

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE : *Mathématiques et Informatique*

DEPARTEMENT : *Informatique*

N° : .....



DOMAINE : Mathématiques et Informatique

FILIERE : Informatique

OPTION : Systèmes d'Information et Génie  
Logiciel (SIGL)

**Mémoire de fin d'études présenté pour l'obtention  
du diplôme de Master Académique**

**Par :**

BELMENAOUAR Manel

LATRECHE Ibtihal

**Intitulé**

**Problème de Localisation et de Routage**

**Soutenu publiquement le 12/06/2024 devant le jury composé de :**

Dr. MOUSSAOUI Adel

Université MB de M'sila

Président

Pr. HEMMAK Allaoua

Université MB de M'sila

Rapporteur

Dr. OULDMOHAMEDJ Nejb

Université MB de M'sila

Examineur

Année universitaire : 2023 /2024

## ***Remerciements***

*Nous remercions **Dieu** pour nous avoir donné la force d'accomplir ce modeste travail pour aller plus loin.*

*Nous tenons, à exprimer notre profonde gratitude à monsieur **ALLAOUA HEMMAK**, Docteur à l'université de M'sila, qui a assumé la direction de ce travail. Qu'il veuille bien trouver ici l'expression de ma reconnaissance pour son dévouement, sa patience, sa disponibilité, ses conseils et son aide constant qu'il nous 'a apporté tout au long de ce travail.*

*Nous tenons, à remercier toutes les personnes qui ont contribué au succès de réalisé nos études et qui nous aidées lors de la rédaction de ce mémoire.*

*Nos sincères remerciement aux **membres du jury** pour l'honneur qu'ils nous ont fait pour accepter le jugement sur ce modeste travail et sur tous leurs commentaires pertinents.*

*Enfin, nous espérons que ce travail aura la valeur souhaitée.*

## SOMMAIRE

Introduction générale .....	1
Chapitre 1 : Optimisation et les approches d'optimisation.	
Introduction.....	3
1. Optimisation combinatoire .....	3
1.1. L'optimisation.....	3
1.2. L'optimisation Combinatoire .....	3
1.3. Problème Combinatoire .....	4
1.4. Modélisation d'un problème d'optimisation.....	5
1.5. Caractéristique d'un Problème D'optimisation .....	5
1.6. Exemples De Problèmes D'optimisation Combinatoire.....	6
1.6.1. Le problème du sac-à-dos .....	6
1.6.2. Problème du voyageur de commerce.....	9
1.6.3. Problème De 8-Queen.....	11
1.6.4. Problème d'ordonnancement .....	12
1.7. Domaines d'application d'optimisation combinatoire.....	13
1.8. Problème de localisation du routage est un problème d'optimisation .....	13
1.8.1 Présentation du LRP .....	14
1.8.2. LRP est un problème d'optimisation.....	15
1.8.3. Application réelle du Problème de Routage de Localisation.....	15
1.8.4. Historique du LRP .....	16
1.8.5. L'objectif de LRP.....	17
1.9. Méthodes de résolution d'un problème d'optimisation .....	18
1.9.1. Méthode Exacte.....	18
1.9.1.1. Procédure par Séparation et Évaluation.....	19

1.9.1.2. Programmation dynamique .....	19
1.9.1.3. La programmation linéaire .....	20
1.9.2. Méthodes Approchées .....	20
1.9.2.1. Méthodes Heuristiques .....	20
1.9.2.2. Méthodes constructives .....	20
1.9.2.3 Méthodes de recherche locale (amélioration) .....	21
1.9.3. Méthode exacte vs Méthode heuristique .....	21
Conclusion .....	22

## **Chapitre 2 : Etat de l'art**

2.1. Introduction.....	22
2.2. Revue de littérature .....	22
2.3. Conclusion.....	26

## **Chapitre 3 : Résolution du problème de localisation du routage**

Introduction .....	27
3.1. L'énoncé du problème.....	27
3.1.1. Données.....	28
3.1.2. Programme.....	28
3.1.3. L'objectif.....	29
3.2. La problématique.....	29
3.3. Motivation pour étudier le (LRP) .....	30
3.4. La Méthode De Résolution (Recuit Simulé) .....	30
3.4.1. Recuit simulé .....	30
3.4.2. Pourquoi la méthode de recuit simulé .....	31
3.4.3. Principe De La Méthode Du Recuit Simulé.....	32
4.4. Formulation mathématique .....	33
3.5. Organigramme Général Du Recuit Simulé .....	35
3.6. Algorithme du recuit simulé .....	36
3.6. Avantages Et Inconvénients Du Recuit Simulé .....	36

Conclusion .....	37
------------------	----

## **Chapitre 4 : Implémentation et résultats**

Introduction .....	38
4.1. Environnement .....	38
4.1.1. Environnement matérielle .....	38
4.1.2. Environnement logiciel .....	38
4.2. Langage de programmation.....	40
4.3. Les jeux de données utilisées .....	41
4.4. Structure des fichiers de test .....	43
4.5. Exemple d'un résultat obtenu .....	47
4.6 La comparaison .....	48
4.6.1. Comparaison des taux de refroidissement .....	48
4.6.2. Comparaison des températures initiales.....	49
4.6.3. Comparaison des paliers .....	53
4.7. Analyse des résultats .....	58
Conclusion.....	59
Conclusion générale .....	60

## Listes des figures

<b>Figure 1-1</b> : Problème combinatoire .....	4
<b>Figure 1-2</b> : Problème du sac à dos .....	6
<b>Figure 1-3</b> : Arbre de recherche .....	9
<b>Figure 1-4</b> : Exemple problème 8-Queen.....	12
<b>Figure 1-5</b> : LRP et sa relation avec FLP et VRP .....	14
<b>Figure01-6</b> : Les différentes méthodes de résolution en optimisation combinatoire .....	18
<b>Figure3-1</b> : Organigramme général du recuit simulé .....	35
<b>Figure 4-1</b> : Navigateur Anaconda.....	39
<b>Figure 4-2</b> : Logo de Jupyter Notebook .....	40
<b>Figure 4-3</b> : Langage (python) .....	41
<b>Figure 4-4</b> : Page source des instances.....	42
<b>Figure 4-5</b> : Liste de fichiers de test proposés par Barreto .....	43
<b>Figure 4-6</b> : Un fichier de test .....	44
<b>Figure 4-7</b> : Structure d'un fichier de test .....	46
<b>Figure 4-8</b> : Résultats de recuit simule.....	47
<b>Figure4-9</b> : Ajustement du taux de réduction.....	52
<b>Figure4-10</b> : Ajustement de la température initiale.....	55
<b>Figure4-11</b> : Ajustement du palier de refroidissement.....	58

## Listes des tableaux

<b>Tableau 1.1</b> : Nombre de tournées possibles et temps de calcul estimé .....	10
<b>Tableau 1.2</b> : Méthode exacte vs méthode heuristique .....	21
<b>Tableau 4.1</b> : Comparaison des taux de refroidissement.....	49
<b>Tableau 4.2</b> : Comparaison des températures initiales.....	53
<b>Tableau 4.3</b> : Comparaison des paliers .....	56



**Introduction générale**

### Introduction générale

Le domaine de la recherche opérationnelle constitue aujourd'hui un puissant outil de soutien aux entreprises pour collecter des données exhaustives, explorer toutes les options disponibles, anticiper tous les résultats possibles et estimer les risques. Il peut être adapté à des opérations commerciales spécifiques ou à des cas d'utilisation pour choisir les meilleures techniques de résolution de problèmes et orienter les décisions vers les résultats souhaités ou optimaux.

La recherche opérationnelle (RO) représente un ensemble d'outils et de techniques scientifiques utilisés pour développer des solutions optimales à une variété de problèmes réels. Ces solutions comprennent des programmes mathématiques visant à améliorer ou à atteindre certains critères tout en tenant compte des contraintes connues, et sont appliquées dans des domaines tels que l'économie, la finance et la logistique. La recherche opérationnelle relève des mathématiques appliquées et vise à trouver les meilleures solutions pour des problèmes complexes.

L'optimisation joue un rôle crucial dans le domaine de la recherche opérationnelle, visant à améliorer les solutions disponibles pour répondre aux besoins de l'utilisateur de la meilleure manière possible. L'optimisation varie entre les approches continues et discrètes, la première se concentrant sur l'amélioration des valeurs dans des domaines continus, tandis que la seconde cherche à choisir les meilleures solutions parmi un ensemble d'options.

L'optimisation combinatoire vise à sélectionner les meilleures solutions en fonction de critères spécifiques parmi un grand nombre d'options possibles, et son défi réside dans la résolution de problèmes complexes appartenant à la classe des problèmes NP-difficiles, pour lesquels il n'existe généralement pas de solutions générales en temps raisonnable. Les solutions d'optimisation combinatoire sont classées en méthodes exactes et en méthodes approchées, ces dernières étant utilisées pour choisir des solutions plus rapidement sur la base de l'expérience et des connaissances antérieures.

Parmi ces problèmes, la distribution des véhicules dans un entrepôt unique et la détermination du site idéal pour une nouvelle installation sont parmi les plus éminents, étant faciles à comprendre mais extrêmement difficiles à résoudre. Bien que les chercheurs aient développé plusieurs algorithmes pour résoudre ces problèmes, il n'existe généralement pas de solution précise, ce qui nous amène à nous fier aux meilleures solutions disponibles.

L'application par les entreprises de solutions efficaces à ces problèmes peut renforcer leur efficacité et réduire leurs coûts, les rendant ainsi plus compétitives sur le marché. De plus, ces

## Introduction Générale

---

solutions peuvent contribuer à améliorer l'utilisation des ressources et à réduire les pertes, ce qui favorise la croissance économique et renforce la compétitivité tant au niveau national qu'international, et donc à renforcer l'économie dans son ensemble. La résolution du problème de la distribution des véhicules constitue un élément essentiel des stratégies des entreprises prospères dans un environnement économique concurrentiel.

Le présent rapport s'articulera autour de quatre chapitres principaux :

- Le premier chapitre sera consacré à la présentation de quelques concepts liés au domaine d'optimisation ainsi qu'un récapitulatif des différentes approches d'optimisation.
- Comme nous le savons, la science est la continuation de ce qui a précédé. Par conséquent, la première étape de toute recherche consiste à examiner les recherches antérieures pour connaître les dernières découvertes des chercheurs. Dans le deuxième chapitre, nous présenterons un état de l'art sur le problème général de localisation du routage.
- Le troisième chapitre est considéré comme une passerelle entre le contexte présenté et les contributions de la thèse, en exposant les motivations qui ont conduit à ce travail et en clarifiant la méthode suivie pour résoudre le problème en énonçant ses expressions mathématiques et son algorithme spécifique.
- Dans le dernier chapitre nous présenterons les résultats de l'implémentation des principes de l'approche décrit dans le troisième chapitre ainsi que la structure des jeux de donnée utilisées pour tester les concepts de l'approche proposée. Enfin, nous terminerons ce mémoire par une conclusion, où nous évoquerons les principaux apports de ce travail.



**Chapitre 1**  
**Optimisation et les approches**  
**d'optimisation**

# Chapitre 1 : Optimisation et les approches d'optimisation

## Introduction

L'analyse et la résolution des problèmes significatifs de notre vie quotidienne sont assez délicates, car le nombre limité de solutions possibles augmente généralement avec la taille et la complexité du problème. C'est là que l'optimisation, et en particulier l'optimisation combinatoire, intervient. Son rôle consiste à traiter les problèmes de choix parmi un ensemble large mais limité d'alternatives pour trouver la meilleure solution selon un critère spécifique. Par exemple, dans la vie quotidienne, on peut être confronté au choix du meilleur itinéraire pour se rendre au travail, avec de nombreuses options possibles et la nécessité de choisir la meilleure en fonction de facteurs tels que le temps, la distance ou la sécurité. Dans ce chapitre, nous nous concentrerons sur les concepts fondamentaux liés à notre sujet, comme les problèmes d'optimisation et les différentes méthodes de résolution, afin de mettre en évidence le contexte général dans lequel notre sujet s'inscrit.

## 1 Optimisation combinatoire

### 1.1 L'optimisation

Commençons d'abord par le terme « optimisation ». Dans son sens le plus large, il consiste à trouver la meilleure solution possible à un problème donné. Cela peut impliquer la réduction ou l'augmentation d'une certaine quantité appelée la fonction objectif, tout en respectant un ensemble de conditions appelées contraintes. Par exemple, dans le contexte de l'entreprise, l'optimisation peut impliquer soit :

Réduire les coûts de production tout en augmentant les bénéfices ou la satisfaction client.

Réduire les coûts de stockage des produits tout en respectant les délais de livraison et la capacité de stockage.

Donc, elle doit rechercher le meilleur compromis répondant à ces conditions. [1]

### 1.2 L'optimisation Combinatoire

L'optimisation combinatoire est une branche des mathématiques appliquées et de l'informatique qui concerne le choix de la meilleure option parmi un ensemble d'alternatives favorables. Elle est également liée à la recherche opérationnelle, aux algorithmes et à la théorie de la complexité. De nombreuses applications peuvent être modélisées comme des

problèmes d'optimisation combinatoire, tels que le problème du voyageur de commerce, l'ordonnancement des tâches et la coloration des graphes. [2]

### 1.3 Problème Combinatoire

Une problématique combinatoire se définit comme une situation nécessitant de trouver une solution qui concorde avec un ensemble de conditions ou de contraintes. La résolution de ce type de problème consiste à trouver un compromis entre ces contraintes de manière à maximiser certains objectifs tout en minimisant l'impact d'autres. Cependant, les contraintes dans les problèmes combinatoires peuvent être contradictoires. Par exemple, nous pourrions vouloir acheter une voiture moderne de qualité tout en respectant un prix raisonnable qui ne dépasse pas un certain seuil. Si l'accent est mis sur le premier aspect (voiture de qualité), cela peut entraîner une augmentation du prix, tandis que si l'accent est mis sur le deuxième aspect (prix raisonnable), cela pourrait entraîner un compromis sur la qualité de la voiture. Cet exemple illustre la difficulté de satisfaire simultanément toutes les contraintes, soulignant le défi auquel les individus sont confrontés lorsqu'ils tentent de trouver un équilibre entre ces considérations contradictoires. [3]

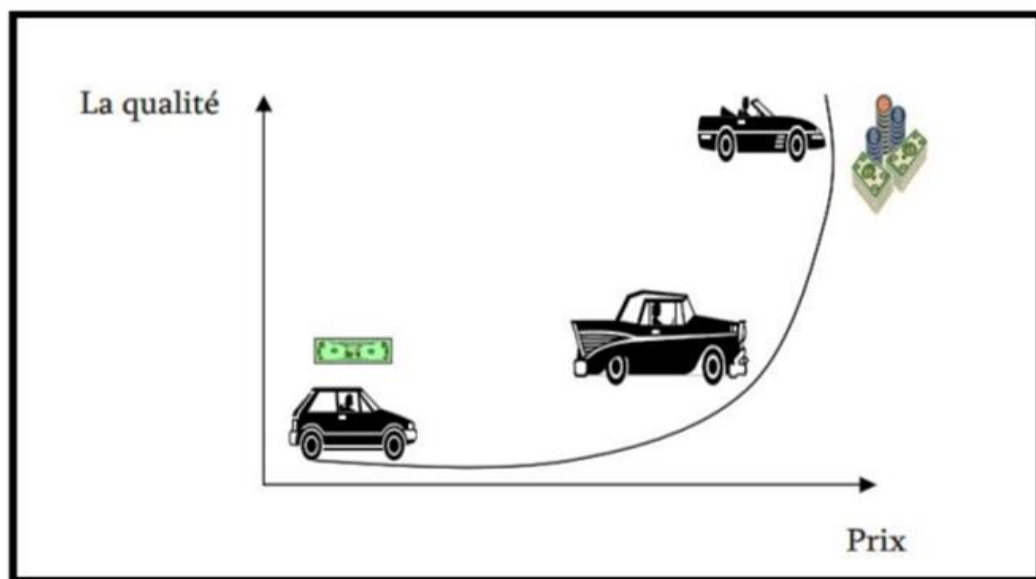


Figure 1.1 : Concept d'optimisation. [4]

### 1.4 Modélisation d'un problème d'optimisation :

Le processus de résolution de problème commence toujours par l'identification et la description précise du problème, ce qu'on appelle la modélisation. Dans le cas des problèmes d'optimisation, le problème est représenté à l'aide d'un ensemble d'éléments.

Tout d'abord, le problème est noté  $P(S, f)$ , où  $P$  fait référence au problème lui-même,  $S$  à l'ensemble des solutions possibles au problème, et  $f$  à la fonction objectif utilisée pour évaluer chaque solution. La fonction objectif  $f$  est la fonction utilisée pour mesurer la qualité de chaque solution potentielle. Cette fonction est également connue sous le nom de fonction de fitness, et elle attribue une valeur à chaque solution, exprimant dans quelle mesure cette solution atteint l'objectif souhaité. En utilisant cette modélisation, nous sommes en mesure d'analyser et de comprendre le problème de manière plus approfondie, et d'appliquer les méthodes et techniques appropriées pour trouver les solutions optimales qui répondent aux objectifs du problème de la meilleure façon possible. [2]

### 1.5 Caractéristique d'un Problème D'optimisation :

Les caractéristiques d'un problème d'optimisation comprennent plusieurs aspects clés qui définissent la nature du problème et influencent la manière dont il est abordé. Voici quelques-unes de ces caractéristiques :

- Espace d'état : Le problème d'optimisation est défini par un ensemble d'états, représentant les différentes solutions possibles au problème.
- Fonction objectif : Le problème comprend une ou plusieurs fonctions objectifs qui définissent ce que l'on cherche à optimiser. Ces fonctions peuvent être de nature linéaire, quadratique, non linéaire, etc.
- Variables de décision : Ces variables représentent les paramètres ou les choix qui peuvent être ajustés pour influencer la solution du problème. Elles peuvent être de nature continue, entière, binaire, etc.
- Contraintes : Les contraintes définissent les conditions ou les limites que les solutions doivent respecter. Elles peuvent être des contraintes d'égalité ou d'inégalité et contribuent à restreindre l'espace des solutions possibles.
- Méthode d'optimisation : Il s'agit de l'approche ou de l'algorithme utilisé pour rechercher la meilleure solution possible dans l'espace des solutions. Cette méthode vise à trouver un ou plusieurs points qui optimisent la fonction objectif tout en respectant les contraintes. [5]

### 1.6 Exemples De Problèmes D'optimisation Combinatoire :

#### 1.6.1 Le problème du sac-à-dos :

L'énoncé de ce problème fameux est simple : consiste à remplir un sac à dos de capacité  $P$  avec plusieurs objets  $\{x_1, x_2, \dots, x_n\}$ , tel que chaque objet possède un poids  $\{p_1, p_2, \dots, p_n\}$ ,

et un cout  $\{c_1, c_2, \dots, c_n\}$ . L'objectif c'est trouver un sous ensemble de  $x$  tel que son poids est borné par  $P$  et son bénéfice soit maximum, c'a veut dire le poids final de sac à dos ne dépasse pas le poids maximal  $P_{\max}$  et la somme totale de la valeur des objets placés dans le sac à dos soit maximal (la plus grande possible).

$$\begin{aligned} \max \sum_{1 \leq i \leq 4} x_i c_i \\ \sum_{1 \leq i \leq 4} x_i p_i < p \\ x_i \in \{0,1\} \end{aligned}$$

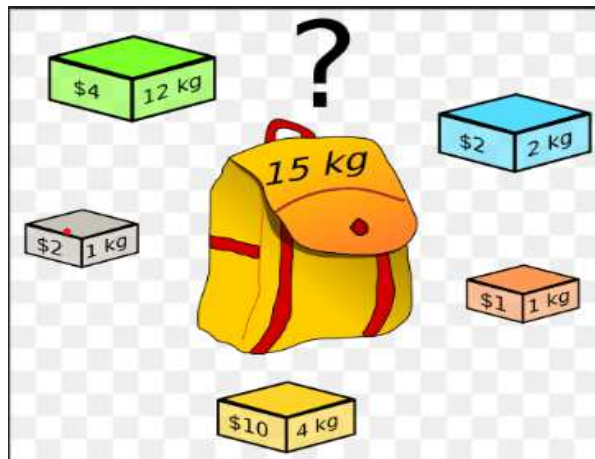


Figure 1.2 : Problème du sac à dos.

Pour résoudre ce problème, Il existe deux grandes catégories de méthodes de :

- Méthode approchée :

Comme pour la plupart des problèmes NP-complets, il peut être intéressant de trouver des solutions réalisables mais non optimales. De préférence avec une garantie sur l'écart entre la valeur de la solution trouvée et la valeur de la solution optimale. On appelle efficacité d'un objet le rapport de sa valeur sur son poids. Plus la valeur de l'objet est importante par rapport à ce qu'il consomme, plus l'objet est efficace. Voici un exemple d'algorithme de ce type :

- calculer le rapport  $(v_i / p_i)$  pour chaque objet  $i$  ;
- trier tous les objets par ordre décroissant de cette valeur ;
- sélectionner les objets un à un dans l'ordre du tri et ajouter l'objet sélectionné dans

le sac si le poids maximal reste respecté. Déroulons cet algorithme sur notre exemple : Pour quatre objets ( $n = 4$ ) et un sac à dos d'un poids maximal de 30 kg ( $P = 30$ ), nous avons par exemple les données suivantes :

Objets	1	2	3	4
$V_i$	7	4	3	3
$P_i$	13	12	8	10

**Première étape:**

Objets	1	2	3	4
$v_i / p_i$	0,54	0,33	0,37	0,30

**Deuxième étape:**

Objets	1	3	2	4
$v_i$	7	3	4	3
$p_i$	13	8	12	10
$v_i / p_i$	0,54	0,37	0,33	0,30

**Troisième étape :** Objet 1 : on le met dans le sac vide, le poids du sac est alors de 13 et inférieur à 30 ; Objet 3 : on le met dans le sac car  $13 + 8 = 21$  est inférieur à 30 ; Objet 2 : on ne le met pas dans le sac car le poids total (33) dépasserait 30. Objet 4 : on ne le met pas dans le sac (poids total de 31).

La solution trouvée est donc de mettre les objets 1 et 3 dans le sac, ce qui donne une valeur de 10. Cette solution n'est pas optimale, puisqu'une solution avec une valeur totale de 11 existe. Néanmoins, cet algorithme reste rapide même si le nombre d'objets augmente considérablement.

Ce type d'algorithme est aussi appelé algorithme glouton, car il ne remet jamais en cause une décision prise auparavant. Ici, lorsque l'objet 2 ne peut pas entrer dans le sac, l'algorithme n'essaie pas d'enlever l'objet 3 du sac pour y mettre l'objet 2 à sa place.

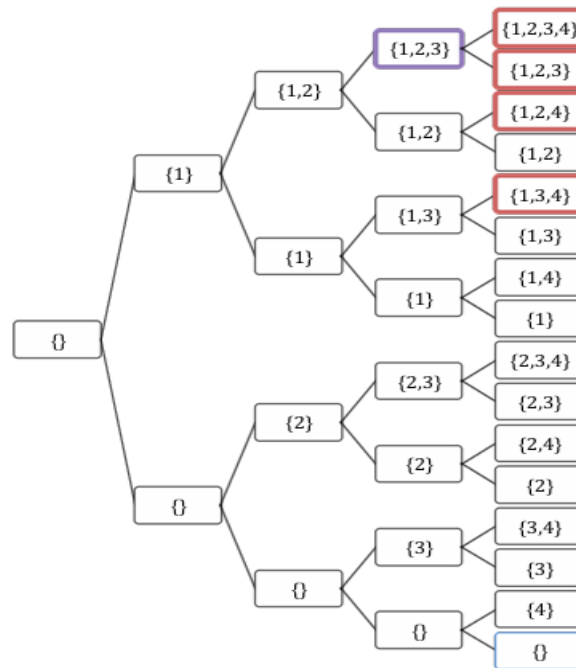
- Méthode exacte :

Pour trouver la solution optimale et être certain qu'il n'y a pas mieux, il faut utiliser une méthode exacte, qui demande un temps de calcul beaucoup plus long (si le problème est difficile à résoudre). Il n'existe pas une méthode exacte universellement plus rapide que toutes les autres. Chaque problème possède des méthodes mieux adaptées que d'autres. Nous allons présenter un exemple d'algorithme de ce type, nommée procédure par séparation et

évaluation (PSE), ou en anglais branch and bound. Nous n'exposerons ici qu'une version simplifiée d'une PSE.

Une PSE est un algorithme qui permet d'énumérer intelligemment toutes les solutions possibles. En pratique, seules les solutions potentiellement de bonne qualité seront énumérées, les solutions ne pouvant pas conduire à améliorer la solution courante ne sont pas explorées.

Pour représenter une PSE, nous utilisons un « arbre de recherche » constitué de nœuds ou sommets, où un nœud représente une étape de construction de la solution et d'arcs pour indiquer certains choix faits pour construire la solution. Dans notre exemple, les nœuds représentent une étape pour laquelle des objets auront été mis dans le sac, d'autres auront été laissés en dehors du sac, et d'autres, encore, pour lesquels aucune décision n'aura encore été prise. Chaque arc indique l'action de mettre un objet dans le sac ou, au contraire, de ne pas le mettre dans le sac. La figure suivante représente l'arbre de recherche du problème donné en exemple.



**Figure 1.3** : Arbre de recherche du KSP.

L'algorithme de recherche utilisé dans le problème du sac à dos utilise la technique des bornes inférieures et supérieures pour réduire la taille de l'arbre de recherche. À chaque nœud exploré, les bornes inférieure et supérieure sont calculées. La borne inférieure est utilisée pour éliminer les solutions faibles et réduire la taille de l'arbre exploré, tandis que la borne supérieure permet de réduire le nombre de nœuds à explorer en coupant les branches de l'arbre lorsque la borne supérieure d'un nœud est inférieure à la borne inférieure. La borne

inférieure est mise à jour lorsqu'une meilleure solution est trouvée. Ces mesures contribuent à réduire le temps de recherche et à améliorer l'efficacité de l'algorithme.

### 1.6.2 Problème du voyageur de commerce

Le problème du voyageur de commerce c'est un problème classique de la recherche opérationnelle. Il consiste à trouver le chemin le plus court pour relier un ensemble donné de villes séparées par des distances données, en passant par chaque ville exactement une seule fois, telle que on commence par n'importe quelle ville et on termine en retournant à la ville de départ. L'objectif est de minimiser la distance totale parcourue tout en respectant ces contraintes.

La difficulté de ce problème est de trouver l'ordre dans lequel chacun des villes sera visite, en minimisant un certain critère (temps ou cout du parcours, ou bien longueur totale parcourue . . .). [3]

**Remarque :** Le voyageur de commerce peut être remplacé par un véhicule, postier, médecin, appareil ou machine, .....et les villes peuvent être remplacées par des clients, machines, hôpitaux, malades, foyers.

Le problème du voyageur de commerce (PVC) appartient à la classe des problèmes NP-difficiles. Cela signifie qu'il n'existe pas d'algorithme polynomiale efficace et rapide pour calculer la solution. La complexité est donc de l'ordre de  $O(n!)$  où  $n$  est le nombre de villes. Donc le nombre de tournées possibles est  $(n-1)!/2$ . Ce facteur est dû au fait que chaque tournée et son inverse (parcours dans le sens inverse) sont considérés comme identiques. Si nous supposons que le temps nécessaire pour évaluer une seule tournée est de 1 seconde, nous pouvons observer la croissance exponentielle du temps de calcul en fonction du nombre de villes. Voici un tableau illustrant cette explosion combinatoire : [6]

Nombre de ville (n)	Nombre de tournée possible $(n-1)!/2$	Temps nécessaire (en secondes)	Temps nécessaire (en jours)
4	3	3	0.0000347
5	12	12	0.0001389
6	60	60	0.0006944
7	360	360	0.0041667
8	2520	2520	0.029167
9	20160	20160	0.233333
10	181440	181440	2.1
11	1814400	1814400	21
12	19958400	19958400	231.2
13	239500800	239500800	2771.3
14	3113510400	3113510400	36018.9
15	43589145600	43589145600	504587.5
20	60822550204416000	60822550204416000	703687441776.0

**Tableau 1.1** : Nombre de tournées possibles et temps de calcul estimé. [7]

**C'est-à-dire :**

**1. Nombre de villes n=4 :**

- Nombre de t tournées possibles :  $(4-1)!/2 = 3!/2 = 3$ .
- Temps nécessaire pour évaluer toutes les tournées : 3 secondes.

**2. Nombre de villes n=10 :**

- Nombre de t tournées possibles :  $(10-1)!/2 = 9!/2 = 181440$ .
- Temps nécessaire pour évaluer toutes les tournées : 181440 secondes (environ 2,1 jours).

**3. Nombre de villes n=15 :**

- Nombre de t tournées possibles :  $(15-1)!/2 = 14!/2 = 43589145600$ .
- Temps nécessaire pour évaluer toutes les tournées : 43589145600 secondes (environ 504587,5 jours ou environ 1382 ans).

**4. Nombre de villes n=20 :**

- Nombre de t tournées possibles :  $(20-1)!/2 = 19!/2$ .

- Calcul (19 !) :  

$$19! = 19 \times 18 \times \dots \times 1 = 121645100408832000.$$
- Nombre de tournée possible  $(n-1)!/2$  :  

$$19! / 2 = 121645100408832000 / 2 = 60822550204416000.$$
- Temps nécessaire pour évaluer toutes les tournées (en secondes) :  
 60822550204416000 secondes.
- Temps nécessaire pour évaluer toutes les tournées (en jours) :  

$$60822550204416000 \text{ secondes} / 60 \times 60 \times 24 \approx 703687441776.0 \text{ jours.}$$
- Temps nécessaire pour évaluer toutes les tournées (en années) :  

$$703687441776.0 \text{ jours} / 365 \approx 1929012320.0 \text{ années.}$$

### 1.6.3 Problème des 8-Queens

Dans le problème des huit reines, nous avons un plateau d'échecs standard (8 par 8) et huit reines doivent être placées sur le plateau de manière à ce qu'aucune reine ne se touche. Ce problème peut être généralisé facilement pour inclure le problème des N reines. De nombreuses approches classiques en intelligence artificielle appliquées à ce problème fonctionnent de manière constructive ou incrémentielle : on commence par placer une reine, puis on place les autres. On tente de placer la  $(n^{\text{ème}}+1)$  reine dans la meilleure position possible, c'est-à-dire une position où la nouvelle reine ne peut pas toucher les autres. Généralement, une sorte de mécanisme de retour en arrière est appliqué : s'il n'y a pas de position possible pour la  $(n^{\text{ème}} + 1)$  reine, la  $n^{\text{ème}}$  reine est déplacée vers une autre position. [3]

18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14	♛	13	16	13	16
♛	14	17	15	♛	14	16	16
17	♛	16	18	15	♛	15	♛
18	14	♛	15	15	14	♛	16
14	14	13	17	12	14	12	18

Figure 1.4 : Exemple : problème des 8-Queens [3]

### 1.6.4 Problème d'ordonnancement

Le problème d'ordonnancement consiste à organiser et à répartir efficacement les tâches ou les activités dans le temps, en attribuant des ressources limitées à ces tâches tout en respectant un ensemble de contraintes. L'objectif principal est généralement d'optimiser un ou plusieurs critères, tels que la minimisation du temps total de traitement, la maximisation de l'utilisation des ressources, ou la minimisation des coûts.

En d'autres termes, le problème d'ordonnancement implique de déterminer l'ordre d'exécution des tâches, les moments auxquels elles doivent être réalisées, ainsi que la manière dont les ressources disponibles doivent être allouées pour accomplir ces tâches dans les délais impartis et de la manière la plus efficace possible. Les contraintes peuvent inclure des dépendances entre les tâches, des contraintes de capacité des ressources, des contraintes temporelles, des préférences des utilisateurs, etc.

Ce problème est couramment rencontré dans de nombreux domaines tels que la production industrielle, la gestion de projets, la logistique, l'ordonnancement des transports, l'emploi du temps des employés, la planification des opérations chirurgicales, et bien d'autres. Les méthodes pour résoudre les problèmes d'ordonnancement varient en fonction de la complexité du problème et des objectifs à optimiser, et peuvent faire appel à des techniques d'optimisation combinatoire, de programmation linéaire, d'heuristiques ou de métaheuristiques. [6]

### 1.7 Domaines d'application d'optimisation combinatoire :

Les domaines d'application de l'optimisation combinatoire sont vastes et variés. Voici quelques-uns des domaines principaux où l'optimisation combinatoire trouve des applications importantes :

- **Logistique et Transport** : L'optimisation combinatoire est utilisée pour résoudre des problèmes de routage de véhicules, de planification des itinéraires, d'optimisation des horaires de transport en commun, etc.
- **Fabrication et Production** : Dans le domaine de la fabrication, l'optimisation combinatoire est utilisée pour optimiser les processus de production, la planification de la chaîne d'approvisionnement, la gestion des stocks, etc.,
- **Télécommunications et Réseaux** : Elle est utilisée pour la conception et l'optimisation des réseaux de télécommunications, la planification des réseaux de communication, la conception d'algorithmes de routage efficaces, etc.

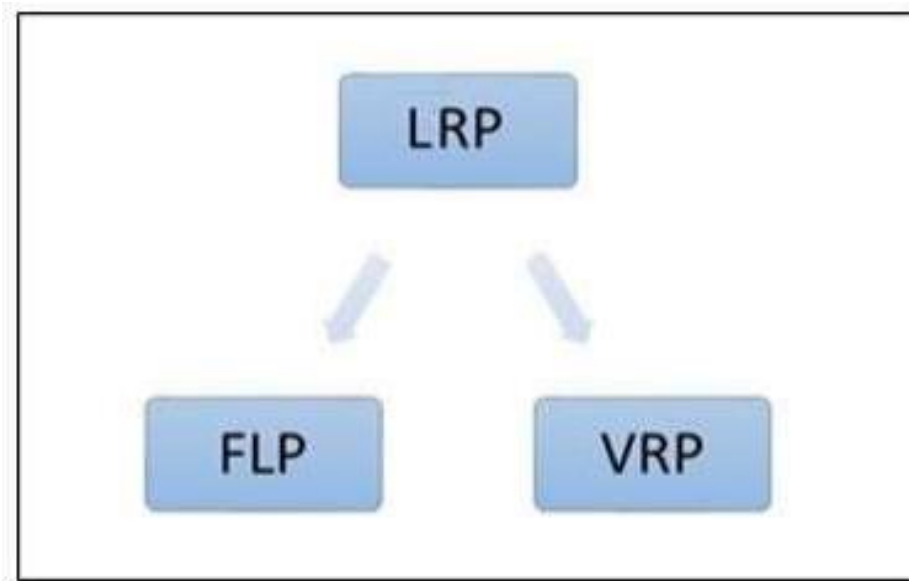
- Informatique et Algorithmes : L'optimisation combinatoire est au cœur de nombreux algorithmes utilisés dans le traitement de l'image, la recherche d'itinéraires, l'apprentissage automatique, l'optimisation de la logique des circuits, etc.
- Planification et Ordonnancement : Elle est largement utilisée dans la planification de projets, la planification de la production, l'ordonnancement des tâches, la planification des horaires, etc.
- Finance et Économie : Dans le domaine financier, l'optimisation combinatoire est utilisée pour l'optimisation de portefeuille, la planification financière, la gestion des risques, etc.
- Biologie et Génétique : Elle est utilisée pour résoudre des problèmes d'optimisation liés à la bio-informatique, à la recherche de séquences génétiques, à la modélisation de réseaux métaboliques, etc.
- Énergie et Environnement : Elle est utilisée pour l'optimisation de la distribution d'énergie, la planification énergétique, la conception de réseaux électriques, etc. Ces domaines ne représentent qu'une fraction des nombreuses applications de l'optimisation combinatoire, qui continue d'avoir un impact significatif dans de nombreux aspects de la vie moderne. [3]

### 1.8 Problème de localisation du routage est un problème d'optimisation :

#### 1.8.1 Présentation du LRP :

Le problème de routage de localisation (LRP) combine les aspects du problème de routage de véhicules (VRP) et du problème d'affectation d'emplacements (FRP). Dans le LRP, l'objectif est de déterminer les itinéraires optimaux pour un ensemble de véhicules afin de desservir un ensemble de clients situés à des emplacements spécifiques, tout en tenant compte de la décision de localisation des installations de service. En d'autres termes, le LRP consiste à déterminer les meilleures affectations de véhicules aux clients, ainsi que les emplacements optimaux des centres de distribution ou des entrepôts, tout en minimisant des objectifs tels que le coût total de transport, la distance parcourue, ou le temps nécessaire pour desservir tous les clients. Cette combinaison de VRP et de FRP introduit des défis supplémentaires en termes de complexité et de dimensionnalité, ce qui en fait un sujet de recherche captivant pour explorer les implications opérationnelles, logistiques et d'optimisation, ainsi que pour

développer des stratégies novatrices pour la gestion efficace des réseaux de distribution et de transport. [8]



**Figure 1.5** : LRP et sa relation avec FLP et VRP [9]

### 1.8.2 LRP est un problème d'optimisation :

Le problème de la localisation et du routage (LRP), est bien un problème d'optimisation. Il vise à trouver la meilleure combinaison possible d'emplacements pour les installations (comme les entrepôts, les centres de distribution) et les itinéraires de distribution pour minimiser ou maximiser certains objectifs prédéfinis.

Les objectifs peuvent inclure la minimisation des coûts de transport, des temps de parcours, des émissions de gaz à effet de serre, ou encore la maximisation de la satisfaction des clients. Pour résoudre ce problème, différentes techniques d'optimisation sont utilisées pour explorer efficacement l'espace des solutions et trouver la combinaison optimale d'emplacements et d'itinéraires qui satisfait aux contraintes et optimise les objectifs définis.

### 1.8.3 Application réelle du Problème de Routage de Localisation :

Le modèle LRP a été utilisé dans différentes variantes de problèmes réels. Dans cette section, nous présentons quelques applications du LRP dans les domaines suivants : la santé, les catastrophes naturelles, le militaire, la distribution alimentaire, le secteur des communications, le secteur des carburants, la collecte et le recyclage des déchets, le secteur de la distribution, le secteur financier et les sciences spatiales.

- **Santé** : En santé, le LRP peut être utilisé pour optimiser le déploiement des ambulances ou des véhicules de transport médical afin de garantir une réponse rapide aux urgences et une couverture efficace des zones géographiques. Les emplacements

optimaux pour les centres de santé et les hôpitaux peuvent également être déterminés en utilisant des variantes du LRP pour maximiser l'accessibilité des services de santé.

Catastrophes

- **Les catastrophes Naturelles :** En cas de catastrophe naturelle, le LRP peut être utilisé pour planifier les itinéraires de secours, déterminer les emplacements optimaux pour les centres de secours et organiser le déploiement des ressources de manière à minimiser les temps de réponse et maximiser la couverture des zones touchées.

- **Militaire :** Dans le domaine militaire, le LRP peut être utilisé pour planifier les mouvements des unités, la logistique de ravitaillement et la distribution des ressources dans les zones de déploiement. Il peut également être utilisé pour optimiser le positionnement des bases militaires et des centres de commandement.

- **Distribution Alimentaire :** Pour la distribution alimentaire, le LRP peut être utilisé pour planifier les itinéraires des camions de livraison, déterminer les emplacements des entrepôts et des points de vente, et minimiser les coûts de transport tout en maximisant la fraîcheur des produits.

- **Secteur des Communications :** Dans le secteur des communications, le LRP peut être utilisé pour optimiser le déploiement des réseaux de télécommunication, déterminer les emplacements des tours de transmission et planifier les itinéraires de maintenance et de réparation.

- **Secteur des Carburants :** Le LRP peut être utilisé dans le secteur des carburants pour planifier les itinéraires des camions-citernes de livraison, déterminer les emplacements optimaux pour les stations-service et minimiser les coûts de transport.

- **Collecte et Recyclage des Déchets :** Pour la collecte et le recyclage des déchets, le LRP peut être utilisé pour optimiser les itinéraires des camions de collecte, déterminer les emplacements des centres de tri et maximiser l'efficacité de la gestion des déchets.

- **Secteur de la Distribution :** Dans le secteur de la distribution, le LRP peut être utilisé pour optimiser les itinéraires de livraison, déterminer les emplacements des entrepôts et des centres de distribution, et minimiser les coûts logistiques tout en maximisant la satisfaction client.

- **Secteur Financier :** Dans le secteur financier, le LRP peut être utilisé pour optimiser le réseau d'agences bancaires, déterminer les emplacements optimaux pour les guichets automatiques, et maximiser la couverture des services bancaires. [9]

### 1.8.4 Historique du LRP

Le sujet général de la Planification de Routage de Lieux (LRP) a été largement étudié au cours des dernières décennies. Maranzana (1964) est considéré comme l'un des premiers auteurs à avoir combiné les décisions de localisation avec les coûts de transport. Plusieurs articles influents ont été publiés dans les années suivantes, abordant divers aspects de la LRP. Par exemple, Jacobsen et Madsen (1980) ainsi que Madsen (1983) ont évalué trois heuristiques pour la distribution de journaux. Perl et Daskin (1985) ont proposé un programme mixte entier pour résoudre une LRP d'entrepôt, appliqué à un système de distribution réel. Laporte et al. (1986) ont présenté un programme linéaire entier pour résoudre une LRP capacité, suivi par Laporte et al. (1988) qui ont étudié une LRP contrainte par coûts. Laporte et al. (1989) ont abordé une LRP stochastique pour la première fois.

Aykin (1995) a traité un problème de localisation de hub avec un programme entier et une heuristique itérative. Des approches de résolution hybrides ont également été proposées, telles que celle de Prins et al. (2006) qui ont combiné GRASP avec un processus d'apprentissage. La LRP a également été appliquée à des domaines spécifiques tels que la gestion des déchets dangereux (Alumur et Kara, 2007 ; Samanlioglu, 2013), la logistique urbaine (Nataraj et al., 2019), et la logistique humanitaire (Ukkusuri et Yushimito, 2008). Certaines études ont considéré des paramètres stochastiques, comme Quintero-Araujo et al. (2019b) et Rabbani et al. (2019) qui ont proposé des algorithmes simheuristiques pour traiter l'incertitude. Des revues de littérature ont également été réalisées, couvrant différents aspects de la LRP, y compris les formulations mathématiques, les approches de résolution et les variantes non standards. Cependant, la prise en compte explicite des décisions de dimensionnement des installations a été rarement abordée. Hemmelmayr et al. (2017), Tunalioglu et al. (2016), et Zhou et al. (2019) sont parmi les rares auteurs à avoir considéré cette dimension. Des approches heuristiques et métaheuristiques ont été proposées pour résoudre ces problèmes, et des études de cas ont été menées pour évaluer les performances des méthodes proposées. En conclusion, la LRP reste un domaine de recherche actif, avec des développements continus dans la modélisation, les méthodes de résolution et les applications spécifiques. [10]

### 1.8.5 L'objectif de LRP :

- Minimisation du nombre de véhicules.
- Minimisation du coût total de la tournée, y compris l'équipement, l'entretien des véhicules et le salaire des chauffeurs, les véhicules et les dépenses variables

- Maximisation du gain généré par la tournée en cas de collecte de produits auprès des clients. Maximisation de la qualité de service.
- Maximisation du chargement des véhicules utilisés pour les tournées. [11]

### 1.9 Méthodes de résolution d'un problème d'optimisation :

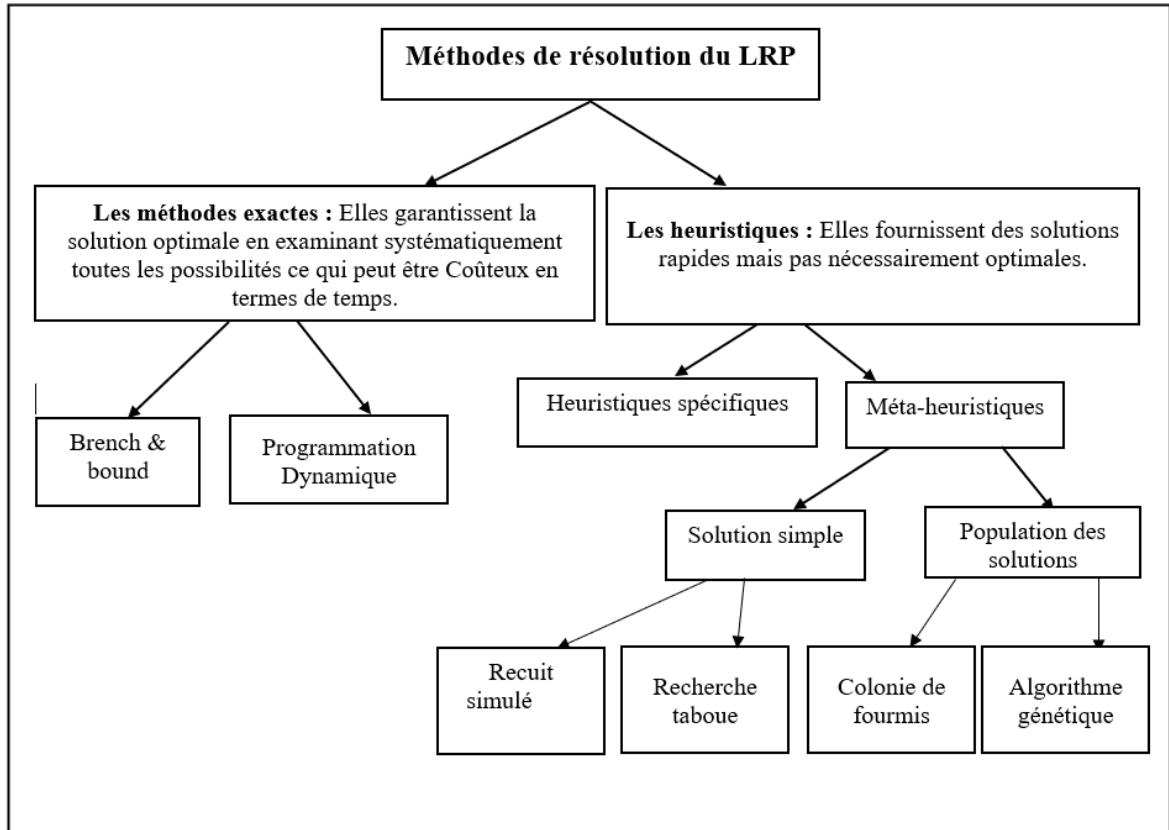


Figure1.6 : Les différentes méthodes de résolution en optimisation combinatoire. [3]

#### 1.9.1 Méthodes Exactes

Les méthodes exactes sont généralement préconisées pour résoudre les problèmes de petites tailles pour lesquelles le nombre de combinaisons possibles est suffisamment faible pour pouvoir explorer l'espace de solutions dans un temps raisonnable. Elles permettent de trouver une solution optimale à un problème donné. Toutefois, ces méthodes peuvent devenir rapidement coûteuses en temps d'exécution, notamment pour les problèmes NP- difficiles. En effet, le temps de traitement et la complexité du problème sont généralement liés (plus c'est complexe, plus le temps d'exécution sera important). Ci-dessous, quelques méthodes exactes parmi les plus connues :

##### 1.9.1.1 Procédure par Séparation et Évaluation :

La méthode de Branch and Bound (B&B), également connue sous le nom de "séparation et évaluation", est une approche algorithmique utilisée pour résoudre des problèmes

d'optimisation combinatoire visant à trouver une solution optimale. Cette méthode repose sur la division du problème initial en sous-problèmes, représentés sous forme d'un arbre d'états. Trois axes principaux structurent cette méthode :

### ✚ L'évaluation :

Cette étape consiste à réduire l'espace de recherche en comparant le coût de chaque nœud visité avec le coût de la meilleure solution connue jusqu'à présent. Si le coût du nœud en cours est supérieur à la meilleure solution actuelle, l'exploration de cette branche est arrêtée, car toutes les solutions de cette branche auront un coût plus élevé que la meilleure solution déjà trouvée.

### ✚ La séparation :

Il s'agit de diviser le problème initial en sous-problèmes. En résolvant ces sous-problèmes et en conservant la meilleure solution trouvée, on peut progressivement résoudre le problème initial. Cela se traduit par la construction d'un arbre qui énumère toutes les solutions possibles. Les nœuds de l'arbre qui n'ont pas encore été explorés et qui pourraient contenir une solution optimale sont appelés "nœuds actifs".

### ✚ La stratégie de parcours :

Trois stratégies principales sont utilisées pour explorer l'arbre d'états : la largeur d'abord, la profondeur d'abord et le meilleur d'abord. Chacune de ces stratégies présente des avantages et des inconvénients en termes d'efficacité et de gestion de la mémoire, et peut être choisie en fonction des caractéristiques spécifiques du problème à résoudre. En utilisant ces trois axes de manière combinée, la méthode de Branch and Bound permet de systématiquement explorer l'espace de recherche de manière efficace, en éliminant rapidement les branches qui ne contiennent pas de solutions optimales et en convergeant vers la solution optimale du problème. [7]

#### 1.9.1.2 Programmation dynamique :

La programmation dynamique est une méthode employée pour résoudre des problèmes où il est nécessaire de trouver une séquence optimale de décisions. Le principe fondamental repose sur la capacité à déterminer la solution optimale d'un problème global en combinant les solutions optimales de sous-problèmes plus petits, chacun correspondant à des séquences de décisions plus courtes. Les solutions aux différents sous-problèmes sont calculées de manière progressive et ascendante : on commence par résoudre les plus petits sous-problèmes, puis on utilise ces solutions pour construire progressivement la solution du problème global. [12]

### 1.9.1.3 La programmation linéaire

La programmation linéaire (PL) est une branche de l'optimisation utilisée pour résoudre divers problèmes économiques et industriels. Elle consiste à résoudre des problèmes où la fonction objectif et les contraintes sont toutes linéaires. Lorsqu'un ensemble de solutions possibles,  $S$ , est formulé comme un ensemble de variables prenant des valeurs réelles, et que les contraintes sont exprimées sous forme d'inégalités linéaires, avec une fonction objectif linéaire, on parle alors de problème de programmation linéaire (PL). De nombreux problèmes pratiques en recherche opérationnelle peuvent être modélisés en tant que problèmes de PL. En conséquence, de nombreux algorithmes conçus pour résoudre d'autres types de problèmes d'optimisation sont basés sur des techniques de résolution de problèmes linéaires. [12]

### 1.9.2 Méthodes Approchées

Les méthodes approchées permettent de trouver de manière rapide une solution réalisable à un problème donné. Cependant cette solution n'est pas forcément la solution optimale. Elle est divisée en deux catégories :

#### 1.9.3 Méthodes Heuristiques :

Une heuristique est une méthode de calcul qui fournit rapidement une solution réalisable, pas nécessairement optimale ou exacte, pour un problème d'optimisation difficile. C'est un concept utilisé entre autres en optimisation combinatoire, en théorie des graphes, en théorie de la complexité des algorithmes et en intelligence artificielle.

Une heuristique s'impose quand les algorithmes de résolution exacte sont de complexité exponentielle, et dans beaucoup de problèmes difficiles. L'usage d'une heuristique est aussi pertinent pour calculer une solution approchée d'un problème ou pour accélérer le processus de résolution exacte. Généralement, une heuristique est conçue pour un problème particulier, en s'appuyant sur sa structure propre, mais les approches peuvent contenir des principes plus généraux.

Elles peuvent être classées en deux catégories :

1.9.3.1 Méthodes constructives : qui génèrent des solutions à partir d'une solution initiale, en essayant d'ajouter des éléments petits à petit jusqu'à ce qu'une solution complète soit obtenue.

1.9.3.2 Méthodes de recherche locale (amélioration) : qui démarrent d'une solution initialement complète (Probablement moins intéressante), et de manière répétitive essaie d'améliorer cette solution en explorant son voisinage. [3]

1.9.4 Méthode exacte vs Méthode heuristique : [1]

	<i>Exact</i>	<i>Heuristiques</i>
Caractéristique	Optimalité	Approximation
Garantie de convergence	Converge vers une solution optimale	Peut converger vers une solution satisfaisante mais pas optimale
Complexité	$O(2^n)$ (souvent)	$O(n^k)$
Applicabilité	Convient aux petites tailles et problèmes bien définis	Adaptées aux grandes tailles NP-difficiles
Preuve	Formelle	Empiriquement

**Tableau1.2** : Méthode exacte vs méthode heuristique.

### Conclusion

L'objectif de ce chapitre est de présenter le contexte et le problème abordé dans notre thèse. Nous avons donné un aperçu global de l'optimisation globale, y compris les méthodes de résolution associées. Nous avons commencé par expliquer les méthodes exactes qui garantissent des solutions précises mais à un coût temporel élevé pour les problèmes de grande taille. Nous avons également discuté des méthodes approximatives qui sacrifient la précision de la solution en échange d'un temps de réponse plus court.



***Chapitre 2 : Etat de L'art***

### 2.1. Introduction

Ce chapitre sera consacré à l'examen de certaines catégories différentes du problème de localisation et de routage (LRP) ainsi que des problèmes connexes tels que le problème de routage des véhicules (VRP), qui ont été traités dans des travaux antérieurs avec une brève description.

### 2.2. Revue de littérature

#### **Mémoire de master intitulé : Le problème de routage de véhicule avec capacité par l'algorithme de colonies de fourmis CVRP :**

Cette thèse a abordé le problème du voyageur de commerce, qui est l'un des problèmes les plus difficiles dans le domaine de l'optimisation combinatoire. Ce problème vise à déterminer les itinéraires optimaux pour un ensemble de véhicules afin de servir un ensemble spécifique de clients. Des algorithmes de colonies de fourmis (ACO) ont été utilisés, qui sont l'une des métaheuristiques spéciales utilisées pour résoudre les problèmes difficiles qui ne peuvent pas être résolus de manière exacte. Cette étude se concentre sur la résolution numérique du problème de distribution de véhicules avec contrainte de capacité (CVRP). [13]

#### ➤ L'algorithme de colonies de fourmis :

L'algorithme des colonies de fourmis (ACO) est une méthode d'optimisation inspirée du comportement des vraies fourmis. Il est utilisé pour résoudre divers problèmes d'optimisation combinatoire, tels que le problème du voyageur de commerce (TSP) et le problème de routage de véhicules (VRP).

L'idée de base derrière l'ACO est de simuler le comportement des fourmis lorsqu'elles recherchent des chemins optimaux entre leur nid et une source de nourriture. Les fourmis communiquent entre elles en déposant des phéromones sur le sol. Ces phéromones servent de piste chimique, permettant aux autres fourmis de suivre le chemin déjà parcouru par leurs congénères.

L'algorithme ACO commence généralement par une phase d'initialisation où les fourmis se dispersent aléatoirement sur le graphe du problème. Ensuite, elles construisent progressivement des solutions en suivant des règles basées sur les phéromones et les caractéristiques du problème. Les solutions sont évaluées en fonction de leur qualité, et les phéromones sont mises à jour en conséquence.

Au fil des itérations, les chemins contenant de bonnes solutions ont tendance à être renforcés par des phéromones, tandis que les chemins conduisant à des solutions médiocres sont affaiblis. Cela conduit à une convergence vers des solutions de meilleure qualité au fil du temps. En résumé, l'algorithme des colonies de fourmis est une technique d'optimisation probabiliste qui imite le comportement des fourmis pour trouver des solutions efficaces à des problèmes d'optimisation difficile [10]

### **Mémoire de master intitulé : (Supply Chain) Problème de Tournée du Véhicules avec Fenêtre de Temps (PTVFT) :**

Cette mémoire aborde le problème du véhicule avec fenêtre de temps (PTVFT), qu'est le problème le plus étudié. Il pose la question suivante : "Quelle est la séquence optimale de routes pour livrer un ensemble donné de clients tout en respectant les délais de livraison et en retournant au dépôt une fois la tournée terminée ?". L'objectif de ce travail est de minimiser à la fois le coût de la distance parcourue et le temps. Pour ce faire, les chercheurs ont utilisé les algorithmes génétiques comme heuristique et ont programmé en Python. [14]

#### ➤ L'algorithme génétique :

Les algorithmes génétiques sont des méthodes inspirées par la nature, introduites par Holland en 1975 dans le cadre d'une analogie avec la sélection naturelle des espèces. Ils ont été largement développés et utilisés par Goldberg en 1989 pour résoudre des problèmes d'optimisation. Les algorithmes génétiques ont été créés pour imiter les processus d'évolution naturelle. À partir des années 1990, la programmation génétique (Genetic Programming).

(GP) a été développée par Koza dans le but de réaliser le rêve des programmeurs : écrire automatiquement des programmes. Ces méthodes reposent sur les processus de reproduction et de sélection génétiques. À partir d'un ensemble de solutions faisables (population) de taille  $n$  (taille de la population), chaque solution se voit attribuer une certaine probabilité d'être sélectionnée comme parent, cette probabilité dépendant du coût de la solution. Ensuite, ces solutions sont croisées pour obtenir un ou plusieurs enfants par paire de solutions.

Les parents sélectionnés pour le croisement le sont en fonction des probabilités calculées précédemment. Ensuite,  $n$  enfants sont sélectionnés en fonction du coût de chaque solution, et le processus est répété. À la manière d'un phénomène nature, la méthode repose sur l'idée qu'à mesure que les générations se renouvellent, la valeur moyenne des solutions faisables s'améliore. Les algorithmes génétiques nécessitent beaucoup de temps de calcul, car ils manipulent plusieurs solutions simultanément, mais leur grand avantage est qu'ils parviennent à trouver de bonnes solutions pour des problèmes très complexes. [15]

### **Thèse de doctorat intitulée : (Résolution d'un problème stochastique de localisation allocation avec gestion des indisponibilités) :**

Dans un contexte de forte concurrence et d'évolution économique, les industriels se trouvent contraints de renforcer leur compétitivité et d'explorer de nouvelles stratégies consistant à localiser des sites permettant de se rapprocher des marchés et à construire des réseaux commerciaux pour maintenir une trajectoire de développement continu. Cette thèse aborde le problème de la localisation aléatoire des sites avec gestion des indisponibilités des sites, dans le but de proposer une approche permettant une gestion efficace de ces situations. Deux problématiques liées à la théorie de la localisation sont examinées : la première concerne la gestion des indisponibilités par une réaffectation efficace des clients aux sites opérationnels en utilisant un algorithme génétique, ce qui réduit les coûts associés à la gestion des indisponibilités. La deuxième consiste à utiliser des sites mobiles dans un environnement dynamique où les sites peuvent être fermés et rouverts plusieurs fois pendant la période de planification. Une approche multi-objectifs est proposée pour réduire les coûts logistiques et les émissions de CO<sub>2</sub>. Les résultats obtenus pour ces deux problématiques sont présentés et analysés afin de démontrer l'efficacité de l'approche proposée. [16]

### **Article intitulé : (The Multi-depot Periodic Vehicle Routing Problem) :**

Cet article aborde le problème du MDVRP (Problème de Routage de Véhicules avec Multiples Dépôts), où plusieurs entrepôts sont distribués géographiquement, et chaque véhicule commence et termine sa tournée à son dépôt d'origine. Cela signifie que chaque véhicule part d'un dépôt spécifique, effectue sa tournée, puis retourne au même dépôt d'où il est parti.

Deuxièmement, il mentionne que dans certains cas, il peut y avoir plusieurs entrepôts où les véhicules sont distribués, mais les tournées ne doivent pas seulement commencer et se terminer au même entrepôt. En effet, chaque client doit également être visité une seule fois par l'un des véhicules appartenant aux entrepôts spécifiés par ce client.

En général, cela illustre la structure fondamentale des problèmes impliquant plusieurs entrepôts, où les objectifs incluent l'acheminement efficace des véhicules pour visiter les différents clients, tout en retournant aux entrepôts d'origine. [17]

### **Article intitulé : (Metaheuristics for the vehicle routing problem and extensions: A categorized bibliography):**

Cet article traite du problème de distribution de gaz de pétrole liquéfié (GPL) en Turquie, où la société ABC, l'un des principaux distributeurs de GPL dans le pays, distribue du gaz aux

stations-service d'Istanbul. L'objectif de l'étude est de trouver les meilleures méthodes pour répondre aux besoins quotidiens de distribution pour 13 stations à Istanbul.

Un modèle a été développé pour améliorer le processus de distribution en utilisant des informations telles que les demandes des clients, les emplacements des stations, les types de véhicules et leurs capacités. Le modèle utilise des réservoirs pour distribuer le GPL aux stations, avec différents réservoirs ayant des capacités différentes. La problématique consiste à déterminer les meilleures routes de distribution entre les stations de manière à réduire les coûts et à maintenir l'efficacité de la distribution.

Les données utilisées comprennent les coordonnées des stations et les distances entre elles, calculées à l'aide de ressources disponibles en ligne. L'objectif principal est de développer un modèle qui réduit les coûts et contribue à une distribution plus efficace du GPL. [18]

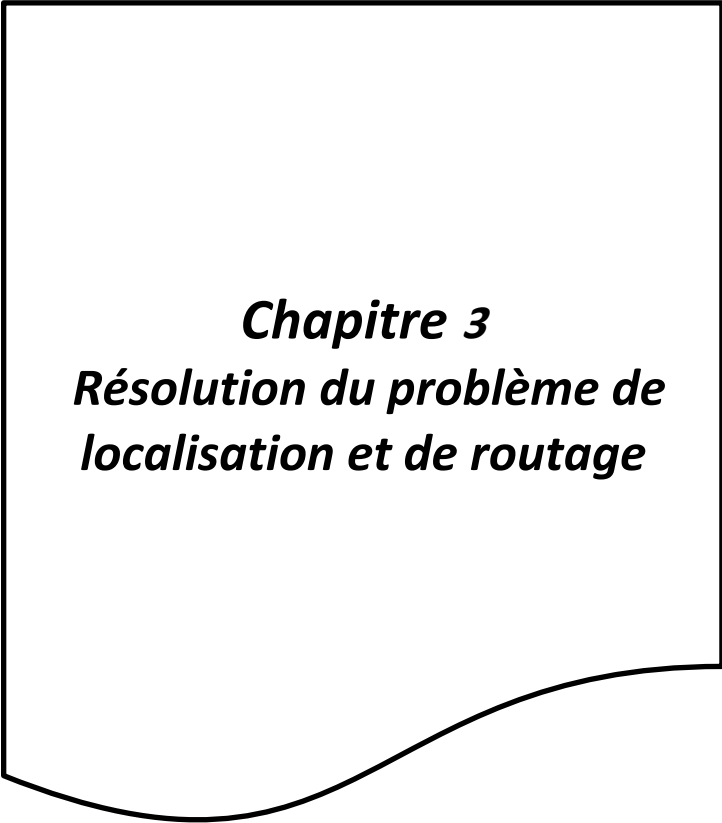
**Article intitulé : (Le problème de l'emplacement et du routage : une approche innovante.) :**

Le document scientifique "Le Problème de Routage de Localisation : Une Approche Innovante" aborde le problème du routage de localisation (CLRP) en le divisant en deux étapes principales : l'allocation de localisation (LAP) et le routage (VRP). Cette approche novatrice vise à améliorer l'efficacité de la distribution des ressources et à réduire les coûts associés à la livraison de services ou de marchandises aux clients. Les points clés étudiés dans l'article sont l'allocation de localisation (LAP) et le routage (VRP), et ils ont utilisé l'algorithme TSP-VRP comme méthode de résolution, fournissant des solutions plus efficaces par rapport aux méthodes traditionnelles pour résoudre le VRP.

Les résultats expérimentaux ont montré que le modèle proposé réduit les coûts d'exploitation et le nombre de chemins nécessaires.

### **2.3. Conclusion**

Dans ce chapitre, nous avons tenté d'aborder certaines des dérivations les plus courantes du problème LRP, ainsi que les méthodes de résolution utilisées. Les scientifiques et les chercheurs continuent encore aujourd'hui à découvrir de nombreuses dérivations de ce problème, telles que le temps, la capacité des véhicules et la dynamique, et ils cherchent également à développer et améliorer les solutions. Dans le prochain chapitre, nous présenterons l'approche adoptée dans ce travail pour résoudre ce problème.



***Chapitre 3***  
***Résolution du problème de***  
***localisation et de routage***

### Introduction

La résolution des problèmes de routage des véhicules et d'emplacement des installations (LRP) représente un défi majeur en raison de sa complexité et de l'augmentation des solutions possibles avec la taille du problème. Malgré l'existence de nombreuses méthodes pour résoudre ce type de problèmes, les méthodes exactes sont rares. Par conséquent, les chercheurs se concentrent principalement sur le développement de méthodes heuristiques pour obtenir des solutions de qualité en peu de temps.

Dans ce chapitre, nous étudierons la méthode du recuit simulé, une approche pour résoudre des problèmes d'optimisation qui nécessitent la considération de grands ensembles de solutions possibles. Dans les applications technologiques, ces problèmes peuvent rapidement devenir complexes, nécessitant des calculs très lourds en raison de l'explosion combinatoire du nombre de solutions potentielles à examiner. Pour mieux comprendre cela, nous utiliserons un exemple simple tel que le problème d'affectation, qui nous aidera à saisir les défis rencontrés dans la résolution de ces problèmes.

Supposons que nous ayons un problème de répartition des tâches parmi un groupe de travailleurs. Nous avons un certain nombre de tâches et un certain nombre de travailleurs, et chaque tâche doit être attribuée à un seul travailleur. L'objectif est de déterminer la meilleure manière de répartir ces tâches afin d'atteindre une efficacité maximale, que ce soit en termes de temps nécessaire pour accomplir les tâches ou en termes de répartition équitable du travail entre les travailleurs.

Par exemple, supposons que nous ayons 5 tâches et 3 travailleurs. Comment pouvons-nous répartir ces tâches entre les travailleurs de manière à les accomplir le plus rapidement possible et avec le moins de coûts possible ? Cet exemple simple reflète les défis que nous pouvons rencontrer dans l'allocation des ressources et la répartition des tâches, et comment la méthode du recuit simulé peut nous aider à trouver des solutions efficaces à de telles problématiques.

### 2.1 L'énoncé du problème

Dans le problème de routage de localisation (LRP), qui est une extension du problème de routage de véhicules capacitaires (CVRP), une flotte de véhicules de livraison de capacité uniforme doit desservir les clients ayant une demande connue pour un seul produit. Contrairement au CVRP, qui traite le problème LRP avec un seul emplacement de dépôt, différents emplacements de dépôt sont disponibles, chacun ayant son propre coût d'ouverture et sa propre capacité. Les véhicules terminent leur parcours à leur dépôt de départ. Chaque client ne peut être servi que par un seul véhicule. L'objectif est de déterminer quels dépôts ouvrir

et quels itinéraires emprunter tout en minimisant le coût total, de telle sorte que tous les clients soient servis et que la demande totale desservie par chaque dépôt et véhicule ne dépasse pas sa capacité.

### 3.1.1. Données :

Le format des fichiers de données est le suivant :

- Le nombre de clients

- Le nombre de dépôts

- Les coordonnées x et y des dépôts et des clients

- La capacité des véhicules de livraison

- La capacité de chaque dépôt

- La demande pour chaque client

- Le coût d'ouverture de chaque dépôt

- Le coût d'ouverture d'un itinéraire

- La valeur booléenne `areCostDouble` indiquant si les coûts doivent être arrondis.

### 3.1.2. Traitement

Le programme définit une séquence pour les véhicules `nbTrucks` sous forme de variable de liste (`customersSequences[r]`). Il correspond à la séquence de clients visités. Pour garantir que tous les clients doivent être servis, toutes les variables de la liste sont contraintes de former une partition en utilisant l'opérateur « partition ». Le modèle définit également un groupe de séquences associées à chaque dépôt en tant que variable définie (`dépôts[d]`). Les dépôts sont également contraints de former une partition, pour garantir que chaque séquence soit associée à un dépôt.

Le nombre de séquences `nbTrucks` est déterminé comme le nombre minimum de véhicules nécessaire pour servir chaque client sans surcharger les véhicules, multiplié par un facteur 1,5 pour garantir la faisabilité. La quantité livrée à un client doit répondre à la demande correspondante. La quantité servie dans une séquence `quantityServed[r]` est définie avec une somme et l'accès au tableau des demandes avec l'opérateur « at » et ne doit pas dépasser sa capacité.

Pour déterminer le dépôt affecté à un itinéraire, on utilise l'opérateur « find ». Un dépôt est considéré comme ouvert si au moins une séquence (non vide) l'utilise. Cette condition est calculée avec l'opérateur « count ». Les `deposCost` sont définis comme la somme des coûts d'ouverture des dépôts ouverts. De même, une séquence est considérée comme ouverte si au moins un client est servi pour cette séquence. Nous calculons la distance totale pour chaque séquence en additionnant la distance entre le dépôt et le premier client, les distances entre

chaque client et la distance retour au dépôt depuis le dernier client. La somme de  $sequenceLength$  et  $openingCostRoute$  correspond au  $routerCost$ .

### 3.1.3. L'objectif

L'objectif est de minimiser la somme de  $depotsCost$  et  $routerCost$

## 2.2 La problématique

La problématique de problème de localisation du routage, concerne l'intégration des aspects de routage des véhicules et de localisation dans le processus de planification des sites. Ce problème est crucial dans de nombreux domaines où des biens doivent être transportés efficacement d'un lieu à un autre. Dans le contexte du transport de marchandises ou de produits, le LRP vise à déterminer les meilleurs emplacements pour les dépôts ou les centres de distribution tout en optimisant les itinéraires des véhicules qui desservent ces sites. Il s'agit essentiellement de minimiser les coûts totaux associés au transport, y compris les coûts de distance parcourue, les coûts de stockage et de manutention, ainsi que les coûts de service à la clientèle. Le LRP peut être complexe en raison de plusieurs facteurs, notamment la demande variable des clients, les capacités des véhicules, les contraintes de temps, les coûts opérationnels et les contraintes géographiques. Pour résoudre ce problème, différentes approches algorithmiques et heuristiques sont utilisées, telles que la programmation mathématique, les métaheuristiques (comme la recherche taboue, le recuit simulé, les algorithmes génétiques), ainsi que des méthodes de résolution exacte ou approximative. En fin de compte, résoudre efficacement le problème de localisation du routage permet aux entreprises de réduire leurs coûts opérationnels, d'améliorer la satisfaction des clients en fournissant des services plus rapides et plus fiables, et de maximiser l'utilisation des ressources disponibles ; Chaque tournée doit respecter les contraintes suivantes :

- ✓ Un client ne peut être servi que par un et un seul véhicule.
- ✓ Chaque véhicule effectue une seule tournée.
- ✓ Tous les clients doivent être desservis.
- ✓ Capacité et/ou temps maximums à respecter et/ou nombre de véhicules à satisfaire.[19]

## 2.3 Motivation

Il existe plusieurs pressions imposées lors de l'optimisation des systèmes logistiques, les plus importantes étant les facteurs économiques, sociaux et environnementaux.

- Sur le plan économique, il s'agit pour les entreprises d'utiliser efficacement leurs ressources, y compris les ressources matérielles, humaines et financières. Dans le

contexte de l'amélioration de la chaîne d'approvisionnement, il est essentiel d'exploiter au maximum les capacités des entrepôts, des fournisseurs et des moyens de transport afin de réduire les coûts logistiques globaux.

- Du point de vue social, les situations de catastrophe telles que les séismes, les tsunamis, etc., nécessitent la mise en place de systèmes logistiques humanitaires pour distribuer de l'eau, des médicaments et d'autres fournitures essentielles à la population en utilisant des ressources limitées. Cela demande de résoudre le problème de l'affectation des ressources logistiques pour déterminer où stocker ces fournitures et comment les distribuer. Tout échec à fournir ces fournitures de base dans de telles situations peut entraîner la perte de vies humaines ou causer des dommages permanents à la société.
- Enfin, du point de vue environnemental, il est largement reconnu que chaque produit que nous consommons a un impact environnemental. Les gouvernements et les clients exigent désormais des produits "plus respectueux de l'environnement". Cela peut être réalisé en réduisant les niveaux de stock, car cela nécessite moins d'énergie de stockage, par exemple pour les produits réfrigérés. De même, réduire les distances parcourues par les véhicules de distribution signifie moins d'émissions de dioxyde de carbone. En résumé, l'amélioration des opérations logistiques peut fournir des solutions qui contribuent à rendre la chaîne d'approvisionnement plus durable. [10]

### 2.4 La Méthode De Résolution (Recuit Simulé) :

#### 2.4.1 Recuit simulé :

La méthode d'optimisation du recuit simulé est une technique probabiliste utilisée pour résoudre des problèmes d'optimisation. Inspirée par le processus métallurgique de recuit, cette méthode simule le processus de chauffage et de refroidissement d'un matériau solide pour atteindre un état de faible énergie. En pratique, le recuit simulé explore l'espace de recherche en effectuant des transitions entre différents états selon une règle probabiliste. Initialement, le système est souvent démarré à une température élevée, ce qui permet une exploration plus aléatoire de l'espace de recherche. Au fil du temps, la température est réduite progressivement selon un schéma prédéfini, ce qui permet au processus de converger vers des états de plus en plus optimaux. La règle de transition utilisée dans le recuit simulé est généralement basée sur l'algorithme de Metropolis, qui spécifie que les transitions sont acceptées avec une certaine probabilité, même si elles conduisent à une augmentation de la fonction objectif. Cette probabilité de transition dépend de la différence de coût entre l'état actuel et l'état proposé, ainsi que de la température actuelle. Plus, la température est élevée, plus il est probable que des

transitions menant à une augmentation de la fonction objectif soient acceptées, ce qui permet d'éviter les minimums locaux.

En résumé, le recuit simulé est une méthode d'optimisation flexible et efficace, capable de trouver des solutions de qualité pour un large éventail de problèmes d'optimisation, y compris ceux avec des espaces de recherche complexes et des fonctions objectif non linéaires. [15]

### 2.4.2 Pourquoi la méthode de recuit simulé :

Après une étude approfondie de plusieurs méthodes proposées dans plusieurs articles traitant du problème LRP, nous avons constaté que la plupart des articles prouvaient l'efficacité de la méthode de récupération simple pour résoudre ce problème car :

#### ➤ Efficace dans la résolution de problèmes combinatoires complexes :

Le problème de localisation du routage implique souvent une combinaison de facteurs et de contraintes qui le rendent complexe. Le recuit simulé est connu pour être efficace dans la résolution de problèmes combinatoires complexes en explorant intelligemment l'espace des solutions possibles.

#### ➤ Capable d'éviter les minima locaux :

Contrairement à certaines méthodes d'optimisation qui peuvent rester coincées dans des solutions sous-optimales (minima locaux), le recuit simulé est conçu pour éviter cet écueil en acceptant parfois des solutions de qualité inférieure avec une certaine probabilité, ce qui permet d'explorer un plus grand espace de recherche

#### ➤ Flexible dans la gestion des contraintes :

Le recuit simulé peut être adapté pour prendre en compte diverses contraintes telles que la capacité des véhicules, les contraintes de temps, les exigences de service à la clientèle, etc. Cette flexibilité permet de modéliser efficacement les aspects complexes du problème de localisation du routage.

#### ➤ Moins sensible aux paramètres :

Contrairement à certaines méthodes d'optimisation qui nécessitent un réglage minutieux des paramètres, le recuit simulé est relativement moins sensible aux paramètres, ce qui le rend plus facile à mettre en œuvre et à utiliser. En combinant ces avantages, le recuit simulé s'est avéré être une méthode efficace pour résoudre le problème de localisation du routage, permettant ainsi d'obtenir des solutions de qualité tout en optimisant les ressources disponibles et en respectant les contraintes spécifiques du problème. C'est ce qui poussé à le proposer dans notre mémoire.

### 2.4.3 Principe De La Méthode Du Recuit Simulé :

La méthode du recuit simulé commence par une solution initiale acceptable et continue à explorer l'espace des solutions en apportant des modifications mineures à la solution actuelle. Si la nouvelle solution obtenue est améliorée par rapport à la précédente, elle est conservée. Si elle est détériorée par rapport au critère d'optimisation, elle est conservée avec une probabilité inversement proportionnelle au nombre d'itérations. Le recuit simulé a l'avantage de couvrir un espace de recherche plus large et d'éviter la convergence prématurée vers un optimum local. Plusieurs problèmes de job shop ont été traités avec succès par la méthode du recuit simulé. Voici un exemple pour illustrer le fonctionnement du recuit simulé :

Supposons que nous gérons une boutique de confiseries et que nous voulons distribuer  $N$  types différents de bonbons sur  $N$  étagères à l'intérieur du magasin. Chaque type a ses propres avantages et inconvénients, et il est nécessaire de les distribuer de manière à ce que l'arrangement final soit rentable et attrayant pour les clients. Le processus commence par une distribution aléatoire des types de bonbons sur les étagères à l'intérieur du magasin comme une première répartition. Ensuite, la qualité de cette distribution initiale est évaluée en calculant la performance économique du magasin, telle que les revenus attendus ou le coût total. Ensuite, la distribution actuelle est ajustée de manière mineure pour obtenir une distribution voisine. Cela peut être fait en déplaçant un type de bonbon d'une étagère à une autre, par exemple. Ensuite, la performance économique de la nouvelle distribution est calculée, et si la nouvelle distribution est meilleure que l'ancienne, elle est acceptée. Sinon, une décision est prise pour l'accepter en fonction d'une certaine probabilité qui diminue avec le temps, aidant à éviter les optima locaux. En résumé, la méthode du recuit simulé explore l'espace des solutions de manière probabiliste, ce qui permet de trouver des solutions satisfaisantes sans avoir besoin d'examiner toutes les possibilités, économisant ainsi du temps et des efforts dans la recherche des solutions optimales. [3]

### 2.4.4 Formulation mathématique :

Le LRP est représenté par le graphe  $G = (V, A)$ ,

$V$  = L'ensemble des nœuds ;

$A$  = L'ensemble des arcs ;

$I$  = L'ensemble des clients ;

$M$  = L'ensemble des emplacements des dépôt potentiels ;

$N_i$  = La demande des clients ;

$Z_{ijk}$  = La variable binaire égale à 1 si l'itinéraire  $i_j$  est emprunté par le véhicule  $k$  dont la capacité est égale à  $q$  ;

$O_m$  = Le coût d'installation du dépôt sur les sites  $m$  ;

$R_m$  = La capacité du dépôt ;

$Y_m$  = Est égal à 1 si le dépôt  $m$  est ouvert ;

$C_{ijk}$  = Est la matrice des coûts d'exploitation entre les dépôts pour clients et clients à clients ;

$K_j$  = L'ensemble des véhicules situés au dépôt  $m$  ;

$K$  = L'ensemble de tous les véhicules ;

Étant donné un ensemble de clients et dépôts, le LRP consiste à trouver un ensemble de dépôts et un ensemble d'itinéraires d'un coût total minimal et répondre à la demande des clients. Le véhicule doit départ et retour au même dépôt et visite du client exactement une fois. La formule standard, La situation du LRP est présentée ci-dessous :

$$\min \sum_{m \in M} O_m Y_m + \sum_{i \in I} \sum_{j \in I} \sum_{k \in K} C_{ijk} Z_{ijk} \quad (1)$$

Sujet à :

$$\sum_{k \in K} \sum_{j \in V} Z_{ijk} = 1 \quad \forall i \in I \quad (2)$$

$$\sum_{j \in V} Z_{ljk} = 1 \quad \forall l \in M, \quad \forall k \in K \quad (3)$$

$$\sum_{i \in V} Z_{ilk} = 1 \quad \forall l \in M, \quad \forall k \in K \quad (4)$$

$$\sum_{i \in V} Z_{ihk} - \sum_{j \in V} Z_{hjk} = 0 \quad \forall h \in I, \quad \forall k \in K \quad (5)$$

$$\sum_{i \in I} N_i Z_{ik} \leq q \quad \forall k \in K \quad (6)$$

$$\sum_{k \in K} \sum_{i \in I} N_i V_{ik} \leq R_m Y_m \quad \forall m \in M, \quad k = \bigcup_{m \in M} K_m \quad (7)$$

$$S_{ik} + T_{ij} - S_{ik} \leq (1 - Z_{ijk}) M_{ijk} \quad \forall i, j \in A, \quad \forall k \in K \quad (8)$$

$$a_i \leq S_{ik} \leq b_i \quad \forall i \in V, \forall k \in K \quad (9)$$

$$Z_{ijk} \in \{0,1\} \quad \forall i, j \in A \quad (10)$$

$$Y_m \in \{0,1\} \quad \forall k \in K, \quad \forall m \in M \quad (11)$$

- Contrainte (2) indique que chaque client doit être visité exactement une fois.
- Contraintes (3) et (4) sont des contraintes de conservation de débit signifiant que si le véhicule quitte le dépôt  $l$ , il doit revenir au même dépôt.
- La contrainte (5) nécessite le véhicule pour laisser le client  $h$  après la visite- $il$ .

- Les contraintes (6) et (7) sont le véhicule et le dépôt contraintes de capacité, respectivement.
- Contrainte (8) est une contrainte de fenêtres temporelles impliquant que si  $V_e$  le véhicule passe de  $i$  à  $j$ , il doit servir le client  $i$  avant  $j$ .
- La contrainte (9) est le temps détendu gagné- dows  $[a_i, b_i]$ .

Les contraintes (10) et (11) renvoient aux variables de décision binaires si l'arc et le dépôt sont sélectionnés, respectivement. [20]

#### 2.5 Organigramme Général Du Recuit Simulé :

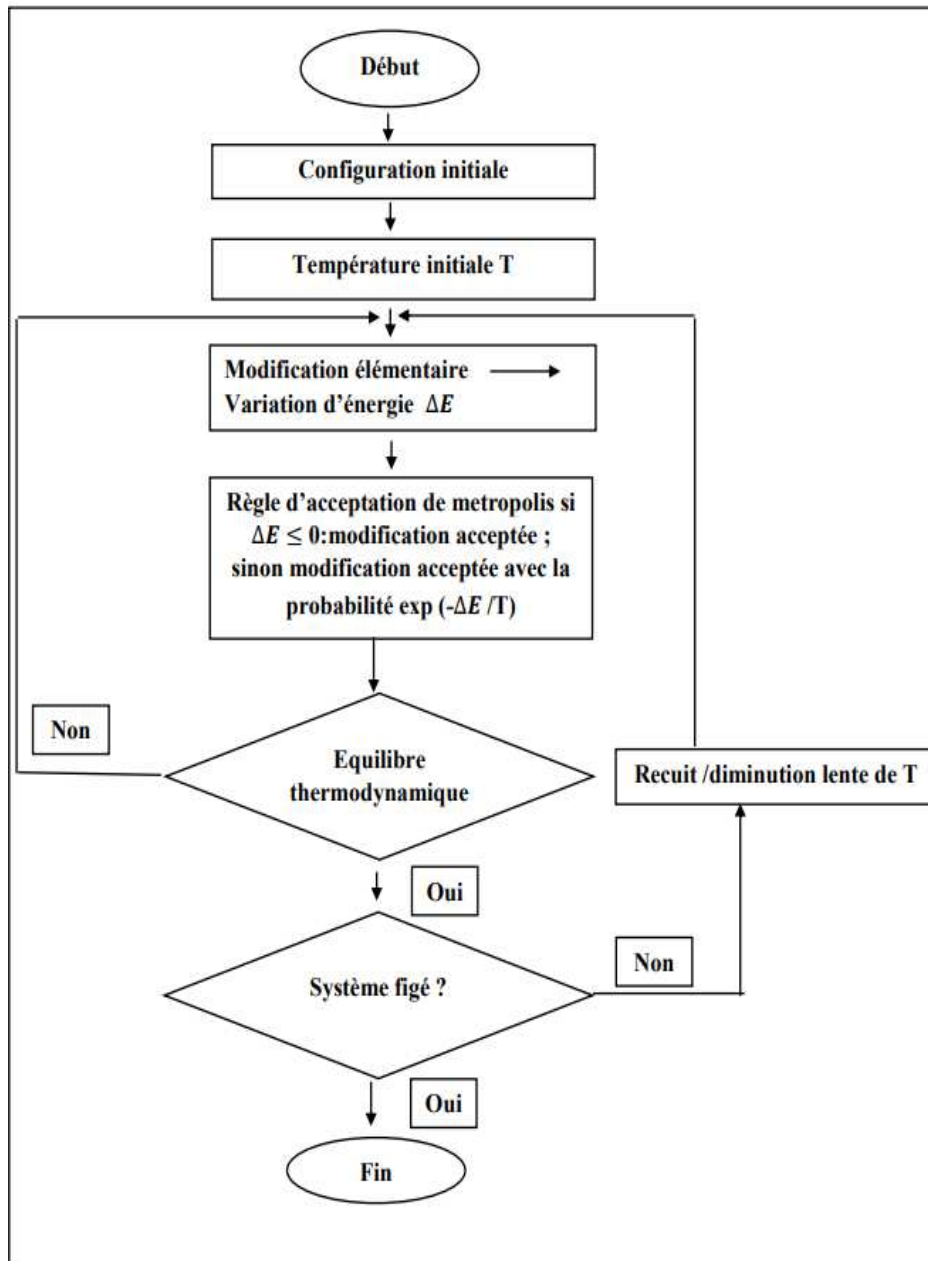


Figure3.1 : Organigramme général du recuit simulé [21]

**Algorithme du recuit simulé : [19]**

---

**Algorithme du recuit simulé**

---

**Début**

```
    /**Initialisation**/  
Var S,  $T_i > 0$ ,  $T_f > 0$ , N, K;  
    /**(S : solution initiale) (T : une température initiale et finale  $T_f$ ) (N : nombre  
    d'itérations) (K : coefficient de diminution de la température  $\emptyset \in [0,1]$   $T = T_i$ **/  
Tant que ( $T > T_f$ ) Faire  
    Pour ( $0 < k < N$ ) Faire  
        Sélectionner une solution voisine ( $S_i$ ) de la solution courante (S) ;  
        Calculer la variation de coût  $\Delta = f(S_i) - f(S)$  ;  
        /**f : la fonction objectif à minimiser**/  
        Si ( $\Delta < 0$ ) Alors  
             $S = S_i$   /**accepter la nouvelle solution ( $S_i$ )**/  
            Si ( $f(S_i) < f(S)$ ) Alors  
                 $S = S_i$   
            Sinon  
                 $S = S_i$  accepter la nouvelle solution avec la probabilité  $e^{-\Delta E / T}$   
                (Critère de Metropolis)  
                 $T = T * \emptyset$  (Schéma de refroidissement) /**Mettre à jour la température**/  
        Fin si  
    Fin pour  
Fin tant que  
    Affiche (S) ; /**Retourner la meilleure solution trouvée** /  
Fin
```

---

**2.6 Avantages et Inconvénients du Recuit Simulé**

La méthode du recuit simulé offre une implémentation facile avec une capacité généralement satisfaisante pour fournir des solutions de qualité par rapport aux algorithmes de recherche classiques. Elle peut être utilisée dans la plupart des problèmes d'optimisation et converge vers un optimum global avec un nombre croissant d'itérations. Cependant, son principal inconvénient réside dans sa propension à rester piégée dans un minimum local lorsque la température est basse, sans possibilité d'en sortir. Pour atténuer ce problème, plusieurs solutions ont été proposées, notamment en augmentant brusquement la température de temps en temps pour relancer la recherche sur d'autres régions plus éloignées. En outre, déterminer la température initiale optimale est difficile, ce qui peut affecter la qualité de la recherche ou prolonger le temps de calcul. De plus, il est impossible de garantir que la solution trouvée est

optimale, et ses performances peuvent diminuer dans les cas où il y a peu de minima locaux.  
[22]

#### **Conclusion**

Nous avons présenté dans ce chapitre les principaux concepts de l'algorithme recuit simulé et dans le prochain chapitre nous présentons les résultats de l'implémentation de notre approche sur des jeux de données standards.



***Chapitre 4 :***  
***Implémentation et résultats***

### Introduction

La méthode du recuit simulé est une méthode de résolution de problèmes qui repose sur la simulation du processus de refroidissement. Dans ce chapitre, nous examinerons la mise en œuvre de la méthode du recuit simulé et testerons l'efficacité de cette méthode pour résoudre le problème de localisation et routage (LRP). Pour ce faire, nous commencerons par décrire les outils utilisés pour le développement du projet, tels que le choix du langage de programmation et le matériel utilisé. Ensuite, nous présenterons un modèle ou un exemple des données sur lesquelles nous effectuerons les expériences, puis nous les comparerons et les discuterons.

### 3.2. Environnement de développement

#### 3.2.1. Environnement matérielle

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

- Un Processeur : Intel® core™ i5-7200u CPU
- Une RAM : 4.00 Go
- Sous un système d'exploitation Windows10

#### 3.2.2. Environnement logiciel :

Pour implémenter notre application, on a choisi comme outils de développement :

➤ Anaconda :

Anaconda est un outil essentiel pour les développeurs travaillant dans le domaine de la science des données. Anaconda est décrit comme un logiciel open source et gratuit conçu pour la programmation en Python et R. Il est largement utilisé dans des domaines tels que la science des données, l'apprentissage automatique et l'intelligence artificielle en raison de sa collection de packages, qui incluent Python, Numpy, Panda, Jupyter, etc. Ces packages sont essentiels pour effectuer diverses tâches liées à l'analyse de données et au développement de modèles d'apprentissage automatique.

L'article souligne également qu'Anaconda est multiplateforme, ce qui signifie qu'il peut être utilisé sur différents systèmes d'exploitation tels que Windows, macOS et Linux. Enfin, l'une des principales fonctions d'Anaconda est de permettre la collecte et la transformation de données à grande échelle grâce à ses outils intégrés. En résumé, Anaconda est présenté comme un outil indispensable pour les professionnels de la science des données, car il fournit une plateforme complète pour le développement, le

déploiement et la gestion de projets liés à l'analyse de données et à l'apprentissage automatique. [23]

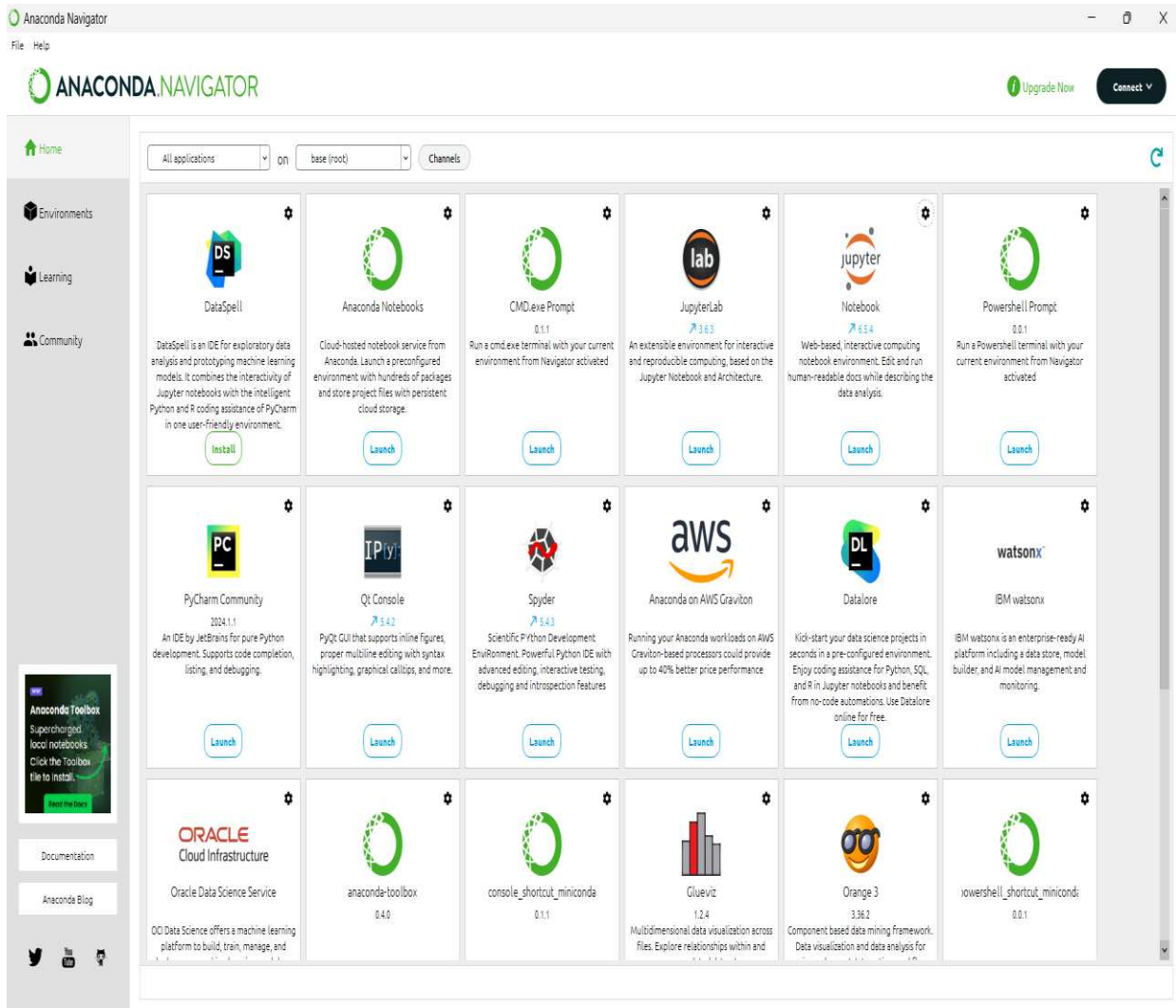


Figure 4.1 : Navigateur Anaconda.

➤ Le notebook Jupyter :

Le notebook Jupyter est une application web qui permet aux utilisateurs de créer et partager des documents interactifs contenant du code dynamique et exécutable, des visualisations de contenus, des textes de documentation et des équations. Le terme "notebook" est lié à la nature intrinsèque de l'outil qui permet d'écrire de petits morceaux de code exécutable (appelés "cellules"), de les documenter pour expliquer ce qu'ils font et d'afficher les données résultant de leur exécution. Tout cela est stocké dans un document partageable avec d'autres utilisateurs. [24]

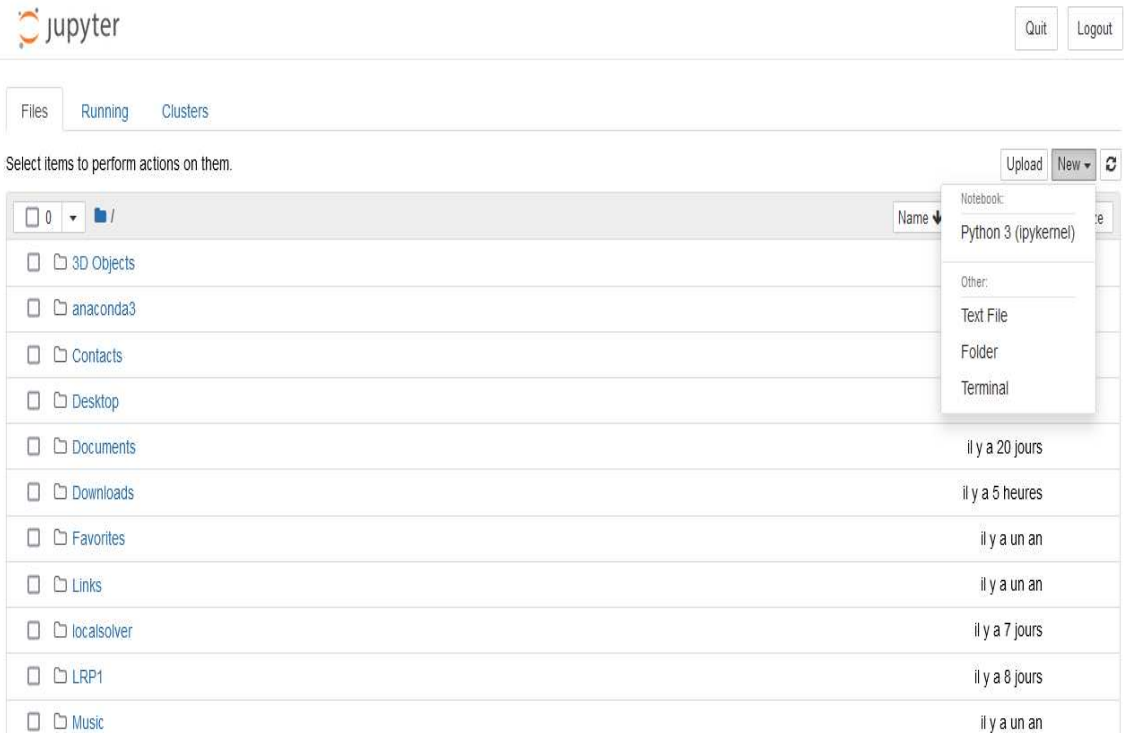


Figure 4.2: Interface de Jupyter Notebook.

### 3.3. Langage de programmation

Pour programmer notre approche, nous avons utilisé le langage de programmation Python. Python est un langage de programmation polyvalent, puissant et facile à apprendre, largement utilisé dans le développement logiciel, la science des données, l'apprentissage automatique et de nombreux autres domaines. Il se distingue par sa syntaxe claire et concise, sa gestion automatique de la mémoire et sa grande bibliothèque standard, qui offre une vaste gamme de fonctionnalités prêtes à l'emploi. Python prend en charge plusieurs paradigmes de programmation, y compris la programmation orientée objet, fonctionnelle et impérative, ce qui en fait un outil flexible pour répondre à divers besoins de développement. Sa popularité croissante est en grande partie due à sa simplicité et à sa polyvalence, ce qui en fait un choix privilégié pour les débutants ainsi que pour les développeurs expérimentés. [25]



Figure 4.3 : Logo du python.

### 3.4. Les jeux de données utilisés :

- Source :

Pour tester le fonctionnement de notre approche les jeux de données de Barreto sont utilisés. Ces jeux de données sont standards et utilisés dans plusieurs articles.

Les fichiers sont disponibles sur le lien suivant :[26]

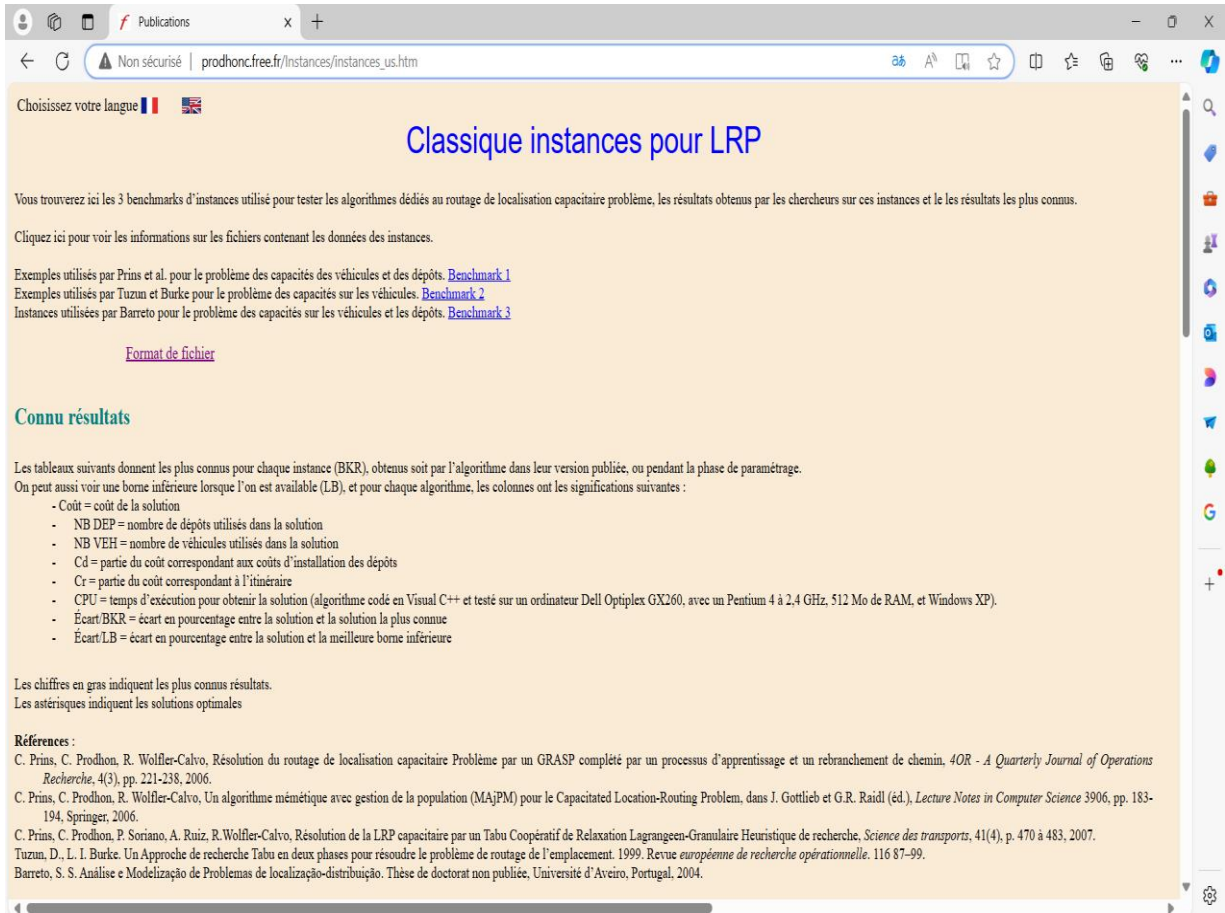


Figure 4.4 : Page source des instances.

Les propositions de Barreto consistent en un ensemble de fichiers de test utilisés dans la recherche scientifique, composé de 14 éléments (fichiers), où chaque fichier présente un contexte différent. Cela se traduit par une distribution différente des clients et des entrepôts, ce qui signifie que chaque fichier représente un cas différent. Cette diversité permet aux chercheurs d'explorer divers scénarios et d'évaluer les performances des algorithmes dans des contextes multiples et variés. Les fichiers ont la même structure. La différence réside dans la distribution des clients et des dépôts (coordonnées) sur un plan (x, y) et la capacité et le nombre des dépôts.

Name	Size	Packed	Type	Modified	CRC32
..			Dossier de fichiers		
coordChrist50.dat	767	296	Fichier DAT	15/06/2005 17:...	3675B67D
coordChrist75.dat	1 128	415	Fichier DAT	15/06/2005 17:...	779FFC3F
coordChrist100.d...	1 589	532	Fichier DAT	15/06/2005 17:...	3D42F271
coordDas88.dat	3 609	789	Fichier DAT	15/06/2005 17:...	B4D9F2DD
coordDas150.dat	6 405	1 571	Fichier DAT	15/06/2005 17:...	D90C5A97
coordGaspelle.dat	493	182	Fichier DAT	15/06/2005 17:...	82F6E700
coordGaspelle2....	491	191	Fichier DAT	15/06/2005 17:...	1098D145
coordGaspelle3....	609	214	Fichier DAT	15/06/2005 17:...	C4938469
coordGaspelle4....	653	243	Fichier DAT	15/06/2005 17:...	6C809DBC
coordGaspelle5....	654	243	Fichier DAT	15/06/2005 17:...	10E75F02
coordGaspelle6....	565	128	Fichier DAT	15/06/2005 17:...	59A59417
coordMin27.dat	1 018	263	Fichier DAT	15/06/2005 17:...	BE7E3308
coordMin134.dat	3 959	929	Fichier DAT	15/06/2005 17:...	577F051C
coordOr117.dat	5 391	1 119	Fichier DAT	15/06/2005 17:...	7CEC4A10

Figure4.5 : liste de fichiers de test proposés par Barreto

### 3.5. Structure des fichiers de test :

Les fichiers de test proposés sont des fichiers texte simples. Ils ont la même structure.

```
Fichier  Modifier  Affichage
50
5
10 49
20 30
5 25
54 17
43 53
37 52
9 49
2 64
0 26
0 30
1 47
7 63
1 62
2 33
51 21
42 41
31 32
5 25
12 42
36 16
52 41
27 23
17 33
13 13
57 58
62 42
42 57
16 57
8 52
7 38
27 68
30 48
43 67
58 48
58 27
37 69
Ln 1, Col 1  638 caractères  100%  Windows (CRLF)  UTF-8
```

Figure 4.6 : Une instance d'un fichier de test.

Chaque fichier de test proposé par Barreto est organisé d'une manière qui le rend facile à lire et à utiliser avec les algorithmes programmés. Barreto a défini la structure de tous ses ensembles de données dans un fichier texte particulier, ainsi que la méthode utilisée pour calculer la distance entre les points. En d'autres termes, il a fourni des instructions sur la façon dont les données sont organisées et sur la façon dont les calculs doivent être effectués pour utiliser efficacement ces fichiers de test.

```
The structure of the files is as follows:

number of customers
number of available depots

coordinates for the depots (x and y)

coordinates for the customers

vehicle capacity

depot capacities (for Tuzun instances, each one is equal to the total demand as there is no capacity on the depots)

customers demands

opening costs for the depots

opening cost of a route (cost of a vehicle)

0 or 1 (0 means that the costs are integer - 1 that costs are real)

To calculate the matrix distance (or the cost to link any 2 points A and B in the graph), we use the mathematical formula:

sqrt( (xA-xB)2 + (yA-yB)2 )

The results are stored in a float variable (in C language) if the costs are real (code 1)
The result is multiplied by 100 and truncated to be stored in an integer variable if the costs are integer (code 0).
```

**Figure 4.7** : Description de la structure des fichiers de test de Barreto [19]

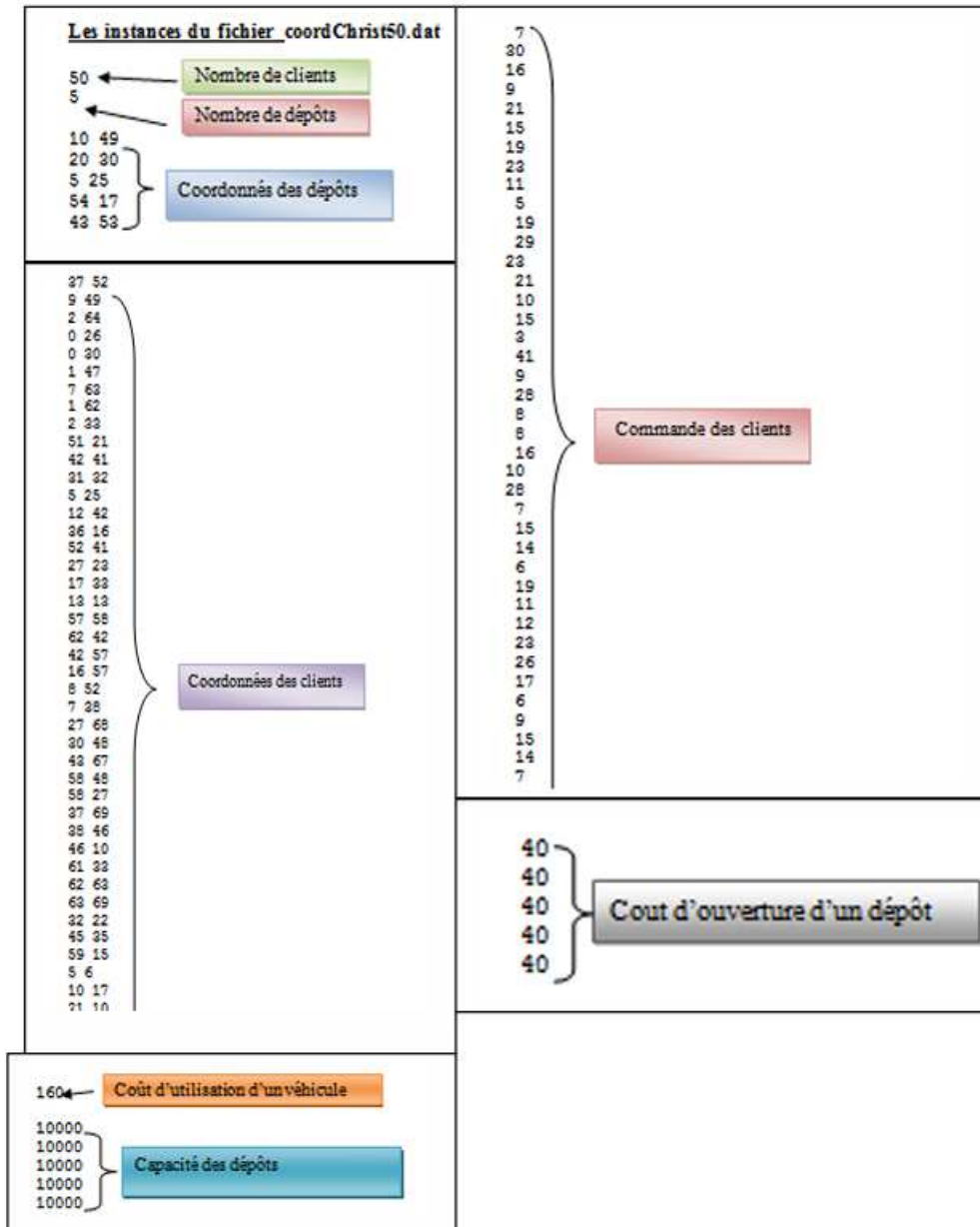


Figure 4.8: Structure d'un fichier de test. [19]

### 3.6. Exemple d'un résultat obtenu :

Le fichier ci-dessous indique les résultats pour l'instance « coordChrist50 » :

```
sys.exit(1)

instance_file = "coordChrist50.dat"
sol_file = sys.argv[2] if len(sys.argv) > 2 else None
str_time_limit = sys.argv[3] if len(sys.argv) > 3 else "1000"

simulated_annealing(instance_file, str_time_limit, sol_file)

Initial data read successfully.
Number of customers: 50
Number of depots: 5
Initial solution cost: 2350.062151245234
Best solution cost: 1497.7863311883887
Objective value (obj): 1497.7863311883887
Gap: 36.26609703089119%
Bounds: min = 2350.062151245234, max = 2350.062151245234
```

**Figure 4.9** : Résultat du recuit simulé.

Les résultats obtenus de l'application de l'algorithme de recuit simulé au problème de localisation et de routage (Location Routing Problem) montrent l'efficacité de cet algorithme dans l'amélioration des solutions initiales. Nous avons commencé par lire les données de base qui comprenaient un total de 50 clients et 5 dépôts. Le coût de la solution initiale aléatoire était de 2350.062151245234. Après l'exécution de l'algorithme de recuit simulé, cet algorithme a réussi à réduire le coût de la solution à 1497.7863311883887, reflétant ainsi une amélioration significative de 36.27%. Les valeurs minimales et maximales des coûts étaient identiques au départ, ce qui indique que la solution initiale n'était pas optimisée, mais l'algorithme a pu atteindre une solution bien meilleure au cours de son exécution. Ces résultats confirment l'efficacité de l'algorithme de recuit simulé dans l'amélioration des solutions pour les problèmes complexes de localisation et de routage.

✓ **Gap: 31. 26609703089119%**

Le gap mesure le pourcentage d'amélioration entre la solution initiale et la meilleure solution obtenue avec l'algorithme. Elle peut être calculée à l'aide de la formule suivante :

$$gap = \frac{initial\ cost - best\ cost}{initial\ cost} \times 10$$

Où :

- Coût Initial est le coût de la solution initiale.
- Meilleur coût est le coût de la meilleure solution trouvée.

✓ **Bounds:** (min = 2350.062151245234, max = 2350.062151245234)

Les bornes inférieure et supérieure des coûts des solutions nous permettent de connaître la plage des coûts pendant la recherche de la solution optimale. Dans ce cas, puisque nous commençons avec un coût initial et que nous l'améliorons, les bornes sont :

- Min Cost=Meilleur Coût.
- Max Cost=Coût Initial.

Donc, L'objectif de ce programme est de trouver une répartition optimale des clients entre les dépôts afin de minimiser le coût total de distribution et de transport, en tenant compte des contraintes de capacité des véhicules et des dépôts.

En résumé, le code lit les données du problème, génère une solution initiale aléatoire, améliore cette solution en utilisant l'algorithme du recuit simulé, et affiche enfin la meilleure solution trouvée avec quelques statistiques sur le coût.

### 3.7. La comparaison

Je vais analyser les résultats de l'algorithme du recuit simulé en fonction de la variation de la valeur du taux de refroidissement pour comprendre l'impact de cette variable sur les performances de l'algorithme. En modifiant la valeur du taux de refroidissement, je vais évaluer comment ce changement affecte la qualité des solutions générées et la capacité de l'algorithme à atteindre de meilleures solutions. Cette analyse renforcera notre compréhension du comportement interne de l'algorithme et aidera à déterminer les valeurs optimales des variables de l'algorithme, ce qui pourrait améliorer ses performances à long terme.

#### 3.7.1. Comparaison des taux de refroidissement

✓ **Paramètres d'exécution :** Nous exécutant l'algorithme, avec les paramètres du tableau.

- Taux de refroidissement (cooling rate) : 0.7, 0.8, 0.9.
- Température initiale (t\_initiale) : 1000000

## Chapitre 4 : Implémentation et résultats

- Température finale ( $t_{\text{finale}}$ ) : 5.

- Palier (equilibrium) : 1000.

- Exécutions

✓ **Résultat :**

Instances	Taux de Refroidissement = 0.7		Taux de Refroidissement = 0.9		Taux de Refroidissement = 0.8	
	Meilleure fonction objectif	Moyenne	Meilleure fonction objectif	Moyenne	Meilleure fonction objectif	Moyenne
CoordChrist50	1497.79	1574.77	1361.72	<b>1425.176</b>	1510.90	1547.06
CoordChrist75	3397.95	3427.95	3117.96	<b>3157.782</b>	3310.27	3351.74
CoordChrist100	2795.65	2919.24	2540.18	<b>2579.802</b>	2726.29	2823.604
CoordDas88	1619.42	1869.35	1415.89	<b>1621.38</b>	1427.80	1699.158
CoordDas150	172461.70	17650.53	158432.96	<b>161946.964</b>	167550.61	171898.74
CoordGaspelle	1193.61	1259.496	1141.18	<b>1207.054</b>	1136.28	1225.408
Coordgaspelle2	1313.20	1379.922	1203.93	<b>1254.716</b>	1271.51	1311.388
CoordGaspelle3	1016.12	1246.222	1163.45	<b>1211.458</b>	1170.40	1275.098
CoordGaspelle4	1075.88	1180.63	1054.22	<b>1067.266</b>	1077.24	1120.144
CoordGaspelle5	1011.71	1044.546	910.62	<b>920.812</b>	932.98	988.91
CoordGaspelle6	1278.45	1306.06	1153.57	<b>1188.766</b>	1188.04	1221.628
CoordMin27	8382.41	8533.652	6693.26	<b>7635.772</b>	8084.89	8438.654
CoordMin134	26637.37	28015.592	24701.29	<b>25157.66</b>	26174.50	27283.446

**Tableau 4.1** : Comparaison des taux de refroidissement.

### ✓ Discussion :

Le tableau présente les résultats de l'algorithme de recuit simulé pour différents taux de refroidissement (0.7, 0.9, 0.8) sur plusieurs instances. Les résultats incluent les meilleures solutions et les solutions moyennes obtenues pour chaque instance. Voici une analyse des résultats :

### ❖ Comparaison des taux de refroidissement :

- Taux de Refroidissement = 0.7 :

Les résultats avec un taux de refroidissement de 0.7 montrent une performance moyenne en général, où les meilleures solutions sont généralement plus élevées comparées au taux de refroidissement de 0.9. Par exemple, pour l'instance "CoordChrist50", la meilleure solution est de 1497.79 avec une moyenne de 1574.77, ce qui est moins efficace par rapport aux taux de refroidissement plus élevés.

- Taux de Refroidissement = 0.9 :

Le taux de refroidissement de 0.9 montre des améliorations claires dans la qualité des solutions. Les meilleures solutions sont généralement plus basses avec ce taux, indiquant une plus grande efficacité de l'algorithme. Par exemple, pour l'instance "CoordChrist50", la meilleure solution obtenue est de 1361.72 avec une moyenne de 1425.176, ce qui représente une amélioration notable par rapport au taux de refroidissement de 0.7.

- Taux de Refroidissement = 0.8 :

Le taux de refroidissement de 0.8 montre des résultats mixtes, mais reste meilleur que 0.7 et légèrement moins efficace que 0.9. Par exemple, pour l'instance "CoordChrist50", la meilleure solution obtenue est de 1510.90 avec une moyenne de 1547.06, ce qui est mieux que le taux de 0.7 mais pas au niveau du taux de 0.9.

### **Donc :**

Les résultats montrent que le taux de refroidissement a un impact significatif sur les performances de l'algorithme de recuit simulé. Le taux de refroidissement de 0.9 donne

généralement les meilleurs résultats, indiquant que ralentir davantage le refroidissement aide à améliorer la qualité des solutions finales. Le taux de refroidissement de 0.8 montre également de bonnes performances, mais légèrement inférieures à 0.9. Tandis que le taux de refroidissement de 0.7 tend à fournir des solutions moins efficaces.

Répéter les exécutions de l'algorithme permet de compenser la nature stochastique de l'algorithme et de fournir une estimation plus fiable de la qualité des solutions. Les résultats indiquent que le choix du taux de refroidissement approprié est un facteur crucial pour obtenir les meilleures performances possibles de l'algorithme.

### ✓ Comparaison de différents taux de refroidissement sous forme de courbe :

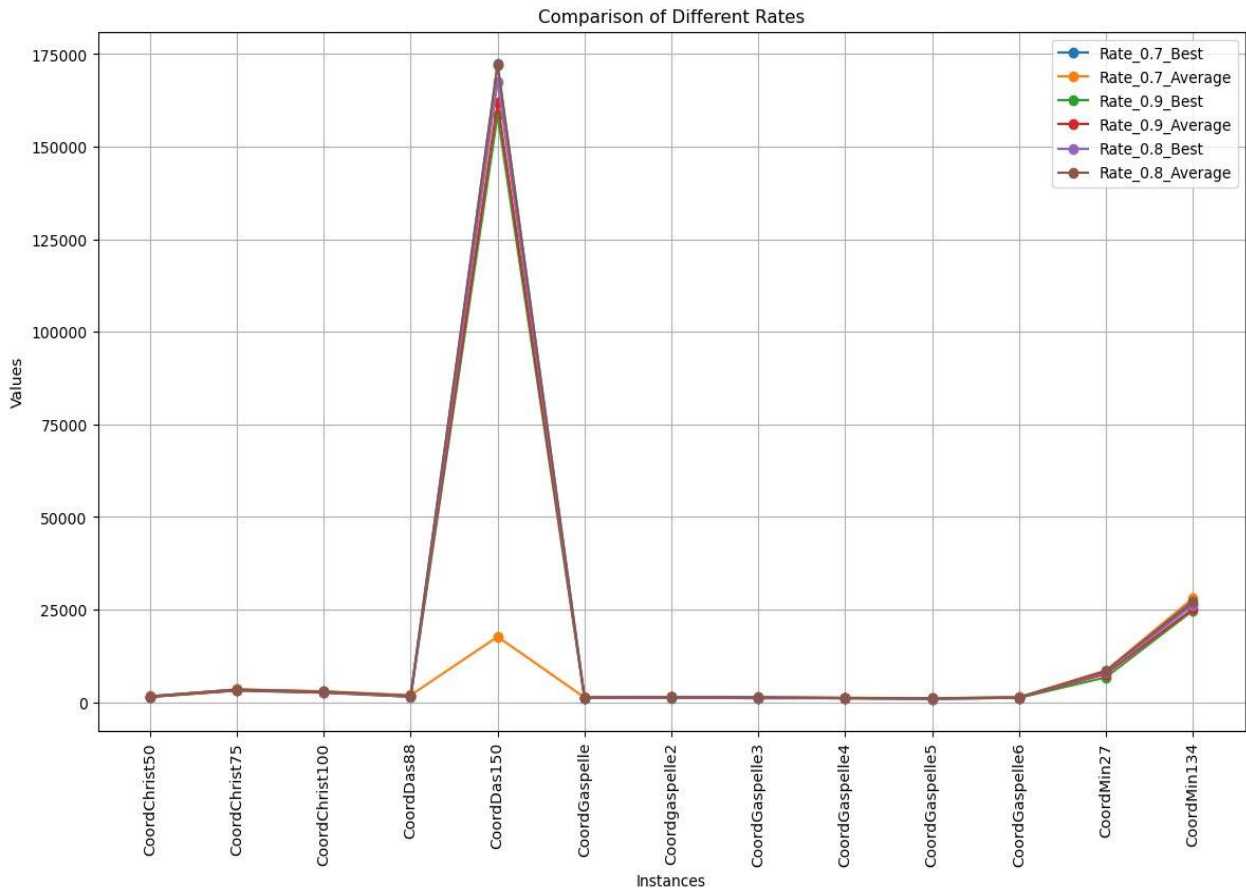


Figure 4.10 : Ajustement du taux de refroidissement.

### 3.7.2. Comparaison des températures initiales :

✓ **Paramètres d'exécution** : Nous exécutant l'algorithme, avec les paramètres du tableau.

- Taux de refroidissement (cooling rate) : 0.9.
- Température initiale (t\_initiale) : 1000000, 500000, 20000000.
- Température finale (t\_finale) : 5.
- Palier (equilibrium) : 1000.
- 5 exécutions

✓ **Résultats** :

Instances	T_initial = 1000000		T_initial = 500000		T_initial = 20000000	
	Meilleure fonction objectif	Moyenne	Meilleure fonction objectif	Moyenne	Meilleure fonction objectif	Moyenne
CoordChrist50	1418.44	1442.80	1412.08	<b>1426.896</b>	1414.75	1440.172
CoordChrist75	3147.82	3217.214	3001.35	<b>3150.499</b>	3177.63	3216.648
CoordChrist100	2426.03	2578.66	2452.49	<b>2549.352</b>	2580.26	2613.716
CoordDas88	1545.68	1635.692	1556.53	1660.59	1433.21	<b>1306.022</b>
CoordDas150	155727.99	161312.276	158263.95	162692.16	1610.66	<b>129552.366</b>
CoordGaspelle	1174.11	1212.282	1166.23	1204.832	1126.80	<b>982.888</b>
Coordgaspelle2	1180.63	<b>1243.57</b>	1220.86	1264.252	1209.51	1249.045
CoordGaspelle3	1054.31	1215.886	1126.43	<b>1214.954</b>	1074.22	1231.666
CoordGaspelle4	1030.02	1081.792	1054.19	1075.69	1007.00	<b>1036.856</b>
CoordGaspelle5	896.77	<b>933.126</b>	925.54	948.828	925.43	975.086
CoordGaspelle6	1181.39	1203.062	888.35	<b>932.816</b>	1084.23	963.352
CoordMin27	7809.56	13988.242	1120.22	<b>1153.676</b>	6763.77	7438.442
CoordMin134	24066.64	24798.388	24995.338	24995.338	24156.21	<b>24510.72</b>

**Tableau 4.2** : Comparaison des températures initiales.

### ✓ **Discussion :**

Le tableau présente les résultats de l'algorithme de recuit simulé pour différentes températures initiales (1 000 000, 500 000, 20 000 000) sur plusieurs instances. Les résultats incluent les meilleures solutions et les solutions moyennes obtenues pour chaque instance. Voici une analyse des résultats :

#### ❖ Comparaison par Température Initiale :

- Température Initiale = 1 000 000 :

Les résultats avec une température initiale de 1 000 000 montrent une performance assez cohérente mais généralement supérieure en termes de coûts par rapport aux autres températures. Par exemple, pour l'instance "CoordChrist50", la meilleure solution est de 1418.44 avec une moyenne de 1442.80, ce qui est moins efficace par rapport aux températures plus élevées.

- Température Initiale = 500 000 :

La température de 500 000 montre des améliorations dans certaines instances par rapport à 1 000 000, avec des meilleures solutions légèrement inférieures. Par exemple, pour l'instance "CoordChrist50", la meilleure solution obtenue est de 1412.08 avec une moyenne de 1426.896, indiquant une meilleure efficacité par rapport à une température initiale de 1 000 000.

- Température Initiale = 20 000 000 :

La température initiale de 20 000 000 montre des résultats variables, souvent avec des meilleures solutions plus optimales mais parfois avec des moyennes plus élevées. Par exemple, pour l'instance "CoordChrist50", la meilleure solution obtenue est de 1414.75 avec une moyenne de 1440.172, ce qui montre une performance comparable mais légèrement inférieure en moyenne par rapport à la température de 500 000.

### **Donc :**

Les résultats montrent que la température initiale a un impact significatif sur les performances de l'algorithme de recuit simulé. Une température initiale de 500 000 tend à produire des

solutions légèrement plus optimales dans certaines instances, bien que les performances spécifiques puissent varier. Une température initiale trop élevée comme 20 000 000 peut parfois améliorer les meilleures solutions, mais peut aussi entraîner une moyenne plus élevée des solutions.

Répéter les exécutions de l'algorithme permet de compenser la nature stochastique de l'algorithme et de fournir une estimation plus fiable de la qualité des solutions. Les résultats indiquent que le choix de la température initiale est un facteur crucial pour obtenir les meilleures performances possibles de l'algorithme.

- **Comparaison des températures initiales :**

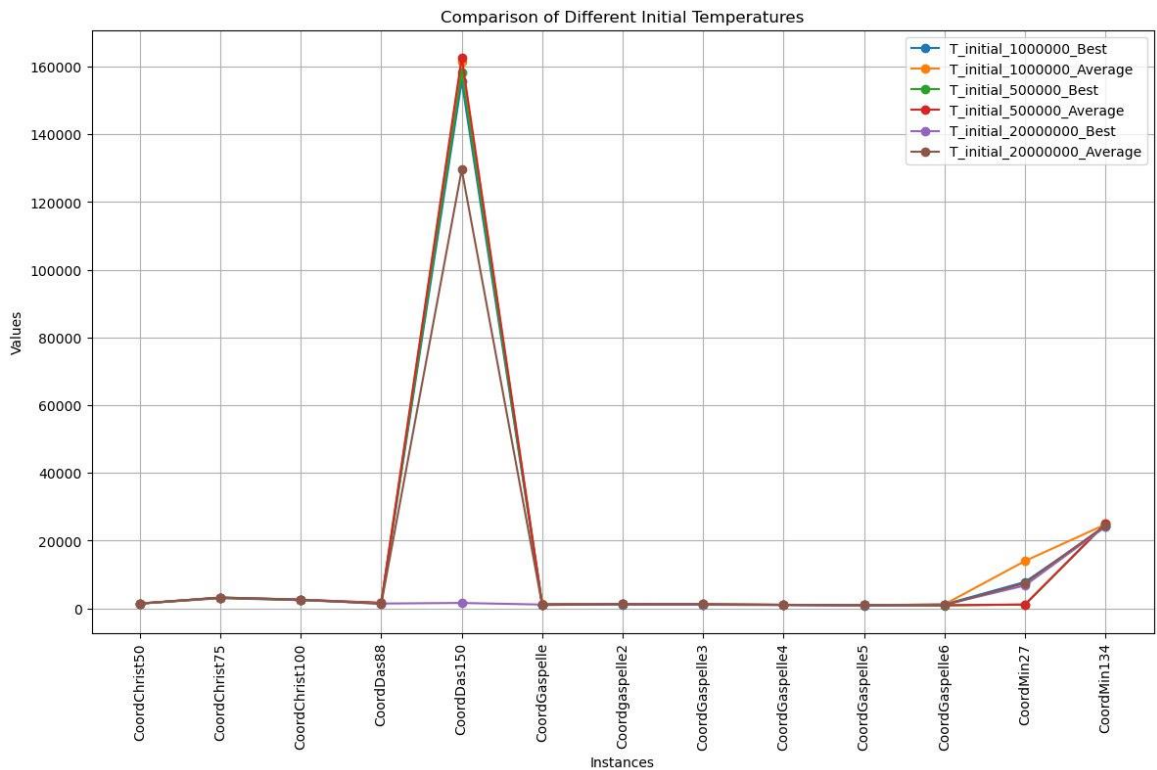


Figure 4.11: Ajustement des températures initiales.

### 3.7.3. Comparaison des paliers :

- ✓ **Paramètres d'exécution :** Nous exécutons l'algorithme, avec les paramètres du tableau.

## Chapitre 4 : Implémentation et résultats

- Taux de refroidissement (cooling rate) : 0.9.
- Température initiale( $t_{\text{initiale}}$ ) : 500000.
- Température finale (finale) : 5.
- Équilibre (equilibrium) : 10, 100, 1000.
- 5 exécutions.

✓ **Résultat :**

**Tableau 4.3 :** Comparaison des paliers.

Instances	Équilibre =10		Équilibre =100		Équilibre =1000	
	Meilleure fonction objectif	La moyenne	Meilleure fonction objectif	La moyenne	Meilleure fonction objectif	La moyenne
CoordChrist50	1644.71	1796.506	1523.69	1638.176	1433.20	<b>1490.464</b>
CoordChrist75	3813.30	3847.198	1577.54	3090.226	3171.09	<b>3290.352</b>
CoordChrist100	3476.77	3555.188	2803.59	2931.594	2534.19	<b>2667.478</b>
CoordDas88	1909.49	2132.99	1680.27	1900.782	1512.05	<b>1661.596</b>
CoordDas150	199229.96	202938.206	174380.13	182605.478	165013.75	<b>167183.594</b>
CoordGaspelle	1215.86	1298.49	1224.16	1279.89	1163.41	<b>1204.484</b>
Coordgaspelle2	1472.77	1532.21	1300.28	1379.364	1244.64	<b>1284.768</b>
CoordGaspelle3	1303.66	1463.02	1203.94	1352.248	1172.06	<b>1276.514</b>
CoordGaspelle4	1282.00	1316.182	1116.41	1154.572	1051.13	<b>1101.53</b>
CoordGaspelle5	1154.80	1203.21	944.62	1031.912	934.91	<b>969.432</b>
CoordGaspelle6	1388.07	1419.63	1255.26	1279.628	1202.17	<b>1237.728</b>
CoordMin27	8305.82	9288.186	7730.47	8210.398	7334.12	<b>8020.976</b>
CoordMin134	32179.52	27563.102	29218.61	29676.76	25576.27	<b>26163.306</b>

### ✓ Discussion

Le tableau présente les résultats de l'algorithme de recuit simulé pour différentes périodes d'équilibre (10, 100, 1000) sur plusieurs instances. Les résultats incluent les meilleures solutions et les solutions moyennes obtenues pour chaque instance. Voici une analyse des résultats :

### ❖ Comparaison par Période d'Équilibre :

- Période d'Équilibre = 10 :

Les résultats avec une période d'équilibre de 10 montrent des performances variables avec des coûts plus élevés comparés aux périodes d'équilibre plus longues. Par exemple, pour l'instance "CoordChrist50", la meilleure solution est de 1644.71 avec une moyenne de 1796.506, indiquant une efficacité moindre par rapport aux périodes d'équilibre plus longues.

- Période d'Équilibre = 100 :

La période d'équilibre de 100 montre des améliorations significatives dans certaines instances comparées à la période d'équilibre de 10, avec des meilleures solutions notablement inférieures. Par exemple, pour l'instance "CoordChrist50", la meilleure solution obtenue est de 1523.69 avec une moyenne de 1638.176, indiquant une meilleure efficacité par rapport à la période d'équilibre de 10.

- Période d'Équilibre = 1000 :

La période d'équilibre de 1000 montre les meilleurs résultats, avec des solutions optimales et moyennes ayant des coûts inférieurs. Par exemple, pour l'instance "CoordChrist50", la meilleure solution obtenue est de 1433.20 avec une moyenne de 1490.464, indiquant une performance améliorée par rapport aux périodes d'équilibre plus courtes.

### **Donc :**

Les résultats montrent que la période d'équilibre a un impact significatif sur les performances de l'algorithme de recuit simulé. Une période d'équilibre de 1000 tend à produire des solutions plus optimales dans la plupart des instances, bien que les performances spécifiques puissent varier. Les périodes d'équilibre plus courtes, comme 10, peuvent conduire à des résultats variables et des coûts plus élevés.

- **Comparaison des paliers :**

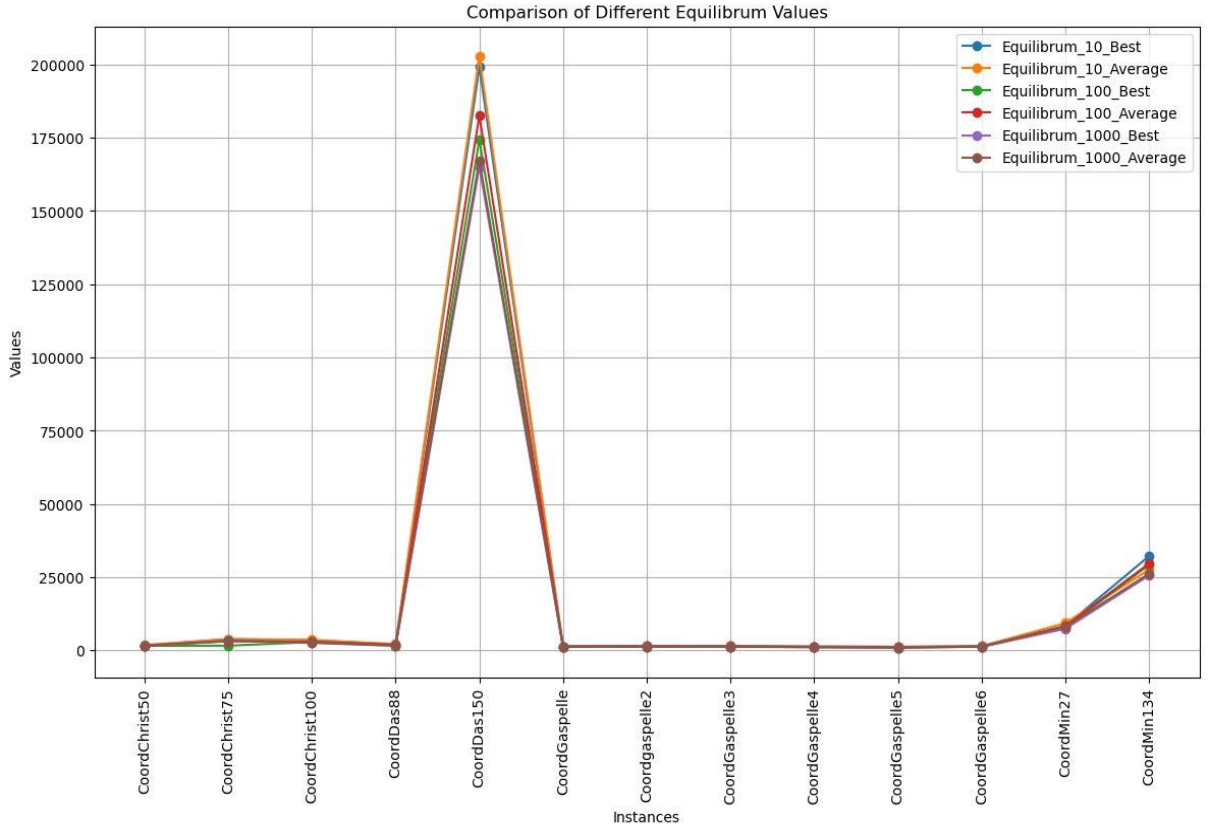


Figure4.12 : Ajustement des paliers.

### 3.8. Analyse des résultats :

Les résultats obtenus après 5 exécutions sont présentés dans les tableaux (...). Ces résultats visent à déterminer les valeurs optimales des paramètres d'exécution qui fournissent les meilleures performances dans un laps de temps réduit. Les analyses des trois tableaux révèlent ce qui suit:

1. Dans le premier tableau, représentant la fonction objectif par rapport refroidissement, il est observé que lorsque le taux de refroidissement est de 0.9, les meilleures solutions pour la fonction objectif ainsi que les meilleures moyennes de solutions sont obtenues dans tous les cas, à l'exception de l'instance « coordgraspelle » où la meilleure moyenne est obtenue mais la solution pour la fonction objectif se classe en deuxième position. Cette analyse indique que la meilleure valeur pour le taux de refroidissement est de 0.9.

2. Dans le deuxième tableau, représentant Comparaison des Valeurs Objectif avec la température initiale, une analyse similaire révèle que la meilleure valeur pour la température initiale est de 500000, ce qui donne les meilleures performances.

3. En ce qui concerne le troisième tableau, représentant Comparaison des Valeurs Objectif avec un Équilibre, il est remarqué que lorsque l'équilibre est fixé à 1000, les meilleures solutions sont obtenues dans la plupart des cas.

Ainsi, à travers l'analyse et la comparaison des résultats des trois tableaux, il devient évident que la détermination des paramètres optimaux n'est pas seulement une question technique, mais un élément crucial qui peut avoir un impact significatif sur l'efficacité et la performance des solutions fournies. Cela met en évidence l'importance de cette comparaison, qui se manifeste à plusieurs niveaux :

- **Amélioration des performances** : La détermination des paramètres optimaux contribue à améliorer l'efficacité et la performance de l'algorithme, permettant ainsi d'obtenir de meilleures solutions dans un laps de temps plus court, ce qui économise du temps et des ressources.
- **Réduction des coûts** : Dans les applications réelles, les solutions optimales peuvent réduire les coûts logistiques et opérationnels, par exemple en réduisant les distances parcourues, en économisant du carburant et en réduisant les coûts globaux.
- **Contribution scientifique** : Cette comparaison fournit une compréhension plus approfondie de l'impact des différents paramètres sur les performances de l'algorithme, ce qui peut servir de référence importante pour les recherches futures visant à améliorer les algorithmes et les problèmes optimaux.

## Conclusion

Dans ce chapitre, nous avons résolu le problème de localisation du routage (LRP) avec l'algorithme de recuit simulé. Nous avons présenté les résultats d'implémentation de notre approche en utilisant le langage de programmation Python. Les principes de nos approches ont été testés sur des jeux de données standard utilisés dans plusieurs articles qui traitent le même problème

### Conclusion générale

Dans ce mémoire, nous allons vu le domaine de recherche opérationnelle surtout le branche de l'optimisation combinatoire qui joue un rôle pour cherche le meilleur choix selon un critère donné parmi une ensemble des choix possibles pour des problèmes d'ordre théorique ou pratique.

Donc, l'objectif de ce travail de master était de proposer et d'implémenter une approche pour la résolution de problème de localisation du routage (LRP). Ce problème est omniprésent dans le domaine de commerce et de l'industrie. En général, il est lié directement aux services de livraison. L'importance de ce problème dans le secteur de commerce est due principalement aux couts engendrés par les opérations de livraisons.

La problématique de la localisation et de la tournée des véhicules (LRP) intègre de nombreux facteurs et variables, rendant sa résolution complexe et nécessitant des techniques et méthodes avancées. Ce travail vise à proposer des solutions efficaces et durables à cette problématique, en se concentrant sur l'amélioration des performances globales en termes de coûts, de temps et de ressources utilisées.

Dans ce travail, nous avons examiné divers concepts liés au domaine de l'optimisation et présenté un résumé complet des différentes méthodes utilisées dans ce domaine. Cette compréhension approfondie de ces méthodes nous a permis d'évaluer les recherches antérieures sur le problème de la localisation et de la tournée, nous fournissant ainsi une base solide pour développer de nouvelles approches.

À travers notre analyse de la littérature existante, nous avons pu identifier les lacunes de la recherche et orienter nos efforts vers le développement d'une méthodologie intégrée basée sur le recuit simulé et des techniques d'optimisation avancées. Cette méthodologie a été conçue dans le but d'atteindre un équilibre optimal entre la réduction des coûts et l'amélioration de l'efficacité de la livraison.

Les deux derniers chapitres sont consacrés à la proposition et à la mise en œuvre d'une approche de recuit simulé inspirée des méthodes étudiées dans les chapitres précédents. Les principes de notre méthode ont été appliqués et testés sur des ensembles de données standards (ensembles de données de Barreto). Les orientations futures de ce travail concernent notamment l'amélioration des paramètres de l'algorithme, en étudiant l'impact de différents paramètres tels que le degré de refroidissement, le taux de refroidissement et le nombre d'itérations sur les performances de

## **Conclusion générale**

---

l'algorithme, ainsi que le développement de techniques adaptatives pour ajuster automatiquement ces paramètres afin d'améliorer les résultats.

## Bibliographie

- [1] H. Allaoua, «Support de cours d'optimisation combinatoire Focus sur les méthodes de résolution approchée,» Mohamed Boudiaf, m'sila, 2017.
- [2] D. Ali, «Méthodes de Résolution Exactes Heuristiques et Métaheuristiques,» Université de Mohamed Boudiaf , M'sila, 2023-2024.
- [3] A. SOUABNI et A. E. BELHOUL , «L'algorithmes Séquentielle (Réduit Simulé+Algorithme 2-Opt Pour La Résolution Du Problème De Routage De Véhicules avec Capacité (PRVC),» Université DE Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, 2020/2021.
- [4] S. LEILA, «Optimisation MultiObjectifs par Programmation Génétique,» l'université de BATNA, IBATNA, 2007.
- [5] A. Berro, «Optimisation multiobjectif et stratégies d'évolution en environnement dynamique,» l'Université des Sciences Sociales Toulouse I, 2001.
- [6] C. Mancel, «Modélisation et résolution de problèmes d'optimisation combinatoire issus d'applications spatiales,» Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS, 2004.
- [7] B. Meriem, «Le problème du voyageur de commerce:solution exacte et approchée,» UNIVERSIT... DE MÍSILA, MÍSILA, 2014/2015.
- [8] D. T. Rafael , R. M.-T. Jairo , Carlos L. Quintero-Araujo, Javier Panadero et Angel A. Juan, «The location routing problem with facility sizing decisions,» *INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH*, p. 945, 2022.
- [9] a. i. almouhanna, «Biased Randomised Heuristics Biased Randomised Heuristics Location Routing Problem,» university of portsmouth, 2019.
- [10] J. William et R. GUERRERO , «Modèles et méthodes d'optimisation pour le problème de localisation-routage avec contraintes de stockage,» l'U NIVERSITE DE TECHNOLOGIE DE TROYES, 2014.
- [11] B. Issam, «Geolocation web application for vehicule routing problem based on genetic algorithms,» UNIVERSITE MOHAMED BOUDIAF, M'SILA, 2015/2016.
- [12] A. Ayache, «Résolution du problème de tournées de véhicules avec fenêtre de temps,» UNIVERSITE MOHAMED BOUDIAF - M'SILA, M'SILA, 2018 /2019.
- [13] K. BENNOUR et Z. BOUBAAYA , «Le problème de routage de véhicule avec capacité par l'algorithme de colonies de fourmis CVRP,» UNIVERSITE MOHAMED EL BACHIR EL-IBRAHIMI, BORDJ BOU ARRERIDJ, 2020/2021.
- [14] N. Hadbaoui et K. Nouibat , «Problème de Tournée du Véhicule avec Fenêtre de Temps (PTVFT),» UNIVERSITE MOHAMED BOUDIAF, M'SILA, 2020.
- [15] N. Boudjerda et I. Djarit , «Le problème de routage dynamique des véhicules avec fenêtres de temps,» UNIVERSITÉ MOHAMMED SEDDIK BENYAHIA, JIJEL, 2022.
- [16] M. Fouad, «Résolution d'un problème stochastique de localisation allocation avec gestion des indisponibilités des sites,» Université de Tlemcen, Tlemcen, 2020.
- [17] A. Mingozzi, «The Multi-depot Periodic Vehicle Routing Problem,» University of Bologna, Italy, 2005.

- [18] G. Michel , P. Jean-Yves , B. Olli , H. Geir et L. Arne , «Metaheuristics for the Vehicle Routing Problem and its Extensions : A Categorized Bibliography,» *cirreft*, p. 25, 2007.
- [19] S. HETTAB et A. LATRACHE , «Une Approche pour la Résolution et l’Optimisation d’un Problème de Tournées de Véhicule (Capacitated Location Routing Problem (CLRP)),» Universitaire Abd Elhafid Boussouf, Mila, 2020/2021.
- [20] P. Sattrawut , T. Eiichi et Q. Ali Gul , «integrated Approach for Location-Routing Problem using Branch-and-Price Method,» p. 5.
- [21] «ResearchGate,» Jan 2004. [En ligne]. Available at: [https://www.researchgate.net/figure/Organigramme-de-lalgorithme-du-recuit-simule-applique-au-decoupage-aerien\\_fig5\\_237301009](https://www.researchgate.net/figure/Organigramme-de-lalgorithme-du-recuit-simule-applique-au-decoupage-aerien_fig5_237301009). [Accès le 2024 MAI 09].
- [22] B. Amine, «Le recuit simulé,» Université Mohamed Boudiaf, Oran, 2011.
- [23] «Data Transition Numérique,» [En ligne]. Available: <https://www.data-transitionnumerique.com/anaconda-python/>. [Accès le 26 mai 2024].
- [24] «openclassrooms,» [En ligne]. Available: <https://openclassrooms.com/fr/courses/6204541-initiez-vous-a-python-pour-lanalyse-de-donnees/6211211-faites-vos-premiers-pas-avec-jupyter-notebook>. [Accès le 26 mai 2024].
- [25] «Python,» [En ligne]. Available: <https://docs.python.org/fr/3.12/tutorial/>. [Accès le 26 mai 2024].
- [26] «free,» [En ligne]. Available: [http://prodhonc.free.fr/Instances/instances\\_us.htm](http://prodhonc.free.fr/Instances/instances_us.htm).. [Accès le 26 mai 2024].
- [27] B. R. Heuristics, «Biased Randomised Heuristics for Location Routing Problem,» university of portsmouth, 2019.
- [28] P. Sattrawut , T. Eiichi et Q. Ali Gul , «Integrated Approach for Location-Routing Problem using Branch-and-Price Method,» Kyoto University, 2013.

مشكلة تحديد المواقع وتوجيه المركبات هي مشكلة لوجستية معقدة تتعلق باختيار المواقع المثلى لمراكز التوزيع وتخطيط مسارات التوصيل الفعالة للمركبات بهدف تقليل التكاليف الإجمالية للتشغيل والنقل. تشمل هذه المشكلة تحديات تتعلق بتحديد المواقع المناسبة للمستودعات وتخطيط مسارات المركبات لتلبية طلبات العملاء المتغيرة ضمن قيود معينة مثل سعة المركبات والأطر الزمنية للتسليم. لحل هذه المشكلة، نستخدم تقنية التبريد المحاكي، وهي خوارزمية تحسينية تعتمد على مبادئ الديناميكا الحرارية للوصول إلى حلول تقريبية فعالة.

**الكلمات المفتاحية:** مشكلة الموقع والتوجيه، التحسين التوافقي، ميتاهيورستيك، محاكاة التلدين.

---

## Résumé

Le problème de localisation et de routage des véhicules (LRP) est un problème logistique complexe qui consiste à choisir les emplacements optimaux pour les centres de distribution et à planifier les itinéraires de livraison efficaces des véhicules afin de minimiser les coûts opérationnels et de transport totaux. Ce problème comporte des défis liés à la détermination des emplacements appropriés pour les entrepôts et à la planification des itinéraires des véhicules pour répondre aux demandes variables des clients, tout en respectant des contraintes telles que la capacité des véhicules et les fenêtres temporelles de livraison. Pour résoudre ce problème, nous utilisons la technique du recuit simulé (Recuit Simulé), un algorithme d'optimisation qui s'appuie sur les principes de la thermodynamique pour atteindre des solutions approximatives efficaces.

**Mots clés :** PLR, Optimisation Combinatoire, Métaheuristiques, Recuit Simulé.

---

## Summary

The Location Routing Problem and (LRP) is a complex logistical issue that involves selecting optimal locations for distribution centers and planning efficient delivery routes for vehicles to minimize total operational and transportation costs. This problem encompasses challenges related to determining suitable locations for warehouses and planning vehicle routes to meet varying customer demands, while adhering to constraints such as vehicle capacity and delivery time windows. To solve this problem, we utilize the simulated annealing technique (Simulated Annealing), an optimization algorithm that relies on principles of thermodynamics to achieve effective approximate solutions.

**Keywords :** LRP, Combinatorial Optimization, Metaheuristics, Simulated Annealing.

---