le temps pris par les véhicules sur le trajet le plus court et de satisfaire davantage de clients. La solution est soutenue par une application web qui aide le gestionnaire à répondre aux demandes des clients à moindre coût. De plus, nous avons mis en œuvre notre solution via une application web pour le côté client.

## **Conclusion Générale:**

Les problèmes de tournées de véhicules sont un élément essentiel de la vie quotidienne des décideurs et des planificateurs. Il s'agit de déterminer les tournées d'une flotte de véhicules pour livrer une liste de clients, ou pour effectuer des interventions (maintenance, réparation, contrôles) ou des visites (médicales, commerciales, etc.). L'objectif est de minimiser le nombre de véhicules, la distance et/ou le coût de livraison.

Dans le cadre de ce mémoire, nous avons examiné l'une des variantes importantes de ces problèmes, à savoir le problème des tournées de véhicules à capacité limitée. Toutes les demandes des clients sont connues à l'avance. Ce problème consiste à déterminer les tournées pour les véhicules qui transportent des marchandises d'un point d'origine à une destination donnée, tout en minimisant la distance totale des tournées.

Avant d'aborder la résolution de ce problème, nous avons présenté différentes problématiques de tournées de véhicules. Nous avons détaillé le Problème de Routage de Véhicules (VRP), qui représente le problème d'origine avec ses diverses variantes. Nous avons également proposé une formulation mathématique pour ce problème.

Ensuite, après avoir rappelé le principe et les caractéristiques des algorithmes génétiques, nous avons mis en œuvre une approche existante pour résoudre le problème du VRP. Nous avons développé une application web pour présenter les solutions obtenues.

## Bibliographie :

- [1] CesarRego et CatrineRoucairol, Le Problème de Tournées de Véhicules : étude et résolution approchée. INRIA, février 1994.
- [2] Raksmei PHAN, Une stratégie hybride pour le problème de tournées de véhicules avec Livraisons et collectes. Laboratoire d'Informatique, de Modélisation et d'optimisation des systèmes, septembre 2009.
- [3] H. Bigdeli, S.M.S. Mirdamadi, Adrian Deaconu, J. Tayyebi, Robust Vehicle Routing Problem in the Presence of a Proactive Attacker, 1998
- [4] Gendreau (M.), Hertz (A.) et Laporte (G.).-New Insertion and Post-Optimization Procedures for the Traveling Salesman Problem. - Rapport technique n° CRT-708, Centre de Recherche sur les Transports, Université de Montréal, 1992.
- [5] T.K. Ralphs, L. Kopman, W.R. Pulleyblank and L.E. Trotter, "On the capacitated vehicle routing problem", Mathematical Programming 94, 343-359, 2001.
- [6] P. Borne, G.D. Tanguy, J.P. Richard, F. Rotella, I. Zambettakis, Commande et optimisation de processus", Edition Technip, France, 1990
- [7] C. Rego et C. Roucairol. "Le problème de tournées de véhicules: Etude et Résolution Approchée". Technical Report, inria, Février 1994.
- [8] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. Introduction to algorithms, chapter 15, pages 350—355. The MIT Press, Cambridge, Massachusetts, 1990.
- [9] S. M. Johnson. Optimal two- and three-stage production schedules with setup times included. Naval Research Logistics Quarterly, 1 :61—68, 1954.
- [10] A. Le Bouthilier. "Modélisation UIVIL pour une architecture coopérative appliquée au problème de tournées de véhicules avec fenêtres de temps". Technical Report, Département d'informatique et de recherche opérationnelle. Faculté des arts et des sciences. Université de Montréal, Canada, Avril 2000.
- [11] L. Bodin and L. Berman. "Routing and scheduling of school buses by computer". Transportation Science 13(2), pages 113-129, 1979.
- [12] B. Goldberg, A. Assad, A. Levy and F. Gheysens "The fleet size and mix vehicle routing problem", Management Science and Statistics. Working paper, n° 82-020, 1982.

- [13] S. Kirkpatrick, C.D. Gelatt, et K. P. Vecchi. Optimization by simulated annealing. *science*. 220 :671–680, 1983.
- [14] V. Cerný. A thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1) :41–51, 1985.
- [15] N. Metropolis, M. N. Rosenbluth, et H. A. Teller. Equation of state calculation by fast computing machines. *Journal of Chemical Physics*, 21(6) :1087-1092, 1953.
- [16] <http://www.fsr.ac.ma/DOC/cours/maths/Souad%20Bernoussi/Cours%20C2SI.pdf> , Sidi Mohamed Douiri, Souad El Bernoussi, Halima Lakhbab, Cours des methods de resolution exactes heuristique et méthaheuristique, page:12 ,01/05/2024,13:04.
- [17] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13 :533–549, 1986.
- [18] P. Hansen. The steepest ascent mildest descent heuristic for combinatorial programming. *Congress on Numerical Methods in Combinatorial Optimization*, Capri, Italie, 1986.
- [19] F. Glover et M. Laguna. *Tabu search*. Kluwer Academic Publishers, Norwell, MA, second edition, 1997.
- [20] J. H. Holland. *Adaptation in natural and artificial systems*. University of Michigan press, 1975.
- [21] D. Goldberg. *Genetic algorithms*. Addison Wesley. Addison Wesley, ISBN : 0-201-15767-5, 1989a.
- [22] D. Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison Wesley, 1989b.
- [23] A. Colomi, M. Dorigo, et V. Maniezzo. An investigation of some proprieties of an "ant algorithm". In *Manner and Manderick*, pages 509–520, 1992b.
- [24] M. Dorigo, A. Colomi, et V. Maniezzo. The ant system : Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26, No.1 :29–41, 1996.
- [25] D. Costa et A. Hertz. Ants can color graphs. *Journal of the Operational Research Society*, 48 :295–305, 1997.

- [26] L. M. Gambardella, E. D. Taillard, et M. Dorigo. Ant colonies for the quadratic assignment problem. *Journal of the Operational Research Society*, 50(2) :167–176, 1999.
- [27] S. Mazzeo et I. Loiseau. An ant colony algorithm for the capacitated vehicle routing problem. *Electronic Notes in Discrete Mathematics*, 18 :181–186, 2004. (Cité page 23.)
- [28] A. Colomi, M. Dorigo, et V. Maniezzo. Distributed optimization by ant colonies. invarela, f. and bourgine, p., editors, proceedings of ECAL'91. First European Conference on Artificial Life, Paris, France, pages 134–142, 1992a.
- [29] J. Kennedy et R. Eberhart. Particle swarm optimization, proceedings of iee international conference on neural networks. doi:10.1109/ICNN.1995.488968 :1942–1948, 1995.
- [30] F. Glover, G.A. Kochenberger, et B. Alidaee. Adaptive memory tabu search for binary quadratic programs. *Management Science*, 44 :336–345, 1998.
- [31] M. Haj-Rachid, C. Bloch, W. Ramdane-Cherif, and P. Chatonnay. Différents opérateurs évolutionnaires de permutation : sélections, croisements et mutations. Research Report RR2010-07, LIFC - Laboratoire d'Informatique de l'Université de Franche Comté, July 2010. 25, 26, 30
- [32] T.G. Crainic and F. Semet. Recherche opérationnelle et transport de marchandises. In *Optimisation combinatoire. 3, Applications*. Hermès Science : Lavoisier, 2006.
- [33] M.B. Baker and M.A. Ayechev. A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, 30(5) :787 , 800, 2003.
- [34] C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12) :1985 2002, 2004.
- [35] J. Berger, M. Salois, and R. Begin. A hybrid genetic algorithm for the vehicle routing problem with time windows. In Robert Mercer and Eric Neufeld, editors, *Advances in Artificial Intelligence*, volume 1418 of *Lecture Notes in Computer Science*, pages 114127. Springer Berlin / Heidelberg, 1998. 30.
- [36] J. Louis Sushil and Tang Rilun. Interactive genetic algorithms for the traveling salesman problem. In *Proceedings of the 1999 Genetic and Evolutionary Computation Conference (GECCO 1999)*, volume 1, pages 385 392, Orlando, Florida, USA, 1999
- [37] H. Housroum. Une approche génétique pour la résolution du problème VRPTW dynamique. PhDthesis, Université d'Artois, 2005. 23, 31.