

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Réf :.....

Centre Universitaire
Abd Elhafid Boussouf Mila

Institut des Sciences et Technologie

Département de Mathématiques et Informatique

**Mémoire préparé en vue de l'obtention du diplôme de
Master
En : Informatique**

**Spécialité : Sciences et Technologies de l'Information et de la
Communication (STIC)**

**Une Approche pour la Résolution et l'Optimisation
d'un Problème de Tournées de Véhicule (Capacitated
Location Routing Problem (CLRP))**

**Préparé par : HETTAB SEDDIK
LATRACHE ABDARRAZAKE**

Soutenue devant le jury

Encadré par ZEKIOUK MOUNIRA	MAA
Président AFRI FAIZA	MAA
Examineur TALAI MERIEM	MAA

Année Universitaire : 2020/2021

REMERCIEMENT

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce Modeste travail.

Nous présentons nos sincères gratitude et remerciements à notre encadreur

Zekiouk Mounira pour le grand soutien morale qu'il nous a apporté au

cours de notre projet et aussi pour son aide précieuse et ses conseils

judicieux qu'il nous a fait bénéficier. Nos remerciements très sincères, vont

aussi à tous les membres du personnel du département informatique, et en

particulier, à tous ceux qui nous ont enseigné durant toutes nos années

d'études universitaires. Qu'il trouve ici l'assurance de notre profonde

gratitude. En fin, à tous ceux qui ont contribué de près ou de loin à

l'achèvement de ce travail.

Tous nos remerciements à : **DJABER, YUCEF** pour les merveilleux

moments qu'on a passé ensemble, et à tous nos amis sans exception.

Dédicaces

Seddik

Je dédie ce modeste travail

À mes chers parents, ma chère MAMAN (رحمة الله عليها) qui sont la cause de mon existence dans cette vie, pour leur soutien, leur patience et leur amour qui m'ont donné la force pour continuer mes études

À mes frères : ACHOUR, MOHAMED ET HOUSSAM

À mes sœurs : HAYET, AWATEF, SOLEF, IBTISEM, AMINA,
NOURHANE

À mes amis : DJABER, YUCEF, WALID, YAKOUB, ISSAM ET MOURAD et à tous mes amis du proche ou du loin sans exception, dont la liste est longue

À toute ma famille.

ABDARAZAKE

À mes très chers Parents pour leur affection, leur patience, leur soutien et leurs encouragements qui m'ont permis d'arriver au bout de ce travail

À mon frère que je l'aime énormément :

À mes sœurs À tous mes amis surtout.....

À toute ma famille.

Résumé

Le problème étudié dans ce travail s'inscrit dans le domaine du transport. Il traite la notion de localisation-routage sous contrainte de capacité ou CLRP (Capacited Location Routing Problem), un sous type du problème général de tournées de véhicule (MDVRP Problème de tournées de véhicules avec dépôts multiples). Ce dernier combine deux niveaux de décision. Un choix doit se faire sur un sous-ensemble de dépôts à ouvrir, niveau de décision stratégique, et des tournées de véhicules doivent être élaborées, niveau tactique ou opérationnel, dans le but de minimiser les coûts totaux, Il appartient au champ de l'optimisation combinatoire.

Pour résoudre ce problème nous proposons une approche génétique basée sur les principes des algorithmes génétiques (sélection, croisement et mutation). Notre approche est basée sur un codage simple pour faciliter l'application des opérations de croisement et de mutation entre les différentes solutions des populations. L'approche est testée sur des jeux de données standards (instances de Barreto)

Mots-clés : le problème de tournées de véhicules multi-dépôts, l'optimisation, CLRP, algorithme génétique, croisement, mutation, instance de Barreto, Java

Abstract

The problem studied in this work falls within the field of transport. It deals with the notion of location-routing or LRP (Location Routing Problem) a sub-type of the general vehicle routing problem (MDVRP Vehicle routing problem with multiple depots). It combines two decision levels. A choice must be made on a subset of depots to be opened, strategic decision level, and vehicle routes must be developed, tactical or operational level, in order to minimize total costs, it is within the scope of the combinatorial optimization.

For this we propose a genetic approach based on the principles of genetic algorithms (selection, crossing and mutation). Our approach is based on a simple coding to facilitate the application of crossover operations between the different solutions.

Keywords: multi-depot vehicle routing problem, optimization, CLRP, genetic algorithm. Java

الملخص

تندرج المشكلة التي تمت دراستها في هذه الرسالة ضمن مجال النقل. إنه يتعامل مع مفهوم توجيه الموقع أو LRP وهو نوع فرعي من مشكلة توجيه مسار المركبات العام (MDVRP مع مستودعات متعددة). يجمع الأخير بين مستويين من القرار. يجب أن يتم الاختيار على مجموعة فرعية من المستودعات التي سيتم فتحها ، ويجب تطوير مستوى القرار الاستراتيجي ، ويجب تطوير طرق المركبات ، على المستوى التكتيكي أو التشغيلي ، من أجل تقليل التكاليف الإجمالية ، فهو يقع في نطاق التحسين الاندماجي. لهذا نقترح نهجاً جينياً يعتمد على مبادئ الخوارزميات الجينية (الاختيار والعبور والطفرة). يعتمد نهجنا على ترميز بسيط لتسهيل تطبيق عمليات التهجين بين الحلول المختلفة

الكلمات المفتاحية: مشكلة توجيه المركبات متعددة المستودعات ، التحسين ، CLRP ، الخوارزمية الجينية. جافا

Table des matières

Introduction générale	12
Chapitre 01 : L'optimisation et les approches d'optimisation	14
1. Introduction	15
2. L'optimisation	15
2.1 Définition.....	15
2.2 Modélisation d'un problème d'optimisation	16
3. Méthodes de résolution d'un problème d'optimisation	17
3.1 Méthodes Exactes.....	17
3.2 Méthodes Approchées.....	17
3.2.1 Heuristiques	17
3.2.1.1 Les heuristiques de constructions.....	18
3.2.1.2 Heuristiques d'insertions.....	18
3.2.1.3 Heuristiques deux phases	18
3.2.2 Les méta-heuristique.....	18
3.2.2.1 Méthodes à base d'une solution	19
3.2.2.2 Méthodes à base de populations.....	22
3.3 Stratégies recherche.....	24
3.3.1 Intensification et Diversification	24
3.3.2 Hybridation des Méthodes	25
3.3.3 Choix d'une structure de voisinage.....	25
4. Classification des Méta-heuristiques.....	25
4.1 Approche trajectoire.....	25
4.2 Approche phénomène naturel.....	25
4.3 Approche mémoire.....	26

4.4 Approche voisinage.....	26
4.5 Approche selon la fonction objective.....	26
5. Conclusion.....	26
Chapitre 2 : Le Problème de Tournées de Véhicules et le CLRP.....	27
1. Introduction.....	28
2. Le problème de VRP	28
2.1 Définition.....	28
2.2 VRP de base.....	28
2.3 Modélisation du VRP	29
2.4 Variantes du problème VRP	30
2.4.1 Problèmes de tournées de véhicules avec fenêtres de temps (VRPTW).....	30
2.4.2 Le problème de tournées de véhicules avec livraison et collecte (VRPB)	31
2.4.3 Le problème de tournées de véhicules avec ramassage et livraison (VRPSD)	31
2.4.4 Le problème de tournées de véhicules Multi-trips (VRPMT).....	31
2.4.5 Le problème de tournées de service (TRP).....	31
2.4.6 Le problème de tournées de véhicules multi-dépôts (MDVRP).....	32
3. Le problème de CLRP	32
3.1 Présentation du problème	32
3.2 Description mathématique	33
3.2.1 Notation	33
3.2.2 Modélisation.....	33
4. Méthodes proposées pour le CL RP.....	35
4.1 Méthode 1 : un algorithme mimétique pour le CLRP.....	36
4.1.1 Principes.....	36
4.1.2 Résultats.....	38
4.2 Méthode 2 : une approche GRASP+ ELS pour le CLRP.....	38

4.2.1 Principes.....	38
4.2.2 Résultats.....	40
4.3 Méthode 3 : une heuristique pour le CLRP	40
4.3.1 Principes.....	40
4.3.2 Résultats	41
4.4 Méthode 4 : un algorithme évolutionnaire simple et efficace pour le CLRP.....	42
4.4.1 Principes.....	42
4.4.2 Résultats	44
5. Conclusion.....	44
Chapitre 3 : Une Approche Génétique pour le CLRP.....	45
1. Introduction.....	46
2. Algorithme génétique	46
2.1 Principes.....	46
2.1.1 Le principe de variation.....	46
2.1.2 Le principe d'adaptation	46
2.1.3 Le principe d'hérédité	46
2.2 Paradigme	46
2.3 Opération d'évolution	47
2.3.1 Sélection	47
2.3.2 Croisement	47
2.3.3 Mutation	47
2.4 Algorithme	47
3. Une approche génétique pour le CLRP.....	48
3.1 Codage des solutions.....	48
3.2 Construction de la population initiale.....	49
3.3 Fonction d'évaluation.....	49
3.4 Opération de croisement.....	50
3.5 Opération de mutation.....	52
3.5.1 Changement des dépôts.....	52
3.5.2 Changement des clients.....	52
3.6 Opération de sélection.....	52
4. Conclusion.....	53
Chapitre 04 : Implémentation et Testes.....	54
1. Introduction.....	55

2. Langage de programmation	55
3. Jeux de données utilisés	55
3.1 Source.....	55
3.2 Structure des fichiers de test	57
4. Interfaces de l'application.....	58
5. conclusion.....	59
Conclusion Générale.....	60

<i>List des figures</i>	
Figure 1.1 : Classification des méthodes d'optimisation.....	17
Figure 2.1 : le problème de tournée de véhicule VRP.....	29
Figure 2.3 : problème de CLRPP.....	33
Figure 2.4 : Modélisation du problème de Localisation et Routage des véhicules.....	34
Figure 2.5 : Modélisation du problème de Localisation et Routage des véhicules.....	35
Figure 2.6 : codage de la solution méthode 1.....	36
Figure 2.7 : opérateur de croisement.....	36
Figure 2.10 : opérateur de voisinage.....	38
Figure 2.12 : opérateurs de voisinage.....	39
Figure 2.13 : 2-OPT.....	41
Figure 2.14 : 3-OPT	41
Figure 2.15 : Inverse 2-opt.....	41
Figure 2.18: opérateur de croisement.....	42
Figure 2.19 : opérateur de mutation.....	43
Figure 3.1 : étapes des algorithmes génétiques.....	48
Figure 3.2 : Codage des solutions.....	48
Figure 3.3 : organigramme de la solution initial.....	49
Figure 3.4 : Fonction d'évaluation	50
Figure 3.5 : Opération de croisement	51
Figure 4.1 : liste des fichiers de Benchmark.....	55
Figure 4.2 : liste des fichiers de test proposés par barreto.....	56
Figure 4.3 : contenu d'un fichier de test.....	56
Figure 4.4 : description de la structure des fichiers de test barreto.....	57

Figure 4.6 : Interface des listes dépôts et clients.....	58
Figure 4.7 : Interface des solutions.....	59
Figure 4.8 : Interface d'une solution.....	59

Liste des algorithmes

Figure 1.2 : la méthode de recuit simulé.....	19
Figure 1.3 : la méthode de descente simple.....	20
Figure 1.4 : la méthode de la recherche taboue.....	21
Figure 1.5 : la méthode GRASP.....	22
Figure 1.6 : la méthode de colonie de fourmis.....	24
Figure 2.8 : algorithme méthode 1.....	37
Figure 2.9 : algorithme recherche locale.....	37
Figure 2.11 : algorithme méthode 2.....	39
Figure 2.16 : algorithme méthode 3.....	41
Figure 2.20 : algorithme méthode 4.....	43

Liste des acronymes

Recherche opérationnelle (RO)	16
Intelligence artificielle (IA)	16
Problème d'optimisation (PO)	16
MDVRP (Problème de tournées de véhicules avec dépôts multiples)	17
LRP (Location Routing Problem)	18
Problème de tournées de véhicule (VRP)	18
Voyageur de commerce (TSP)	18
Recherche taboue (TS)	20
GRASP (Greedy Randomized Adaptive Search Procedure)	21
CVRP (Problème de tournées de véhicules avec contraintes de capacité ou Capacitated VRP).....	29
VRPTW (Problèmes de tournées de véhicules avec fenêtres de temps)	31
VRPB (Le problème de tournées de véhicules avec livraison et collecte)	31
VRPSD (Le problème de tournées de véhicules avec ramassage et livraison)	31
VRPMT (Le problème de tournées de véhicules Multi-trips)	31

TRP (Le problème de tournées de service)	31
MDVRP (Le problème de tournées de véhicules multi-dépôts)	32
CLRP (Capacitated Location-Routing Problem)	32

Liste des tables

Tableau 1 : résultat des tests de la méthode 1.....	38
Tableau 2 : résultat des tests de la méthode 2.....	40
Tableau 3 : résultat des tests de la méthode 3.....	42
Tableau 4 : résultat des tests de la méthode4.....	44

Introduction générale

De nos jours, dans un environnement économique de plus en plus concurrentiel, l'optimisation des chaînes de production occupe une place importante dans la vie quotidienne des entreprises. Le but étant l'amélioration de leur compétitivité en optimisant leurs performances et en réduisant leurs coûts. En effet, la chaîne de production englobe toutes les tâches réalisées pour la production d'un service ou d'un produit, à savoir l'approvisionnement, la transformation, le stockage, et la livraison. Bien que, la chaîne de production englobe toutes ces tâches, des études ont montré que les coûts de transport et de livraison constituent le tiers des coûts opérationnels d'une chaîne logistique.

Par conséquent, depuis plusieurs années, un nombre croissant d'entreprises et de chercheurs constatent les bénéfices de l'optimisation de la logistique de distribution et de livraison, et elle est devenu un challenge majeur pour eux. Dans ce sens, l'élaboration des tournées de véhicules pour l'approvisionnement en matières premières et la livraison des produits finis aux clients constitue l'une des activités principales de la logistique de distribution/approvisionnement et de transport. L'optimisation des activités de distribution/livraison consiste à résoudre en partie les problèmes de tournées de véhicules.

Le problème étudié dans ce mémoire s'inscrit dans le domaine de livraison. Il traite la notion de localisation-routage avec contrainte de capacité ou CLRP (Capacited Location Routing Problem) un sous type du problème général de tournées de véhicule (MDVRP Problème de tournées de véhicules avec dépôts multiples). Ce dernier combine deux niveaux de décision. Un choix doit se faire sur un sous-ensemble de dépôts à ouvrir, niveau de décision stratégique, et des tournées de véhicules doivent être élaborées, niveau tactique ou opérationnel, dans le but de minimiser les coûts totaux, Il appartient au champ de l'optimisation Combinatoire

L'objectif principal à travers ce projet consiste à proposer et implémenter une approche permettant la résolution et l'optimisation des coûts de transport/livraison dans une entreprise de production/livraison multi-dépôt en se basant sur les approches générales proposées dans le domaine d'optimisation.

Pour cela nous proposons une approche évolutionnaires basée sur les principes des algorithmes génétiques (sélection, croisement et mutation). Notre approche propose un codage simple pour faciliter l'application des opérations de croisement et de mutation entre les différentes solutions des populations.

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.

- Dans le deuxième chapitre, nous présenterons un état de l'art sur le problème général de tournées de véhicules ainsi qu'une modélisation et une définition correcte du problème étudié dans le mémoire CLRP . Nous clôturons ce chapitre par une énumération des méthodes et des résultats collectés de quelques articles récents qui ont déjà traité le même problème.
- Dans le troisième chapitre, nous expliquerons les principes de notre approche génétique utilisée pour la résolution du problème. Ce chapitre résume le codage utilisé pour la présentation des chromosomes (solution) ainsi que la description et l'explication des opérations de croisement et de mutations cœur des approches génétiques.
- 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 01 :

Notion de base : l'optimisation et les approches d'optimisation

1. Introduction

La résolution des problèmes d'optimisations rencontrés dans notre vie quotidienne est assez délicate puisque le nombre fini de solutions réalisables croît généralement avec la taille du problème, ainsi que sa complexité. Cela a poussé les chercheurs à développer de nombreuses méthodes de résolution en recherche opérationnelle (RO) et en intelligence artificielle (IA), capable d'améliorer leurs performances en termes de temps de calcul nécessaire et/ou de la qualité de la solution proposée. Ces approches de résolution peuvent être classées en deux catégories : les méthodes exactes et les méthodes approchées. De nombreuses applications pouvant être modélisées sous la forme d'un problème d'optimisation (PO) telles que le problème du voyageur de commerce, l'ordonnancement de tâches, le problème de la coloration de graphes, etc.

Dans ce chapitre, Nous allons présenter quelques notions liées au domaine d'optimisation ainsi qu'un récapitulatif des différentes techniques d'optimisation.

2. L'optimisation

2.1 Définition

L'optimisation réfère au fait de choisir le meilleur élément parmi un ensemble de cas alternatifs favorables.

Un problème d'optimisation est défini par un espace de recherche, une ou plusieurs fonctions objectives et un ensemble de contraintes.[35]

2.2 Modélisation d'un problème d'optimisation

Que ce soit en mathématique ou en informatique, l'étude et la résolution d'un problème commencent toujours par la modélisation de celui-ci. Les problèmes d'optimisation n'échappent pas à cette loi et sont modélisés comme suit :

Un problème d'optimisation est noté : $P(S, f)$ et il est défini par :

– **Un espace de recherche** S de dimension n où tout $s \in S$ représente une solution possible au problème P .

– **Une fonction objectif** f (ou plusieurs) appelée aussi fonction de fitness qui pour toute solution $s \in S$ affecte une valeur fs appelée valeur de fitness. La valeur de la fitness représente la qualité de la solution et définit un espace de solutions potentielles au problème. Une méthode d'optimisation recherche un ou plusieurs points dans l'espace de recherche qui satisfait au mieux l'objectif à atteindre. [1]

3. Méthodes de résolution d'un problème d'optimisation

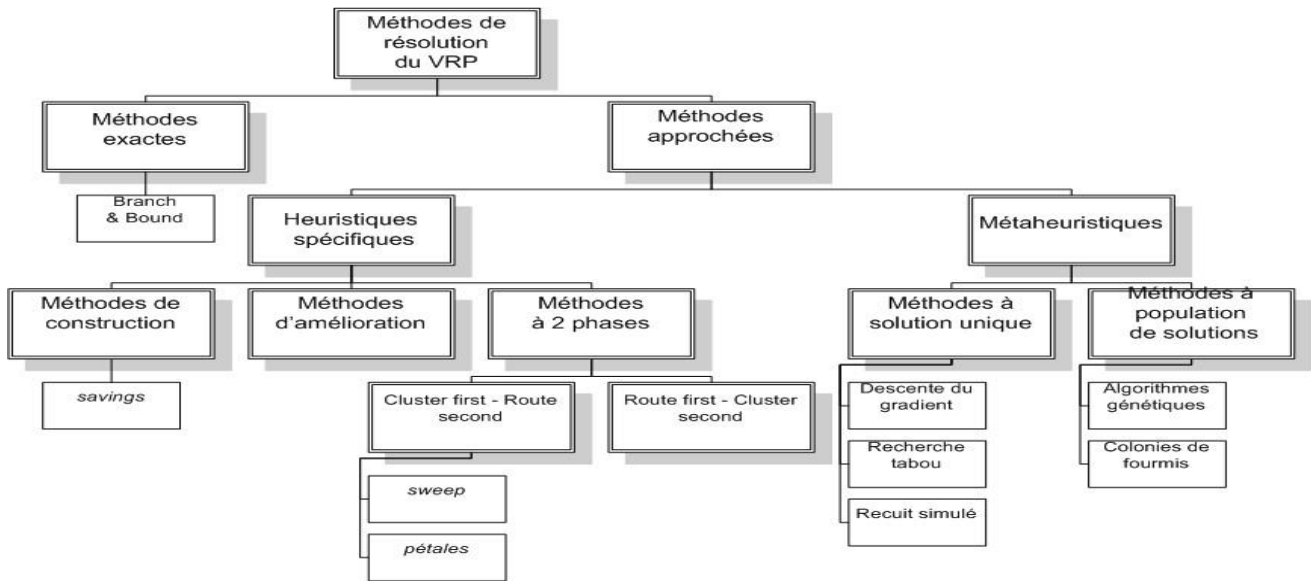


Figure 1.1 : Classification des méthodes d'optimisation.

3.1 Méthodes Exactes

Les méthodes de résolution exactes sont des méthodes qui garantissent l'obtention de solutions optimales pour des problèmes raisonnables (taille finie) au profit d'un temps de calcul réduit et de prouver son optimalité. Malgré les progrès réalisés notamment en termes de programmation linéaire entière (PLNE), le temps de calcul nécessaire pour trouver une solution est probable d'augmenter exponentiellement avec la taille du problème. Parmi les méthodes exactes on trouve les méthodes les plus traditionnelles (développées au cours des 30 dernières années) telles que la technique de séparation et évaluation (branch and bound) ou les algorithmes avec retour arrière (backtracking).

3.2 Méthodes Approchées

Les méthodes de résolution approchées exploitent généralement des processus aléatoires dans l'exploration de l'espace de recherche pour faire face à l'explosion combinatoire engendré par l'utilisation des méthodes exactes, elles sont adaptées à chaque problème traité ; mais d'autre part ne garantissent pas de trouver une solution exacte, mais seulement une approximation, donc l'inconvénient est de ne disposer en retour d'aucune information sur la qualité des solutions obtenues.

3.2.1 Heuristiques

Une heuristique est une méthode approchée spécifique que l'on applique à la résolution d'un problème sans garantir les performances, les trois principales classes d'heuristiques utilisées pour la Résolution des problèmes combinatoires sont :

3.2.1.1 Les heuristiques de constructions :

L'idée de base d'une heuristique constructive est de réduire la taille du problème à chaque étape pour limiter progressivement l'ensemble des solutions réalisables. Les heuristiques de construction élaborent graduellement la tournée en ajoutant une ville (nœud) à chaque étape. Elles s'arrêtent dès que la solution est trouvée et n'essayent pas de l'améliorer. Dans cette catégorie il y a l'heuristique du plus proche voisin, l'algorithme glouton. Les heuristiques d'amélioration consistent, une fois qu'une tournée est générée par une heuristique de construction à l'améliorer pour obtenir une tournée de qualité meilleure.

3.2.1.2 Heuristiques d'insertions(amélioration)

Les heuristiques d'insertion sont des heuristiques de construction très efficaces, l'heuristique d'insertion la plus connue dans la littérature est le plus proche voisin, a été développée en 1983, puis modifiée et améliorée par plusieurs chercheurs, elle est souvent utilisée dans le domaine du transport tel que le problème de tournées de véhicule(VRP), voyageur de commerce (TSP) et d'autre types de problème de transport selon les besoins spéciaux [7], son principe est de sélectionner à chaque itération le plus proche nœud de la tournée en construction, puis l'insère à l'emplacement du cycle qui minimise le cout de la tournée.

3.2.1.3 Heuristiques deux phases

- On distingue deux types d'heuristiques en deux-phases [8] :
Le groupe en premier et la route en second (cluster first - route second) : c'est de mettre des sous-ensembles de sommets voisins pour lesquels on construit ensuite une tournée et pour chaque sous ensemble on affecte un véhicule.
- La route en premier et le groupe en second (route first -cluster second) : le principe de l'heuristique est de construire une tournée comportant un grand nombre de clients et traite toutes les tâches en ignorant certaines contraintes afin d'obtenir des solutions acceptables pour le problème.

3.2.2 Les méta-heuristique

Les méta-heuristique sont très utilisées de nos jours. Une méta- heuristique est une approche polyvalente applicable et adaptable aux différents problèmes, et possède différents outils pour guider et améliorer la qualité des solutions, les méta-heuristique peuvent combiner différent heuristique.

L'objectif d'une méta-heuristique est de réussir à trouver un optimum global, l'idée est à la fois de parcourir l'espace de recherche, et d'explorer les zones qui semblent être prometteuse ; mais sans être « Piège » par un optimum local.

Les méta-heuristiques peuvent être classées en deux catégories : les méta-heuristiques à base de solution unique et les méta-heuristiques à base de population de solutions [9].

3.2.2.1 Méthodes à base d'une solution

Une méta-heuristique à base de solution unique manipule une seule solution au cours de la procédure de recherche, elle démarre la recherche avec une seule solution et tente d'améliorer sa qualité au cours des itérations. [10].

- **La méthode de Recuit Simulé**

Le recuit simulé a été introduit par (Kirkpatrick.1983) comme une méthode méta-heuristique de recherche locale normale, en utilisant une stratégie pour éviter les minima locaux [11].

Elle est basée sur une procédure qui a été utilisée depuis longtemps par des métallurgistes qui, pour obtenir un alliage sans défaut, chauffent d'abord leur pièce de métal avant de refroidir très lentement l'alliage (technique de recuit). Pour simuler cette évolution d'un système physique vers son équilibre thermodynamique à une température T, la méthode du recuit simulé exploite l'algorithme de Métropolis [12].

Dans ce dernier on parle d'une configuration donnée, et on fait subir au système une modification élémentaire. Si cette perturbation a pour effet de diminuer la fonction objectif (ou énergie) du système, elle est acceptée. Sinon, elle est acceptée avec une probabilité. En appliquant itérativement cette règle, on engendre une séquence de configurations qui tendent vers l'équilibre thermodynamique.

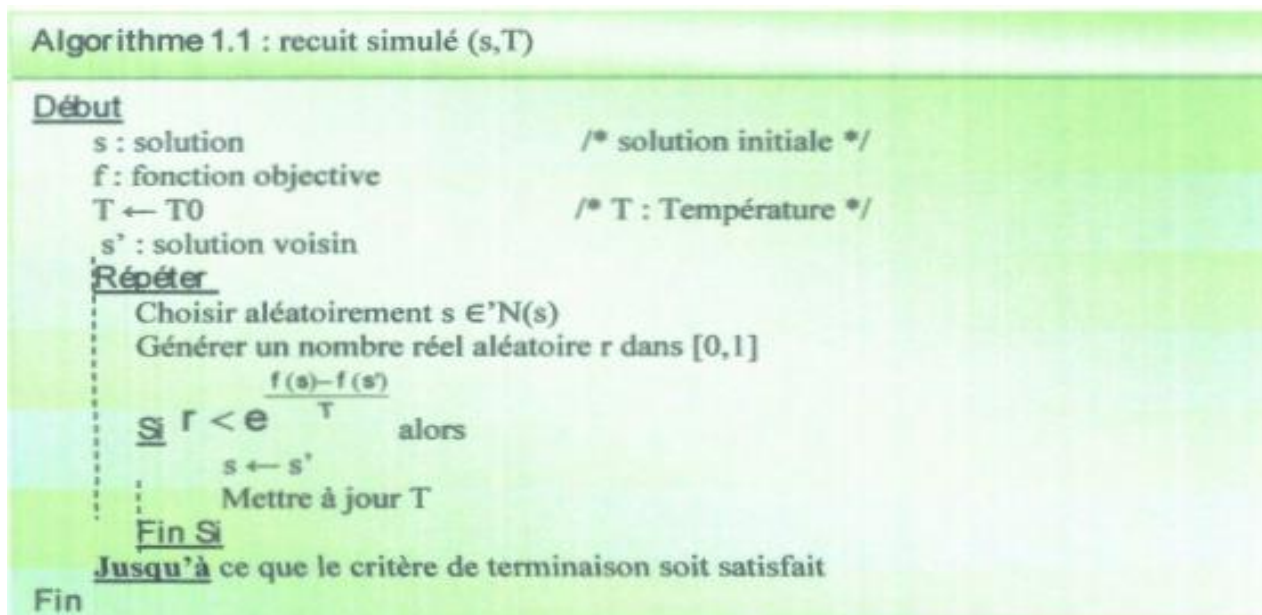


Figure 1.2 : la méthode de recuit simulé.

L'inconvénient est qu'une fois l'algorithme de recuit simulé est piégé à une baisse de température dans un minimum local, il lui est impossible de s'échapper par lui-même. De nombreux essais sont nécessaires pour trouver le bon paramètre pour définir les voisinages permettant un calcul effectif.

L'avantage du recuit simulé est le fait d'être souple vis-à-vis des évolutions du problème et facile à implémenter d'excellents résultats pour un nombre de problèmes complexes.

- **La méthode de la Descente**

Pour un problème de minimisation d'une fonction f , la méthode peut être décrite dans l'algorithme 1.2 de descente simple :

```
Algorithme 1.2 : Descente_simple (s)

Début
  s : solution          /* solution initiale */
  f : fonction objective
  s' : solution voisin
  Répéter
    Choisir s' dans un voisinage N(s) de s ;
    Si  $f(s') < f(s)$  alors
       $s \leftarrow s'$  ;
  Jusqu'à ce que  $f(s') \geq f(s), \forall s' \in N(s)$ .
Fin
```

Figure 1.3 : la méthode de descente simple.

L'inconvénient majeur de la méthode de descente est son arrêt au premier minimum local rencontré. Pour améliorer les résultats, on peut démarrer plusieurs fois l'algorithme en partant d'un jeu de solutions initiales différentes, mais la performance de cette technique décroît rapidement. Pour éviter d'être bloqué au premier minimum local rencontré, on peut décider d'accepter, sous certaines conditions, de se déplacer d'une solution s vers une solution $s' \in N(s)$ telle que $f(s') > f(s)$.

- **La méthode de Recherche Taboue**

La recherche taboue (TS) est une méthode de recherche locale combinée avec un ensemble de techniques permettant d'éviter d'être piégé dans un minimum local ou la répétition d'un cycle. La recherche taboue est introduite principalement par Glover (Glover 1986) [3],

Cette méthode devenue très classique en optimisation combinatoire, elle distingue des méthodes de recherche locales simples en utilisant un historique des solutions visitées, afin de rendre la recherche un peu moins « aveugle ». Il devient possible de sortir d'un minimum local, mais, pour éviter d'y retomber périodiquement, certaines solutions sont bannies, elles sont rendues « taboues ». A l'inverse du recuit simulé qui génère de manière aléatoire une seule solution voisine $s' \in N(s)$ à chaque itération, Taboue examine un échantillonnage de solutions de $N(s)$ et retient la

meilleure s' même si $f(s') > f(s)$. La recherche Taboue ne s'arrête donc pas au premier optimum trouvé.

Le grand problème alors devient immédiatement à s , puisque s est meilleure que s' , pour éviter de tourner en rond, on crée une liste T qui mémorise les dernières solutions visitées et qui interdit tout déplacement vers une solution de cette liste. Cette liste Test appelée liste Taboue [3].

Le pseudo-code de l'algorithme 1.3 représente la méthode recherche taboue :

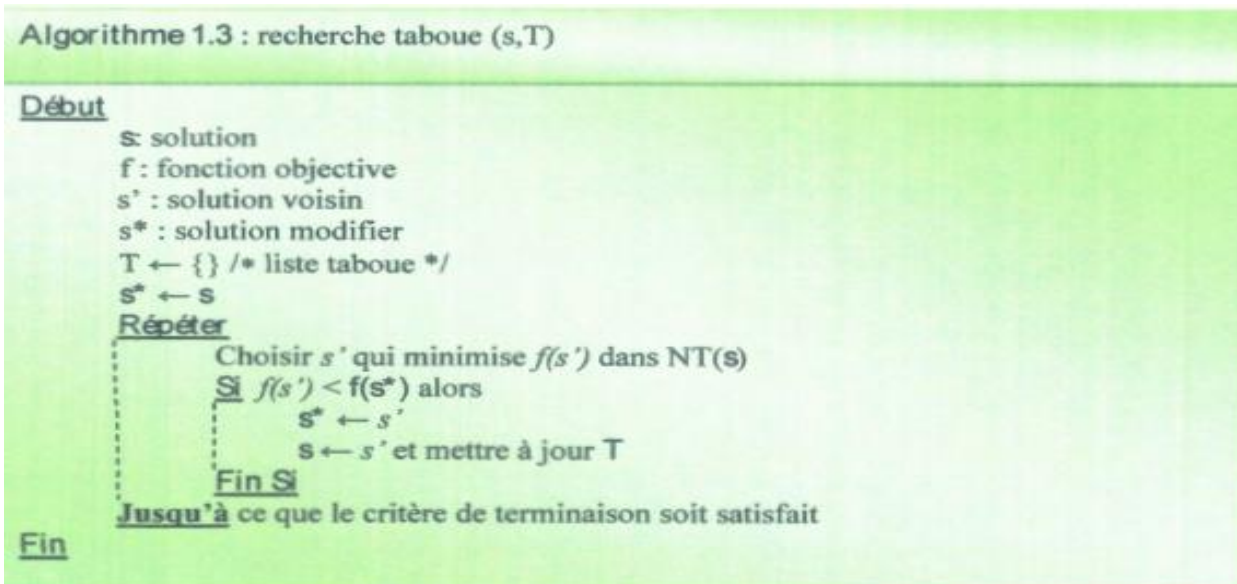


Figure 1.4 : la méthode de la recherche taboue.

La méthode taboue exige une gestion de la mémoire de plus en plus lourde en mettant des stratégies de mémorisation complexe. L'efficacité de la méthode taboue offre son utilisation dans plusieurs problèmes d'optimisation combinatoire classiques tels qu'a été testée avec succès sur les grands problèmes classiques et elle est fréquemment appliquée sur le problème de voyageur de commerce (TSP), le problème d'ordonnancement, les problèmes de constitution de planning, le problème de tournées de véhicules (VRP), l'exploration géologique, le routage etc.

- **La méthode GRASP**

Le GRASP (Greedy Randomized Adaptive Search Procedure) crée une nouvelle solution à chaque itération, le nombre des itérations est donnés au début, on effectue la construction d'une solution suivant une heuristique semi-glouton (Randomized Greedy), puis on applique une recherche locale sur le voisinage de cette solution afin d'obtenir un optimum local, la solution finale sera la meilleure solution obtenue parmi toutes les solutions trouvées. Le GRASP est une métaheuristique d'optimisation combinatoire introduite pour la première fois dans l'article Feo et Resende (1989) [15] [16], basée sur deux phases :

- 1- Phase de la construction d'une solution par une heuristique glouton aléatoire ;
- 2- La phase d'amélioration utilise une autre métaheuristique de recherche local (ex : recuit simulé).

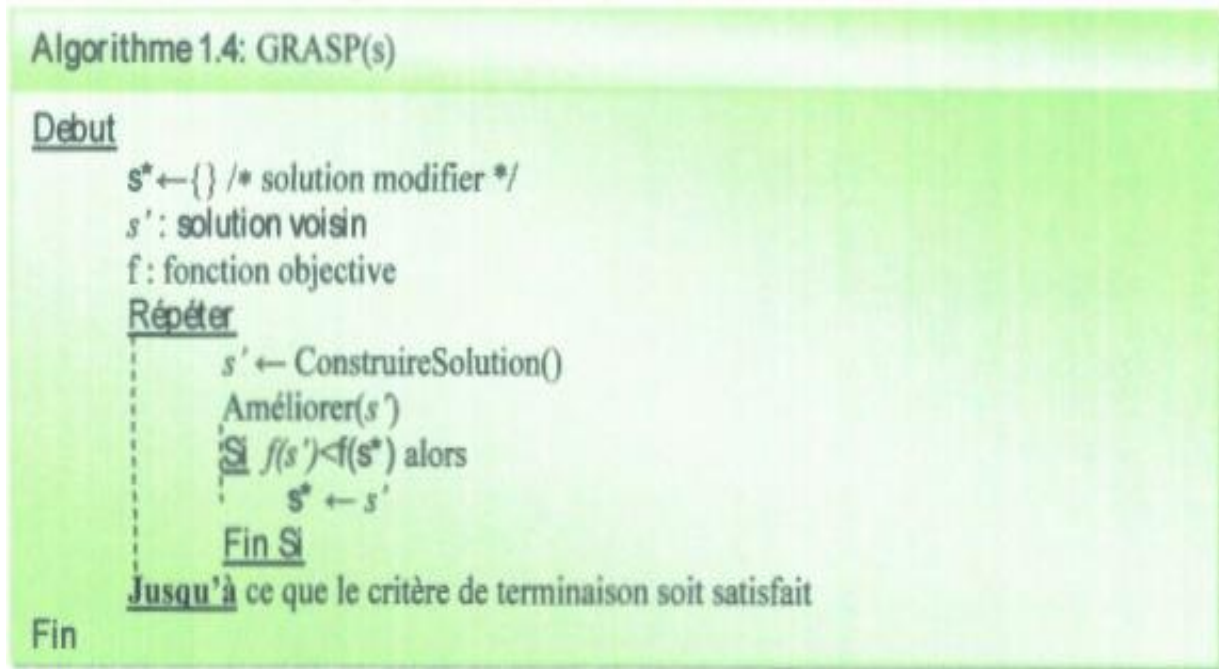


Figure 1.5 : la méthode GRASP

3.2.2.2. Méthodes à base de populations

Un méta heuristique à base de population de solutions manipule un ensemble de solutions en parallèle. La recherche est lancée avec une population de solutions et essaie d'améliorer leurs qualités au cours des itérations afin de fournir les meilleures solutions trouvées.

- **Les algorithmes génétiques**

Les algorithmes génétiques dérivent de la famille des algorithmes évolutifs. Ils ont été développés par J. Rolland en 1975 comme outils de modélisation de l'adaptation travaillant dans un espace de chaînes de bits. Les plus populaires aux chercheurs de différentes disciplines. Largement utilisés et développés par D.E. Goldberg en 1989 [23]. A partir des années 1990 la Programmation Génétique (Genetic Programming - GP) a été développée par J. Koza dans le but d'attendre un des rêves des programmeurs : écrire un programme automatiquement. La représentation est basée sur des arbres représentant les programmes, les algorithmes génétiques s'appuient sur des techniques dérivées de l'évolution naturelle : croisement, sélection, mutation, etc.

Les algorithmes génétiques sont coûteux en temps de calcul, puisqu'ils manipulent plusieurs solutions simultanément, et qu'un bon individu se met en l'espace de quelques générations, à conquérir toute la population. Le grand avantage des algorithmes génétiques est qu'ils parviennent à trouver de bonnes solutions sur des problèmes très complexes.

- **Les algorithmes mimétiques**

Les algorithmes mimétiques sont une hybridation entre les algorithmes de recherche locale et les algorithmes génétiques. Le principe général est le même que pour les algorithmes génétiques mis à part qu'un opérateur de recherche locale est ajouté après celui de mutation. La partie génétique de ces algorithmes peut être vue comme une forte diversification alors que la partie recherche locale correspondrait à une forte intensification (accompagnée d'une faible diversification).

- **L'algorithme de colonies de fourmis**

L'algorithme de colonies de fourmis est l'un des algorithmes basés sur l'intelligence par troupe. Il a été introduit au début des années 90 par le trinôme Colorni, Dorigo et Maniezzo [24]. L'idée de base imite le comportement collectif des fourmis lorsqu'elles se déplacent entre la fourmilière et la source de nourriture, l'objectif du comportement collectif des fourmis est de ramasser les aliments sans perdre le chemin de leur nid.

Les fourmis communiquent indirectement entre elles, provoquant des changements dans leur environnement. Au début de la recherche, les fourmis se propagent aléatoirement en prenant des chemins de différentes tailles (court, long,) dont elles déposent sur le sol une matière odorante appelée « phéromone » d'intensités égales. Afin d'attirer l'attention de leurs congénères en retournant au nid, les fourmis déposent des phéromones un peu différentes contenant un message concernant la qualité du site visité.

Elles ont tendance à suivre le chemin d'une intensité plus élevée des phéromones, plus le trajet est court, plus la quantité de phéromones déposée trop élevée, et plus l'intensité de phéromones est grande, plus le nombre de fourmis utilisant ce passage augmente. Par conséquent, le chemin le plus long sera abandonné car l'intensité de phéromones y compris est petite et s'évapore rapidement [25].

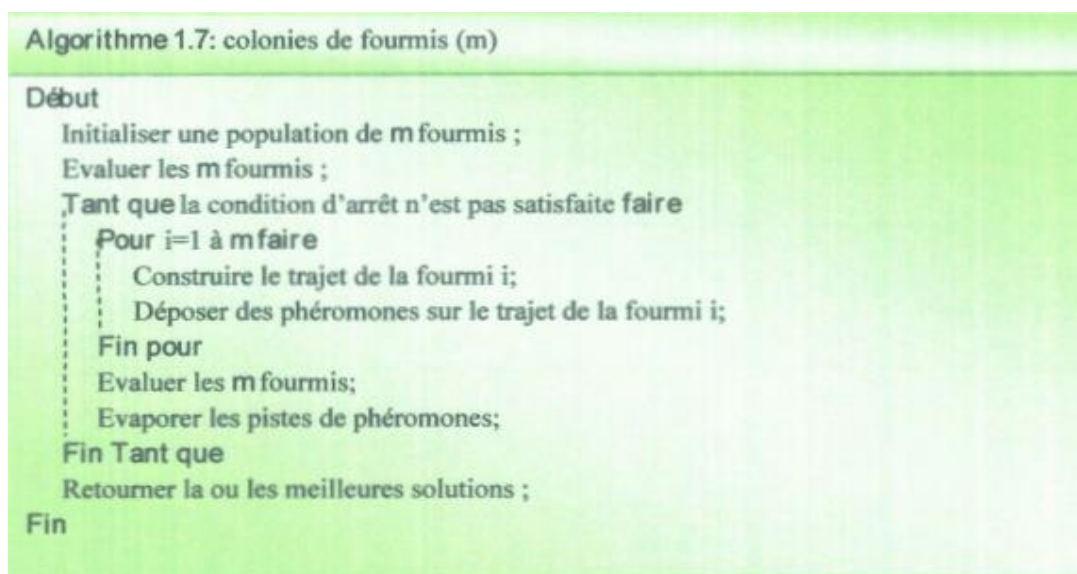


Figure 1.6 : la méthode de colonie de fourmis.

L'algorithme de colonies de fourmi (Ant Colony- AC) : algorithme à colonie de fourmis, se parallélise de façon très naturelle, en affectant par exemple un processus pour traiter la marche de chaque fourmi, et un autre pour mettre à jour les pistes de phéromones.

Ce dernier offre finalement beaucoup de souplesse, et il a été possible de l'adapter à la majorité des problèmes combinatoires classiques, l'algorithme de par son dynamisme intrinsèque, s'adapte aussi très bien aux espaces de solutions qui varient dynamiquement dans le temps.

3.3 Stratégies recherche

Un certain nombre de propriétés sont trouvées dans toutes les méta-heuristiques et jouent un rôle prépondérant dans leur efficacité dans le traitement d'un problème donné.

3.3.1 Intensification et Diversification

- L'intensification se réfère à l'exploitation des informations recueillies par le système à un moment donné. C'est ce qui permet d'améliorer la valeur d'une solution trouvée dans un certain voisinage.
- La diversification concerne l'exploitation de tout l'espace de recherche en passant d'une zone à une autre.

Les notions d'intensification et de diversification sont prépondérantes dans la conception des méta-heuristiques, qui doivent atteindre un équilibre délicat entre ces deux dynamiques de recherche.

Les deux notions ne sont donc pas contradictoires mais complémentaires [26]. Il existe de nombreuses stratégies qui combinent les deux aspects.

3.3.2 Hybridation des Méthodes

L'hybridation est une tendance observée dans de nombreux travaux réalisés sur les méta-heuristiques durant ces dix dernières années. Elle permet de profiter des avantages cumulatifs des diverses méta-heuristiques, à tel point que les méta-heuristiques que nous avons vues jusqu'ici ne sont plus que des voiles des points de départ, pour commencer à résoudre un problème d'optimisation.

3.3.3 Choix d'une structure de voisinage

Le choix du voisinage en optimisation combinatoire a une grande influence sur le résultat de l'application d'une méta-heuristique, est à l'origine de la méthode de voisinage variable à une solution est un sous ensemble des solutions, dont les membres sont des solutions proches (voisines)

de cette solution. Alors, on dit qu'une solution S'est une voisine d'une solution S, si elle peut être obtenue en modifiant légèrement cette solution [27].

4. Classification des Méta-heuristiques

4.1 Approche trajectoire

Il est parfois constaté que la classification des méta-heuristiques est de distinguer ceux qui travaillent avec une population des solutions de ceux qui manipulent une seule solution à la fois, les méthodes qui tentent itérativement d'améliorer une solution sont appelées méthodes de recherche locale ou méthodes à trajectoire, la méthode Taboue, le Recuit Simulé et la Recherche à Voisins Variables sont des exemples typiques des méthodes à trajectoire, ces méthodes construisent une trajectoire dans l'espace des solutions en essayant de progresser vers des solutions optimales, l'exemple le plus connu de la méthode qui fonctionne avec une population de solutions est l'algorithme génétique [28].

4.2 Approche phénomène naturel

La différence entre les méta-heuristiques qui s'inspirent des phénomènes naturels et ceux qui ne s'y inspirent pas, comme exemple, les algorithmes génétiques et les algorithmes des fourmis s'inspirent respectivement de la théorie de l'évolution et du comportement de fourmis à la recherche de nourriture. D'autre part, la méthode de Taboue ne semble pas avoir été inspirée par un phénomène naturel. Cependant, une telle classification ne semble pas très utile et parfois difficile à réaliser.

En effet, il y a beaucoup de méta-heuristiques récentes qu'il est difficile de classer dans l'une des deux catégories, Certains se demanderont par exemple si l'utilisation d'une mémoire dans la méthode Taboue n'est pas directement inspirée de la nature.

4.3 Approche mémoire :

Certains méta-heuristiques utilisent l'historique de recherche lors de l'optimisation, tandis que d'autres ne se souviennent pas du passé, les algorithmes sans mémoire sont en fait des processus de Markov puisque l'action à exécuter est totalement déterminée par la situation actuelle, Les méta-heuristiques qui font usage de l'historique de la recherche peuvent le faire de différentes façons.

Les méthodes avec mémoire à court terme sont généralement différenciées de celles à mémoire à long terme, le meilleur représentant de la méta-heuristique avec mémoire reste la recherche Taboue. D'autre part, pour ce qui est des « sans mémoire », on trouve le recuit simulé comme exemple.

4.4 Approche voisinage

La plupart des méta-heuristiques travaillent sur une seule structure de voisinage. En d'autre terme, la topologie de paysage de fitness ne change pas en cours de l'algorithme. Autre méta-heuristiques telles que la recherche par voisinage variable utilise un ensemble des structures de voisinage qui donne la possibilité de diversifier la recherche en échange entre les structures de voisinage qui ont des formes différentes [29].

4.5 Approche selon la fonction objective

Les méta-heuristiques peuvent également être classées selon leur utilisation de la fonction objective, par exemple , un problème d'optimisation qui cherche à minimiser une fonction f sur un espace S de solutions, certaines méta-heuristiques qui s'appellent statiques travaillent directement sur f alors que d'autres, dites dynamiques, font usage d'une fonction g obtenue à partir de f en ajoutant quelques composantes qui permettent de modifier la topologie de l'espace des solutions \$ ces composantes additionnelles pouvant varier durant le processus de recherche [28]

5. Conclusion :

Dans ce chapitre nous avons présenté les différentes méthodes d'optimisation, tout d'abord nous avons commencé par des méthodes exactes et ses variantes, ensuite nous avons donné une vue globale sur les approches heuristiques, puis nous avons focalisé sur les approches des méta-heuristiques.

Chapitre 02 :

Etat de l'art : Le Problème de Tournées de Véhicules et le CLRP

1. Introduction :

Les problèmes de tournées de véhicules sont largement considérés dans la littérature pour leurs intérêts applicatifs. Par ailleurs, des publications sont régulièrement proposées que ce soit sur la problématique initiale ou sur des variantes du problème. Dans ce chapitre nous présentons un état de l'art sur les problèmes de tournées de véhicules et ses variantes et nous expliquons en détail le problème étudié dans ce mémoire (CLRP).

2. Le problème de VRP

2.1 Définition

Le problème de tournées de véhicules (VRP) consiste en la recherche des meilleures itinéraires pour une flotte de véhicules partant d'un dépôt afin de visiter un ensemble de clients. L'évaluation de ces itinéraires se fait selon une fonction objectif prédéfinie, telle que le temps de parcours, une distance ou un coût global à minimiser. Ce problème est une extension classique du problème de voyageur de commerce, et tout comme ce dernier, il appartient à la classe des problèmes d'optimisation NP-Complets [34]

Les véhicules doivent commencer et finir leurs tournées au dépôt. L'objectif est donc de visiter chaque client exactement une fois tout en vérifiant les contraintes qui lui sont propres telles qu'une demande à satisfaire (livraison de marchandises) ou un service à effectuer (temps nécessaire). Un ensemble de visites faites par un véhicule est une tournée. 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. [85]

2.2 VRP de base

Au-delà de sa définition globale, le VRP possède plusieurs déclinaisons inspirées des différentes problématiques réelles. Nous définissons dans ce qui suit le cas classique du VRP puis présentons dans la section suivante brièvement d'autres variantes.

Il est commun dans la littérature de nommer "VRP" le cas du problème de tournées de véhicules avec contraintes de capacité ou Capacitated VRP (CVRP) qui modélise la problématique de livraison de marchandises. Chaque client ayant une demande qui est la quantité de marchandise à lui livrer. Les véhicules peuvent transporter jusqu'à une quantité maximale de marchandises. En plus des autres contraintes, il est nécessaire de vérifier qu'un véhicule puisse respecter les

demandes de tous les clients sur sa tournée et donc que la somme des demandes sur sa tournée ne dépasse pas sa capacité. [34]

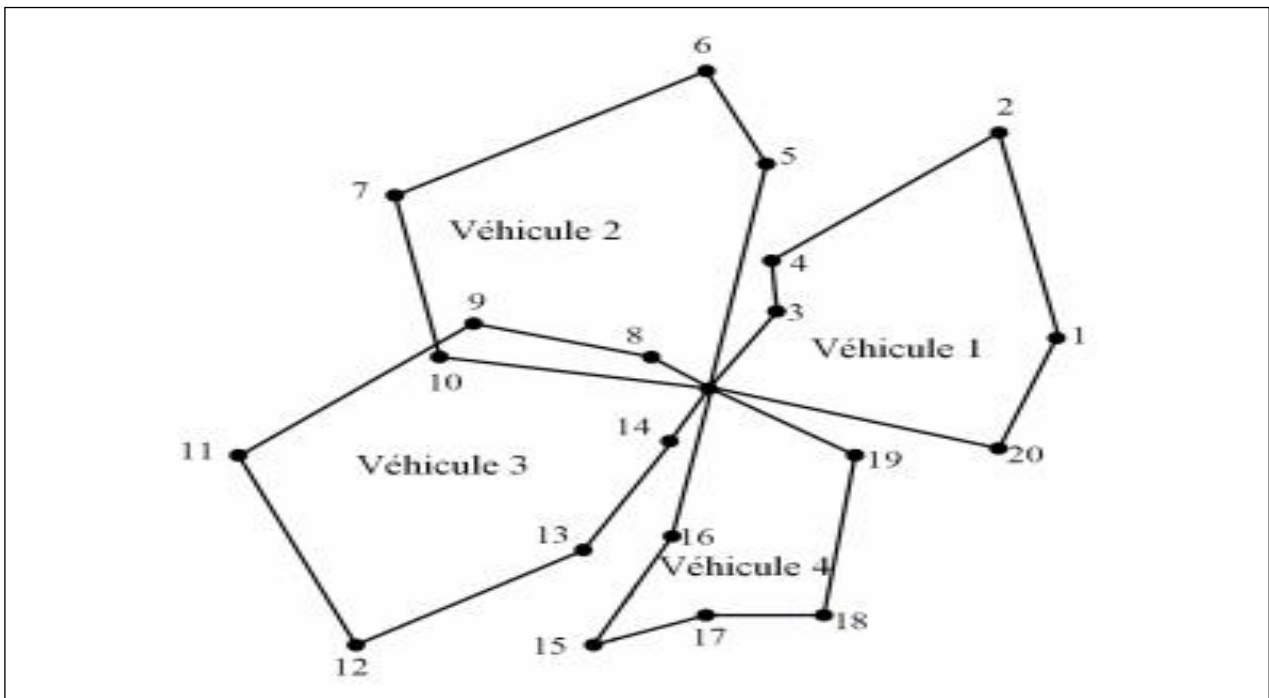


Figure 2.1 : le problème de tournée de véhicule VRP

2.3 Modélisation du VRP

Nous modélisons le problème de VRP un graphe orienté $G = (C \cup \{0\}, A)$. $\{0\}$ est le dépôt où un ensemble de m $V = \{1, \dots, m\}$ véhicule est disponible, chaque véhicule ayant une capacité maximale Q . Chaque client $i \in C$ a une demande $0 \leq q_i \leq Q$. Le poids c_{ij} de chaque arc $(i, j) \in A$ correspond à son coût. Le modèle est comme suit [34] :

$$\text{La variable de décision } x_{ijk} = \begin{cases} 1 & \text{Si le véhicule } k \text{ parcourt l'arc } (i, j) \\ 0 & \text{Sinon} \end{cases}$$

La fonction objective étant de minimiser le coût total engendré par les routes, le programme linéaire s'écrit comme suit :

$$\begin{aligned}
 \min_{x_{ijk}} \quad & \sum_{(i,j) \in A} c_{ij} \sum_{k=1}^m x_{ijk} & (2.1a) \\
 \text{s.t.} \quad & \sum_{i=0}^n \sum_{k=1}^m x_{ijk} = 1 & \forall j \in C & (2.1b) \\
 & \sum_{i=0}^n x_{ipk} = \sum_{j=0}^n x_{pjk} & \forall p \in C, \forall k \in V & (2.1c) \\
 & \sum_{j=1}^n x_{0jk} \leq 1 & \forall k \in V & (2.1d) \\
 & \sum_{j=1}^n x_{0jk} = \sum_{i=1}^n x_{i0k} & \forall k \in V & (2.1e) \\
 & \sum_{i=1}^n q_i \times \sum_{j=0}^n x_{ijk} \leq Q & \forall k \in V & (2.1f)
 \end{aligned}$$

Figure 2.2 : Modélisation du VRP

Les contraintes **2.1b** imposent que chaque client est visité exactement une fois. Les contraintes de conservation de flot **2.1c** imposent que le même véhicule qui a visité un client, "reparte" de chez ce client. Les contraintes **2.1d** et **2.1e** assurent que si un véhicule fait une tournée, elle sera unique et commencera et finira au dépôt. Les contraintes **2.1f** assurent que le véhicule a assez de capacité pour satisfaire la demande de chacun des clients sur sa tournée. [34]

2.4 Variantes du problème VRP

De multiples applications du VRP ont généré diverses variantes de ce problème et la littérature sur ces variantes est immense. Dans cette section nous présentons premièrement quelques variantes et dans un deuxième temps nous détaillerons plus particulièrement la variante multi dépôts du VRP qui est la variante du VRP qui se rapproche le plus de notre problématique. [36]

2.4.1 Problèmes de tournées de véhicules avec fenêtres de temps (VRPTW)

Dans cette variante le client définit une fenêtre de temps (intervalle) dans laquelle il peut être visité par le véhicule. Si le véhicule arrive en dehors de la fenêtre de temps, il existe deux formulations : La première dite à fenêtres de temps molles (soft time windows constraints), où le client est servi mais une pénalité est ajoutée à la fonction objective. La deuxième formulation dite à fenêtres de temps dures (hard time window constraints), où le véhicule attend le début de l'intervalle s'il arrive plus tôt, et ne pourra pas servir le client s'il arrive plus tard. Un des premiers travaux relatifs au VRPTW date de 1967, les auteurs traitent un cas pratique d'un VRP avec fenêtres de temps sans le nommer ou le modéliser. Dans, Solomon propose des instances du VRPTW à 100 clients qui sont devenues des benchmarks reconnus pour la

Problématique. L'une des variantes de ce problème est celle où chaque client a plusieurs fenêtres de temps et nécessite une seule visite lors de l'une d'elles.[37]

2.4.2 Le problème de tournées de véhicules avec livraison et collecte (VRPB)

Les clients peuvent demander une livraison de marchandises ou une collecte de marchandises. On retrouve donc deux types de clients, des clients nécessitant une livraison de marchandises depuis le dépôt et ceux nécessitant une collecte de marchandises vers le dépôt. Chaque véhicule partant du dépôt doit premièrement livrer tous les clients de sa tournée puis collecter les marchandises des clients pour les ramener au dépôt [36].

2.4.3 Le problème de tournées de véhicules avec ramassage et livraison (VRPSD)

Similaire à la variante précédente, on retrouve aussi deux types de clients, sauf que la livraison/collecte se fait entre les clients, le(s) dépôt(s) n'étant qu'un point de départ et de fin pour les véhicules. Les clients sont donc organisés en paires (collecte, livraison) et l'objectif du "VRP with Pickup and Delivery" est de livrer chaque client "livraison" avec les produits récupérés chez le client "collecte" correspondant. Il existe diverses variantes à cette problématique, telles que celle où on considère que le produit est le même chez tous les clients, il n'est donc pas obligatoire de respecter les paires (collecte, livraison), [36]

2.4.4. Le problème de tournées de véhicules Multi-trips (VRPMT)

Cette variante permet aux véhicules de repasser par le dépôt pour se ravitailler en marchandises, le véhicule peut donc visiter plus de clients en une journée que ne lui permet sa capacité lors d'une tournée simple. Elle permet donc d'utiliser moins de véhicules mais elle n'est adaptée que dans le cas où il est relativement facile en termes de distance et de temps de repasser par le dépôt [37].

2.4.5 Le problème de tournées de service (TRP)

A l'opposé des livraisons de marchandises, les problématiques de services comme les tournées de soins médicaux ou les services à domicile (dépannage, réparation...), sont modélisés tels que chaque client nécessite un service pouvant être réalisé par la totalité ou un sous-ensemble des techniciens (véhicules) et dont la durée est fixe ou dépendante du technicien affecté. Les techniciens n'ont pas les mêmes compétences ou/et à des niveaux différents. Dans les auteurs présentent le Technicien Routing and Scheduling Problem, où chaque technicien a un ensemble de compétences et peut prendre des ressources renouvelables (outils) et non renouvelables (pièce de rechanges) au dépôt, chaque client a une tâche nécessitant une compétence spécifique. L'objectif étant que chaque technicien puisse effectuer les tâches des clients de sa tournée. Dans [36] les auteurs considèrent qu'un véhicule transporte une équipe puisque les tâches des clients peuvent nécessiter plusieurs

techniciens, la solution est donc de constituer les équipes et les tournées satisfaisant l'ensemble des clients. Dans [37], les techniciens ont les mêmes compétences mais à différents niveaux d'expériences, les clients ont plusieurs tâches dans le cadre d'une entreprise de services à domicile, la durée des tâches dépend de l'expérience du technicien sur chacune. Ces problématiques sont variées, donc difficiles à englober dans une seule modélisation. De plus elles sont plus proches des problématiques d'affectation ce qui explique le peu de littérature dans le domaine du VRP et du routing en général. La littérature sur les problématiques de tournées regorge d'autres variantes du VRP que nous ne pouvons pas citer ici vu leurs nombres et leurs spécificités plus ou moins complexes. En plus de ces diverses variantes, il est très commun de trouver des travaux sur des problématiques mixant plusieurs variantes telles que le Rich VRP dans qui combine le Multi-dépôt VRP avec le VRP With Time Windows et y rajoute différentes autres contraintes et caractéristiques. Plusieurs Survey ont été élaborés au fil des années pour synthétiser les différents travaux sur les problèmes de tournées, nous citerons ici trois livres Survey traitant du VRP et de ses diverses variantes. [36]

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

Ce problème considère plusieurs dépôts au lieu d'un, dans chacun d'entre eux est affectée une partie de la flotte de véhicules. Chaque client est caractérisé par un sous-ensemble de dépôts, il ne doit être visité qu'une fois à partir de l'un d'eux. Les véhicules doivent commencer et finir leur tournée au dépôt où ils sont affectés et ne peuvent pas visiter les autres dépôts. Le problème CLRP et une sous variante du problème de MDVRP objet de la section suivante.[36]

3. Le problème de CLRP

3.1 Présentation du problème

Si un dépôt ne répond pas à la capacité demandée de l'ensemble des clients, alors il est obligatoire d'ouvrir plusieurs, on a donc un problème d'affectation des clients pour quel dépôt serait servi ? sans oublier le problème d'élaboration des tournées. On aura alors le problème de tournées multi-dépôts, bien que supposant que ces dépôts ne sont pas encore cout d'ouverture, le problème devient alors un problème de localisation et de routage (LRP- Location-Routing Problem -LRP) [30].

La particularité de CLRP (Capacitated Location-Routing Problem) se trouve dans les capacités imposées aux dépôts et aux véhicules, alors on trouve que tous les sites possèdent des positions avec une capacité et un coût d'ouverture, chaque client possède une position et une demande, et on trouve aussi une flotte illimitée des véhicules contenant une capacité fixe et homogène. L'objectif est de sélectionner un sous-ensemble de sites qui couvriront tous les clients, chaque véhicule sert un sous-ensemble de clients de façon que chaque client soit visité une seule fois avec un coût minimal tout en respectant les capacités des sites et véhicules

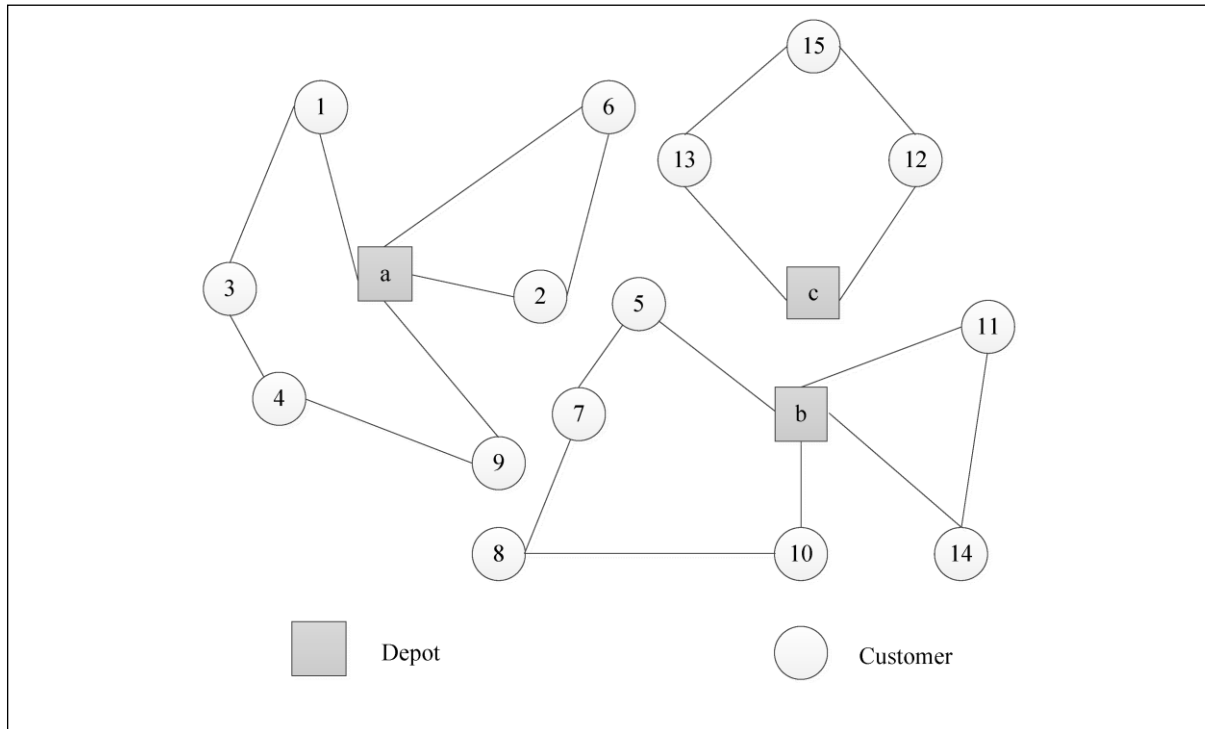


Figure 2.3 : problème de CLRP.

3.2 Description mathématique

3.2.1 Notation

- N : Le nombre de villes (clients).
- M : Le nombre de dépôts.
- $C = \{ C_0 \dots C_{n-1} \}$: l'ensemble des villes (clients).
- $D = \{ D_0 \dots D_{m-1} \}$: l'ensemble des dépôts.
- Q : la capacité de véhicule.
- W_i : la capacité de dépôt i .
- C_{ij} : le cout de transport entre ville i et j .
- O_i : le cout d'ouverture de dépôt i .
- C_{uv} : le cout d'utilisation de véhicule
- d_j : demande de client j .

3.2.2 Modélisation

Le problème de Localisation et Routage des véhicules avec capacité CLRP est défini sur un graphe $G = (D, E, C)$ non-orienté, D est un ensemble de nœuds, I est un ensemble de m Dépôts, $J = D/I$ est l'ensemble de n clients. E est l'ensemble d'arêtes reliant chaque paire de nœuds dans D , chaque client j a une demande d_j qui est connue, un ensemble de véhicules K ayant une capacité limitée Q . L'utilisation de chaque véhicule a un coût fixe C_{uv} . Enfin, chaque dépôt i possède une capacité limitée W_i et un coût d'ouverture O_i . L'objectif de ce problème est d'ouvrir un ensemble de dépôts

couvrant toutes les demandes de clients avec un coût total minimum (le coût total c'est la somme des coûts d'ouverture des dépôts, les coûts de routage et les coûts d'utilisation des véhicules). Le but est alors de déterminer quels dépôts ouvrir (combien et sur quels nœuds du graphe), et quelles tournées construire (combien, associées à quels dépôts, et composées de quelles chaînes de clients) de manière à respecter les contraintes énumérées ci-dessus et à minimiser le coût total, comprenant les coûts d'ouverture des dépôts, les coûts fixes d'utilisation de véhicules et la somme des coûts C_{ij} (*la fonction de cout*) des arêtes traversées par les véhicules [30].

L'objectif alors est de Minimiser Z :

$$\text{Min } z = \sum_{i \in I} O_i * y_i + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} F * x_{ij}^k + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ij} x_{ij}^k \quad (2.15)$$

Sous les contraintes :

$$\sum_{k \in K} \sum_{a \in \delta^-(i)} x_a^k = 1 \quad \forall j \in J, \quad (2.16)$$

$$\sum_{j \in J} \sum_{a \in \delta^-(j)} d_j x_a^k \leq Q \quad \forall k \in K, \quad (2.17)$$

$$\sum_{a \in \delta^+(i)} x_a^k - \sum_{a \in \delta^-(i)} x_a^k = 0 \quad \forall k \in K, \quad \forall i \in I, \quad (2.18)$$

$$\sum_{i \in I} \sum_{j \in J} x_{ij}^k \leq 1 \quad \forall k \in K, \quad (2.19)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij}^k \leq |S| - 1 \quad \forall k \in K, \quad (2.20)$$

$$\sum_{a \in \delta^+(j) \cap \delta^-(j)} x_a^k + \sum_{a \in \delta^-(j)} x_a^k \leq 1 + f_{ij} \quad \forall i \in I, \quad \forall j \in J, \quad \forall k \in K, \quad (2.21)$$

$$\sum_{j \in J} d_j * f_{ij} \leq W_i * y_i \quad \forall i \in I, \quad (2.22)$$

$$x_a^k \in \{0,1\} \quad \forall a \in A, \quad \forall k \in K, \quad (2.23)$$

$$y_i \in \{0,1\} \quad \forall i \in I, \quad (2.24)$$

$$f_{ij} \in \{0,1\} \quad \forall i \in I, \forall j \in J, \quad (2.25)$$

Figure 2.5 : Modélisation du problème de Localisation et Routage des véhicules

La fonction-objectif (2.15) minimise les coûts fixes d'ouverture des dépôts, les coûts d'utilisation de véhicules et les coûts des tournées. La Contrainte (2.16) assure que chaque client appartient à une seule tournée et n'a qu'un prédécesseur dans sa tournée. Les contraintes (2.17) jusqu'à (2.22) sont des contraintes qui imposent le respect des capacités des véhicules dans la tournée et la capacité de dépôt. La contrainte (2.17) indique que la quantité des demandes ne dépasse pas la capacité du véhicule. La contrainte (2.18) permet d'assurer que le coût d'un arc est calculé une seule fois. La contrainte (2.19) impose qu'une tournée appartient à un seul dépôt. La contrainte (2.20) pour éviter les sous-cycles. La contrainte (2.21) précise que le client peut être affecté à un dépôt si et seulement si la tournée qui les relie est ouverte. Les contraintes (2.23), (2.24) et (2.25) sont les contraintes d'intégrité des variables booléennes.

Notons que :

- La Variable booléenne x_{ij}^k égale 1 si l'arc i, j est visité par le véhicule k , 0 si non.
- La Variable booléenne Y_i égale 1 si le dépôt i est ouvert, 0 si non.
- La variable f_{ij} égale 1 si le client j affecté au dépôt i , 0 si non.

4. Méthodes proposées pour le CLRP

Le problème de CLRP est traité dans plusieurs articles durant plusieurs années, chaque article propose des techniques pour optimiser et accélérer la résolution de ce problème en se basant sur les différents principes et branches du domaine d'optimisation. Cette section est consacrée à la présentation de quelques méthodes proposées par les chercheurs du domaine pour traiter le problème de CRLP.

4.1 Méthode 1 : un algorithme mimétique pour le CLRP

4.1.1 Principes

Cet algorithme est proposé par L. KECHMANE et al [30] dans un article intitulé « *A Memetic Algorithm for the Capacitated Location Routing Problem* ». La méthode de résolution proposée dans cet article est basée sur les principes des algorithmes génétique combinés avec la performance de la recherche locale afin d'assurer une meilleure investigation de l'espace de recherche.

La méthode est basée sur un codage simple d'un chromosome qui contient à la fois des dépôts et des clients :

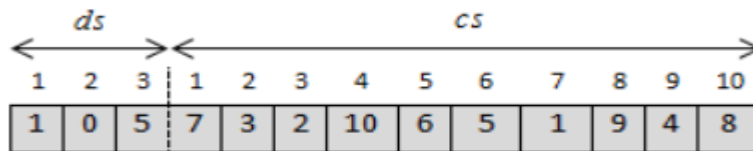


Figure 2.6 : codage de la solution methode1.

Le croisement des solutions est réalisé d’une manière simple via le choix des points de coupures.

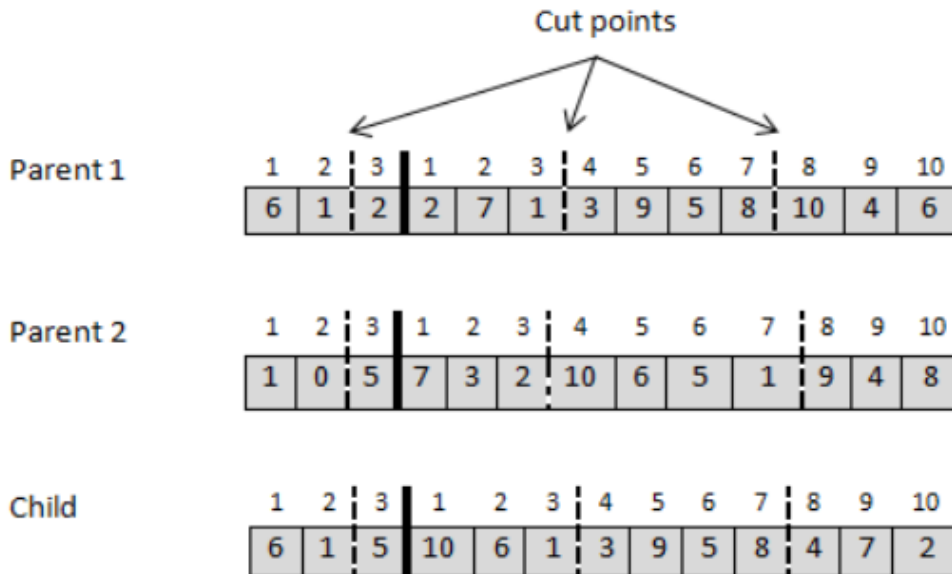


Figure 2.7 : opérateur de croisement

Procedure: MemeticAlgorithm Input : Problem data Output : Best solution P_{best}
$t \leftarrow 0$ Initialize population $P(t)$ using GenInitialSol procedure while ($t < \text{max gen}$) do Select parents p_1 and p_2 using binary tournament
Cross p_1 and p_2 to obtain child c Repair c using ChromRepair procedure if $c \in P(t)$ then Repeat selection and crossover else Apply LocalSearch procedure to c with a probability p_b to obtain c' if $c' \notin P(t)$ then remove worst solution from $P(t)$ and add c' to $P(t)$ end if end if end while Return P_{best}

Figure 2.8 : algorithme methode1

La recherche locale utilisée après chaque croisement est basée sur une exploration de voisinage selon quatre types d'opérateur :

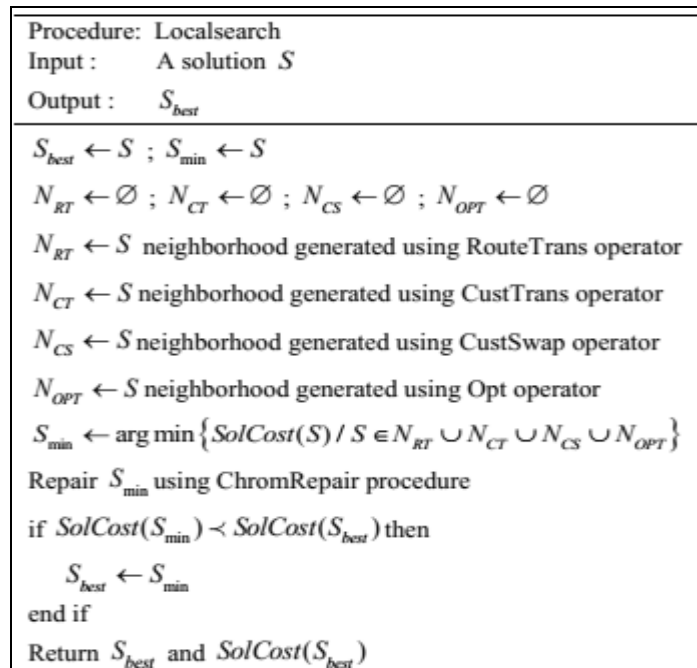


Figure 2.9 : algorithme recherche locale

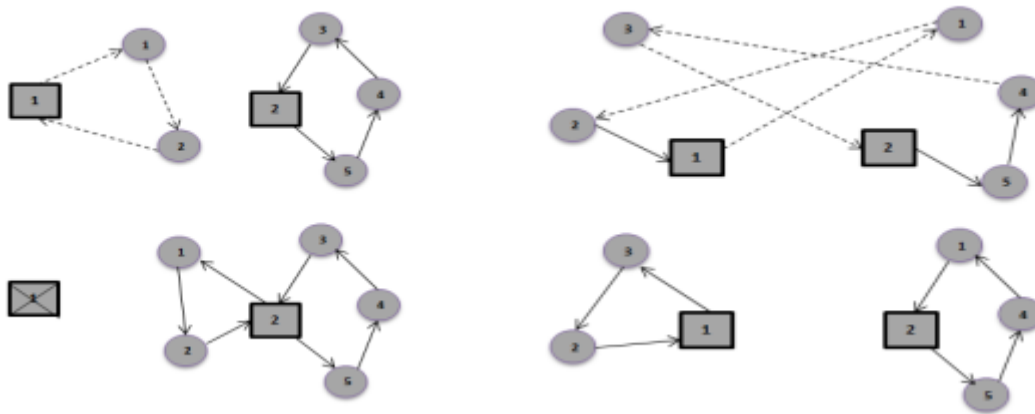


Figure 2.10 : opérateur de voisinages

4.1.2 Résultats :

La méthode est appliquée sur un ensemble de jeux de données et les résultats des tests sont résumés dans le tableau suivant :

Number of potential depots	Number of customers	GRASP			MA			Gap
		Number of opened depots	Number of tours	cost	Number of opened depots	Number of tours	cost	
5	20	2.5	4	45144	2.5	4	45301	0.35
5	50	2.62	9	74701	2.62	9	74685	0.02
5	100	2.33	16	202210	2.33	15.83	201303	0.45
10	100	3.33	18.33	257048	3.33	17.66	250040	2.73
10	200	3	35	447657	3	34.5	436293	2.54
Average		2.76	16.47	205352	2.76	16.2	201524.4	1.08

Tableau 1 : résultat des tests de la méthode 1

4.2 Méthode 2 : une approche GRASP+ ELS pour le CLRP

4.2.1 Principes :

Cet algorithme est proposé par C. Duhamel et al dans un article intitulé [31] « *A GRASP×ELS approach for the Capacitated location-routing problem* ». C'est une hybridation d'un GRASP avec un ELS. Elle tire parti des deux méthodes :

- l'approche multi-start du GRASP, basée sur une heuristique aléatoire qui fournit n_p solutions initiales, améliorées par une recherche locale
- l'efficacité d'ELS, une extension de la recherche locale itérée (ILS). Un ILS classique construit n_i solutions successives améliorées par la Recherche locale, créant à chaque itération une solution enfant, en utilisant la mutation et recherche locale. ELS est similaire mais génère $nc > I$ children-solutions à chaque itération et sélectionne la meilleure.

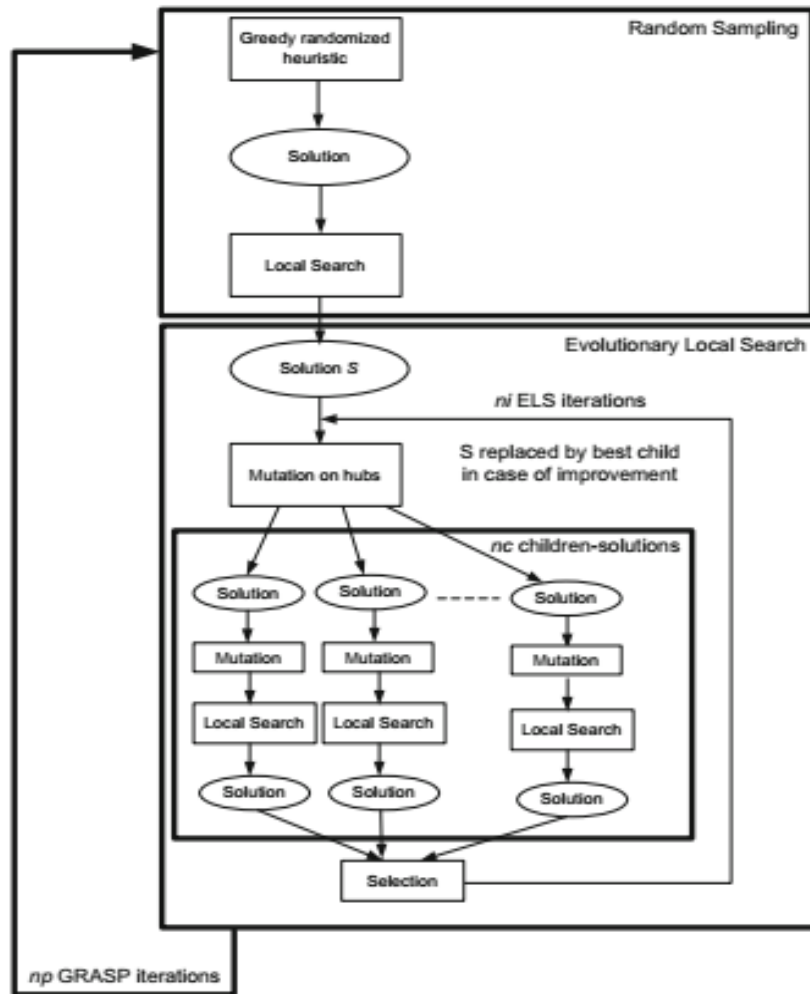


Figure 2.11 : algorithme methode2

La recherche locale dans cet algorithme est basée sur un ensemble d'opérateur de voisinage

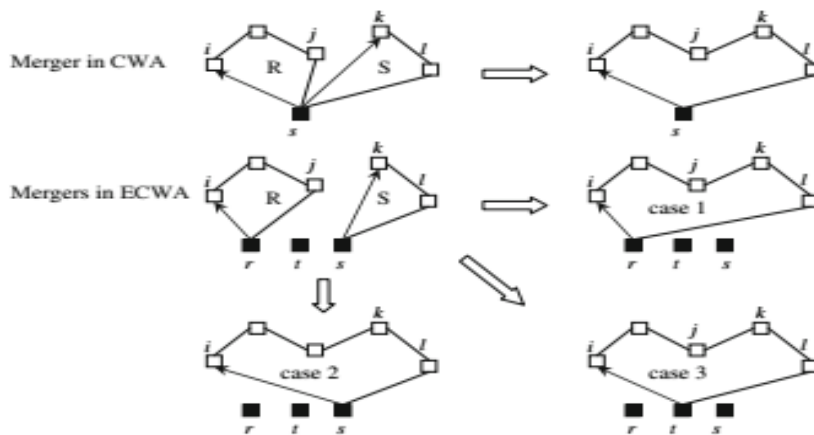


Figure 2.12 : opérateurs de voisinage

4.2.2 Résultats :

La méthode est appliquée sur un ensemble de jeux de données et les résultats des tests sont résumer dans le tableau suivant :

	GRAPS				MAPM				LRGTS				GRASP+ELS					
	LB	BKR	Cost	Gap LB	Gap BKR	CPU	Cost	Gap LB	Gap BKR	CPU	Cost	Gap LB	Gap BKR	CPU	Cost	Gap LB	Gap BKR	CPU
Christofides69-50x5	551.1	565.6	599.1	8.71	5.93	2	565.6	2.63	0.00	3	586.4	6.41	3.68	2	565.6	2.63	0.00	8
Christofides69-75x10	791.4	861.6	861.6	8.87	0.00	8	866.1	9.43	0.52	7	863.5	9.11	0.22	8	850.8	7.50	-1.26	86
Christofides69-100x10	818.1	842.9	861.6	5.31	2.22	21	850.1	3.91	0.86	36	842.9	3.02	0.00	22	833.4	1.87	-1.12	127
Daskin95-88x8	347.0	355.8	356.9	2.84	0.30	14	355.8	2.53	0.00	27	368.7	6.25	3.63	14	355.8	2.53	0.00	130
Daskin95-150x10	43406	44011.7	44625.2	2.81	1.39	125	44011.7	1.40	0.00	204	44386.3	2.26	0.85	95	43963.6	1.28	-0.11	1697
Gaskell67-21x5	424.9	424.9	429.6	1.10	1.10	0	424.9	0.00	0.00	0	424.9	0.00	0.00	0	424.9	0.00	0.00	0
Gaskell67-22x5	585.1	585.1	585.1	0.00	0.00	0	611.8	4.56	4.56	0	587.4	0.39	0.39	0	585.1	0.00	0.00	15
Gaskell67-29x5	512.1	512.1	515.1	0.59	0.59	0	512.1	0.00	0.00	1	512.1	0.00	0.00	0	512.1	0.00	0.00	9
Gaskell67-32x5	562.2	571.9	571.9	1.73	0.00	1	571.9	1.73	0.00	1	584.6	3.98	2.21	1	562.2	0.00	-1.70	18
Gaskell67-32x5	504.3	504.3	504.3	0.00	0.00	1	534.7	6.02	6.02	1	504.8	0.09	0.09	1	504.3	0.00	0.00	34
Gaskell67-36x5	460.4	460.4	460.4	0.00	0.00	1	485.4	5.44	5.44	1	476.5	3.50	3.50	1	460.4	0.00	0.00	0
Min92-27x5	3062.0	3062.0	3062.0	0.00	0.00	0	3062.0	0.00	0.00	1	3065.2	0.11	0.11	0	3062.0	0.00	0.00	35
Min92-134x8	5423.0	5809.0	5965.1	10.00	2.69	40	5950.1	9.72	2.43	89	5809.0	7.12	0.00	38	5719.3	5.46	-1.55	280
Avg.				3.23	1.09			3.64	1.53				3.25	1.13		1.64	-0.44	

Tableau 2 : résultat méthode 2

4.3 Méthode 3 : une heuristique pour le CLRP

4.3.1 Principes

Cet algorithme est proposé par A. Jokar et al dans un article intitulé [33] « *A Heuristic Based Approach to Solve a Capacitated Location-routing Problem* » l’heuristique proposé dans l’article est basé sur le principe de recuit simulé qui est généralement connue par deux caractéristiques : la facilité de son implémentation, et sa rapidité d’exécution comparée à d’autres métaheuristiques. En outre, elle fournit de bons résultats pour de nombreux problèmes d’optimisation combinatoire. Par analogie avec la thermodynamique, le recuit simulé repose sur le principe d’accepter une transformation locale qui engendre une dégradation de la solution courante, dans l’attente d’avoir une amélioration ultérieurement, ce qui permet d’éviter d’être piégé dans des minima locaux. En métallurgie, des transformations élémentaires sont appliquées sur le système thermodynamique en réduisant lentement la température le long du processus de recuit, afin d’atteindre un état d’équilibre qui correspond à une énergie minimale. L’algorithme est basé sur une étape de recherche locale en utilisant les opérateurs de voisinage suivants :



Figure 2.13 2-OPT

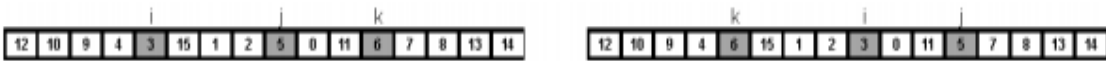


Figure 2.14 3-OPT

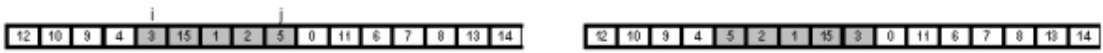


Figure 2.15 Inverse 2-opt

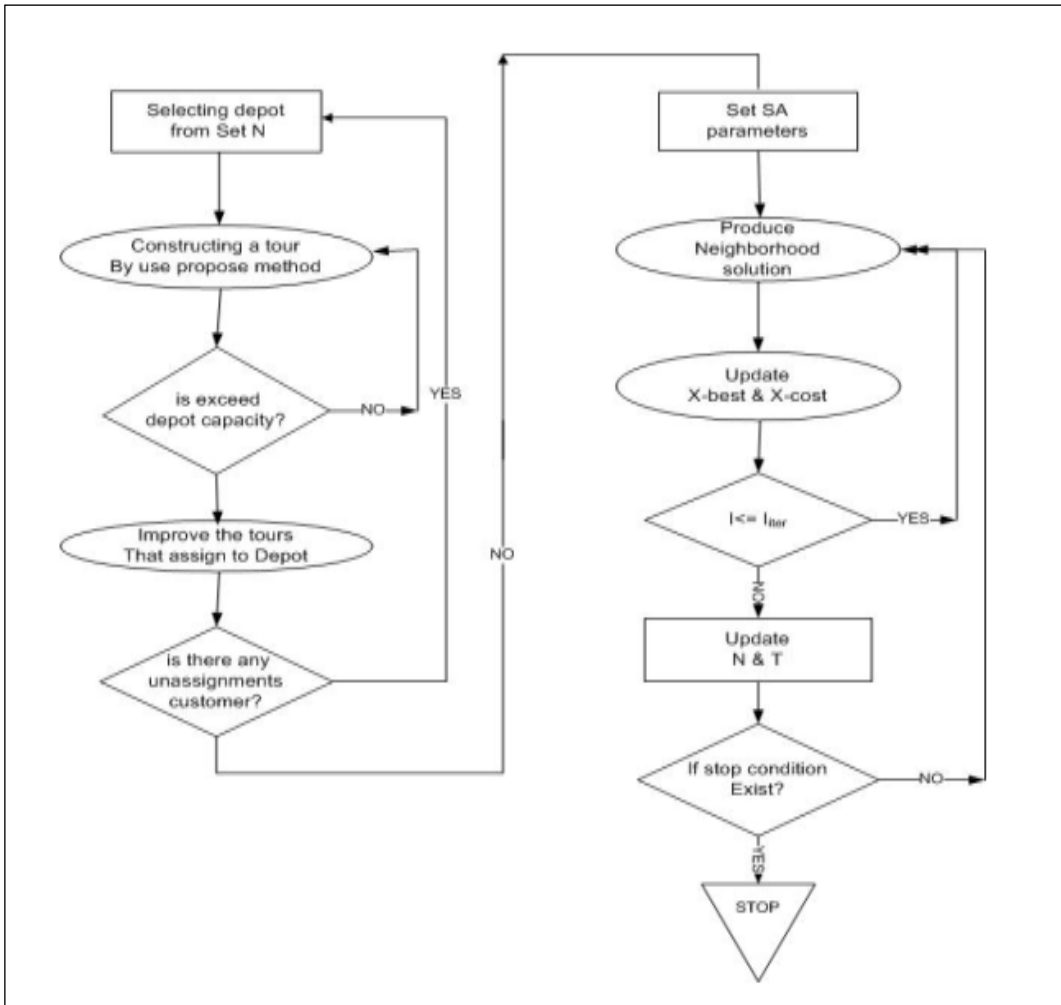


Figure 2.16 : algorithme methode3

4.3.2 Résultats :

La méthode est appliquée sur un ensemble de jeux de données et les résultats des tests sont résumés dans le tableau suivant :

CLRP Instance	Vehicle capacity	BKS	Heuristic method	Gap %	Heuristic method	Sahraeian method	Gap %
Christofides69-50x5	160	582.7	565.6	-2.93	565.6	582.7	-2.93
Christofides69-75x10	140	886.3	896.9	1.3	896.9	886.3	1.2
Christofides69-100x10	200	889.4	891.5	0.24	891.5	889.4	0.24
Gaskell67-21x5	6000	432.7	424.9	-1.8	424.9	427.7	-0.65
Gaskell67-22x5	4500	587.9	585.1	-0.48	585.1	591.5	-1.08
Gaskell67-29x5	4500	512.1	512.1	0	512.1	522.4	-1.97
Gaskell67-32x5	8000	570.5	562.3	-1.45	562.3	567.2	-0.88
Gaskell67-32x5	11000	510.9	504.3	-1.31	504.3	504.3	0
Gaskell67-36x5	250	470.7	460.4	-2.19	460.4	469.2	-1.87
Perl83-12x2	140	204	201.5	-1.22	201.5	205.3	-1.8
Perl83-55x5	160	1121.1	1121.4	0	1121.4	1183.1	-5.2

Tableau 3 : résultat méthode 3

4.4 Méthode 4 : un algorithme évolutionnaire simple et efficace pour le CLRP

4.4.1 Principes

Cet algorithme est proposé par R. Borges Lopes et al dans un article intitulé [32] « *A simple and effective evolutionary algorithm for the capacitated location–routing problem* ». L’algorithme est basé sur hybridation des principes algorithme génétique avec les avantages de la recherche locale.

Un opérateur de croisement évolué est proposé dans cet article on se basant sur un transfert d’un pourcentage des tournées vers les solutions fils. Pour la mutation deux principes sont proposé le premier réalise un changement des dépôts ouverts par ceux qui sont fermés et le deuxième opérateur assure un changement de la place des clients dans les tournées.

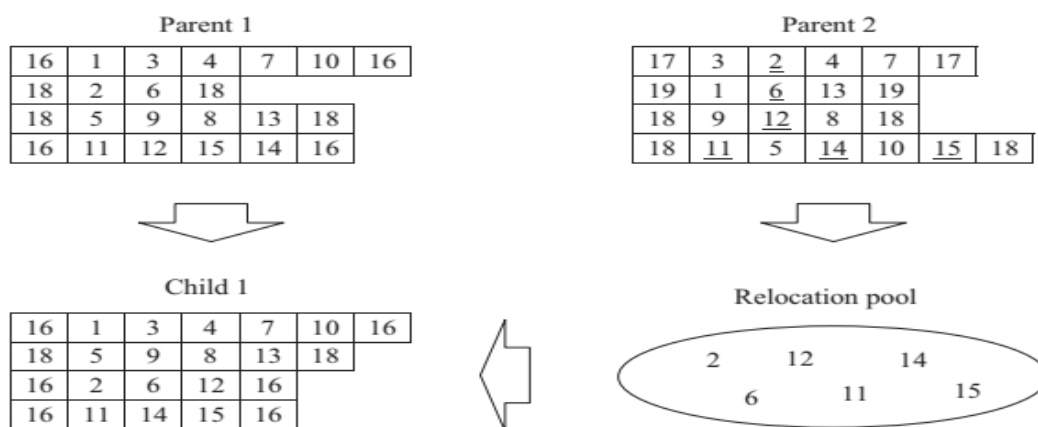


Figure 2.18: opérateur de croisement

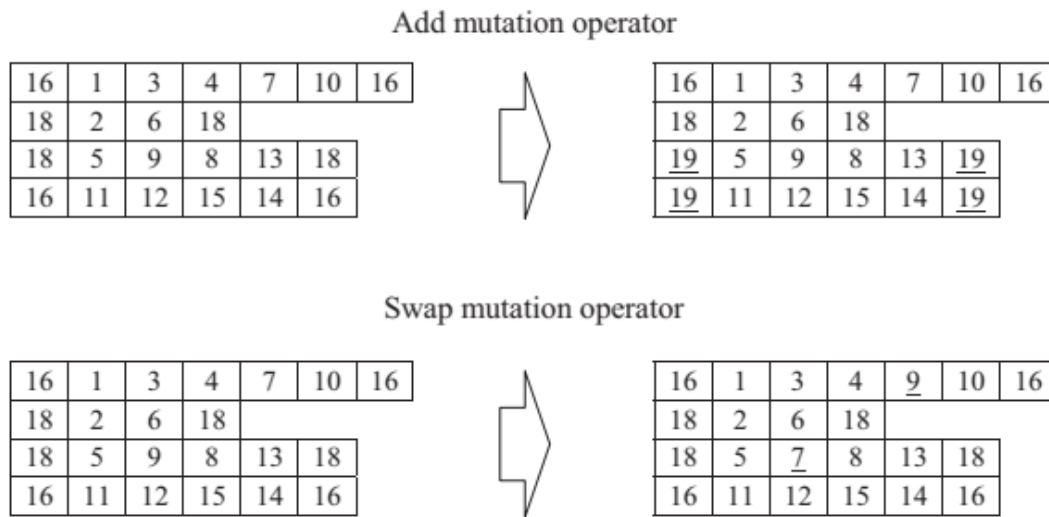


Figure2.19 : opérateur de mutation

```

1: //Initialization
2:  $\alpha := 0$ 
3: Fitness(BestSolution) :=  $+\infty$ 
4: Generate(P)
5: //Main loop
6: Repeat
7:    $\alpha := \alpha + 1$ 
8:   //Crossover
9:   Select two parents  $P_1$  and  $P_2$  using binary tournament
10:  Apply RCX to  $(P_1, P_2)$  and obtain children  $C_1$  and  $C_2$ 
11:  Add  $(C_1, C_2)$  to population P
12:  Remove from P the two worst solutions
13:  for each solution in P
14:    //Mutation
15:    if random <  $p_{location}$  then AddMutation(solution) end if
16:    if random <  $p_{routing}$  then SwapMutation(solution) end if
17:    if random <  $p_{location}$  then  $LS_{location}(solution)$  end if
18:    if random <  $p_{routing}$  then  $LS_{routing}(solution)$  end if
19:    //Memorization
20:    if Fitness(solution) < Fitness(BestSolution) then
21:      BestSolution := solution
22:       $\alpha := 0$ 
23:    end if
24:  end for
25: until  $\alpha = \alpha_{max}$ 
26: return BestSolution

```

Figure 2.20 : algorithme méthode 4

4.4.2. Résultat

La méthode est appliquée sur un ensemble de jeux de données et les résultats des tests sont résumé dans le tableau suivant :

	Instance	LB	BKR	HybridGA				
				CPU	Average		Best	
					Result	Std dev	Gap _{BKR}	Result
1	Gaskell67	424.9	424.9	0.8	424.9	0.00	0.00	424.9
2	Gaskell67	585.1	585.1	0.6	585.1	0.00	0.00	585.1
3	Gaskell67	512.1	512.1	0.9	512.1	0.00	0.00	512.1
4	Gaskell67	562.2	562.2	1.1	562.2	0.00	0.00	562.2
5	Gaskell67	504.3	504.3	0.9	504.3	0.00	0.00	504.3
6	Gaskell67	460.4	460.4	1.4	460.4	0.00	0.00	460.4
7	Christofides69	565.6	565.6	4.2	565.6	0.00	0.00	565.6
8	Christofides69	810.9	844.4	23.1	853.7	0.69	1.10	846.5
9	Christofides69	833.4	833.4	17.2	839.0	0.36	0.67	833.4
10	Perl83	204.0	204.0	0.1	204.0	0.00	0.00	204.0
11	Perl83	1074.8	1112.1	20.9	1112.8	0.03	0.07	1112.1
12	Perl83	1568.1	1622.5	33.7	1623.6	0.04	0.07	1622.5
13	Perl83		557,275.2	256.7	580,397.2	0.92	4.15	572,038.3
14	Perl83		670,118.5	215.0	685,951.5	0.55	2.36	680,246.9
15	Min92	3062.0	3062.0	1.0	3062.0	0.00	0.00	3062.0
16	Min92	5423.0	5709.0	62.9	5794.7	0.62	1.50	5730.5
17	Daskin95	351.5	355.8	32.4	356.0	0.04	0.05	355.8
18	Daskin95	43,406.0	43,919.9	96.0	44,179.6	0.30	0.59	43,993.4
19	Or76	12,048.4	12,290.3	36.6	12,324.8	0.25	0.28	12,290.3
			Average	42.4		0.20	0.57	
			Median	17.2		0.03	0.05	

Tableau 4 : résultat méthode 4.

Chapitre 3 : Contribution

Une Approche Génétique pour le CLRP

1. Introduction

Dans les chapitres précédents nous avons présenté un état de l'art sur les notions et les concepts liés à notre domaine d'étude. Une description de quelques méthodes proposées pour la résolution du problème de CRLP est établie ainsi que les résultats obtenus sur des jeux de données standards.

Dans le présent chapitre nous allons présenter notre approche basée sur les concepts des algorithmes génétiques. L'approche est basée sur un système de codage simple et des opérations de croisements et de mutation inspirés des opérateurs généraux utilisés dans la résolution des problèmes de tournées de véhicule.

2. Pourquoi une approche génétique ?

Le choix d'une approche génétique dans notre proposition est orienté par l'étude de la majorité des méthodes proposées dans plusieurs articles traitant le problème de CLRP. La majorité de ces articles ont prouvé l'efficacité des approches évolutionnaires (génétique).

3. Algorithme génétique

Avant de présenter l'application de l'approche génétique sur notre problème nous présentons dans cette section une description générale du fonctionnement des algorithmes génétiques.

3.1 Principes

Les algorithmes génétiques utilisent la théorie de Darwin sur l'évolution des espèces. Elle repose sur trois principes : le principe de variation, le principe d'adaptation et le principe d'hérédité.[14]

3.1.1. Le principe de variation

Chaque individu au sein d'une population est unique. Ces différences, plus ou moins importantes, vont être décisives dans le processus de sélection.

3.1.2. Le principe d'adaptation

Les individus les plus adaptés à leur environnement atteignent plus facilement l'âge adulte. Ceux ayant une meilleure capacité de survie pourront donc se reproduire davantage.

3.1.3. Le principe d'hérédité

Les caractéristiques des individus doivent être héréditaires pour pouvoir être transmises à leur descendance. Ce mécanisme permettra de faire évoluer l'espèce pour partager les caractéristiques avantageuses à sa survie.

2.2. Paradigme

Ce paradigme, associé avec la terminologie de la génétique, nous permet d'exploiter les algorithmes génétiques : Nous retrouvons les notions de Population, d'Individu, de Chromosome et de Gène. [16]

- **La population** est l'ensemble des solutions envisageables.
- **L'individu** représente une solution.
- **Le Chromosome** est une composante de la solution.
- **Le Gène** est une caractéristique, une particularité.

2.3. Opération d'évolution

2.3.1. Sélection

La sélection consiste à choisir les individus les mieux adaptés afin d'avoir une population de solution la plus proche de converger vers l'optimum global. Cet opérateur est l'application du *principe d'adaptation* de la théorie de Darwin. Il existe plusieurs techniques de sélection.

Voici les principales utilisées : [17]

- **Sélection par rang** : Cette technique de sélection choisit toujours les individus possédant les meilleurs scores d'adaptation.
- **Probabilité de sélection proportionnelle à l'adaptation** : Technique de la roulette ou roue de la fortune, pour chaque individu, la probabilité d'être sélectionné est proportionnelle à son adaptation au problème.
- **Sélection par tournoi** : Cette technique utilise la sélection proportionnelle sur des paires d'individus, puis choisit parmi ces paires l'individu qui a le meilleur score d'adaptation.
- **Sélection uniforme** : La sélection se fait aléatoirement, uniformément et sans intervention de la valeur d'adaptation.

2.3.2 Croisement

Le croisement, ou enjambement, crossing-over, est le résultat obtenu lorsque deux chromosomes partagent leurs particularités. Celui-ci permet le brassage génétique de la population et l'application du *principe d'hérédité* de la théorie de Darwin. [19]

2.3.3 Mutation

La mutation consiste à altérer un gène dans un chromosome selon un facteur de mutation. Ce facteur est la probabilité qu'une mutation soit effectuée sur un individu. Cet opérateur est l'application du *principe de variation* de la théorie de Darwin et permet, par la même occasion, d'éviter une convergence prématurée de l'algorithme vers un extremum local. [20]

2.4 Algorithme

La résolution d'un problème d'optimisation en utilisant les principes de la génétique passe par les étapes suivantes :

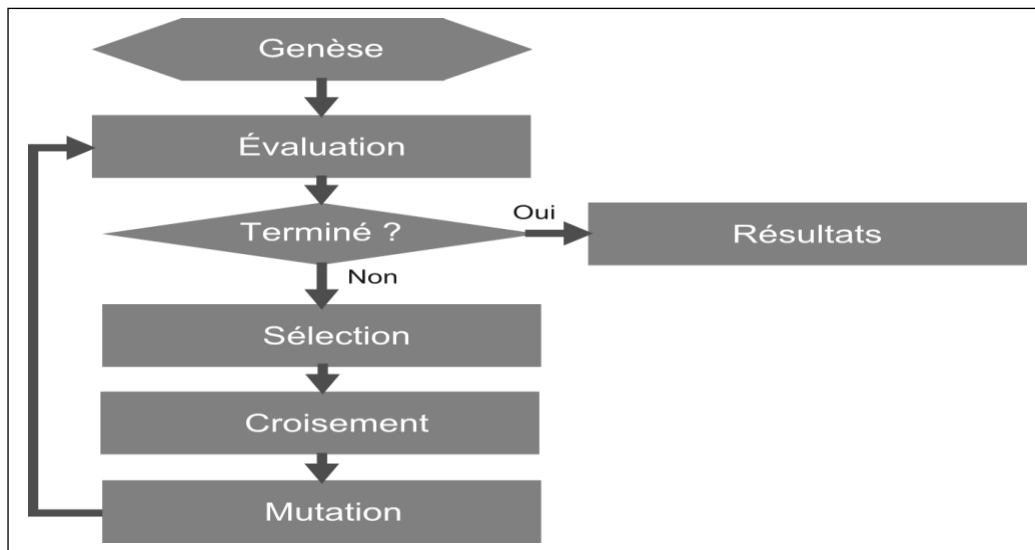


Figure 3.1 : étapes des algorithmes génétiques

- **La genèse** est l'étape de la création d'une population initiale. C'est le point de départ de notre algorithme
- **L'évaluation** est l'analyse des individus. Pour ceci, nous utilisons une fonction de coût, afin de définir le score d'adaptation des individus lors du processus de sélection. Nous effectuons une boucle tant que l'évaluation estime que la solution n'est pas optimale.

3. Une approche génétique pour le CLRP

Dans cette section nous décrivons en détails l'adaptation des principes des algorithmes pour la résolution de notre problème CLRP.

3.1 Codage des solutions

L'application d'une approche génétique sur un problème d'optimisation passe toujours par une étape cruciale « le codage des solutions » dans laquelle les futures solutions du problème sont

Présentées sous forme de chromosomes. Pour notre problème nous proposons un codage simple sous forme d'une liste d'éléments. Ce codage est inspiré des codages utilisés dans la résolution de plusieurs variantes des problèmes de tournées de véhicules (VRP).

- **Exemple** : pour une solution composée de 2 dépôts et 10 clients nous proposons la codification suivante : les tournées de chaque dépôt sont séparées par un nœud dépôt **suivie** par la liste des clients de chaque tournée.



Figure 3.2 : Codage des solutions

3.2 Construction de la population initiale

Le fonctionnement d'un algorithme génétique commence toujours par la génération d'une population initiale. Dans notre approche nous proposons une méthode de génération simple basée sur un choix aléatoire des dépôts à ouvrir selon les besoins et les demandes des clients du jour. Notre algorithme de génération débute par un choix aléatoire d'un premier dépôt après lequel les clients sont affectés au dépôt selon la capacité de ce dernier. Si le stock du dépôt est achevé et la liste des clients non servis est non vide, nous procédons à l'ouverture d'un nouveau dépôt. Cette opération sera répétée jusqu'à ce que la liste des clients non servis soit vide.

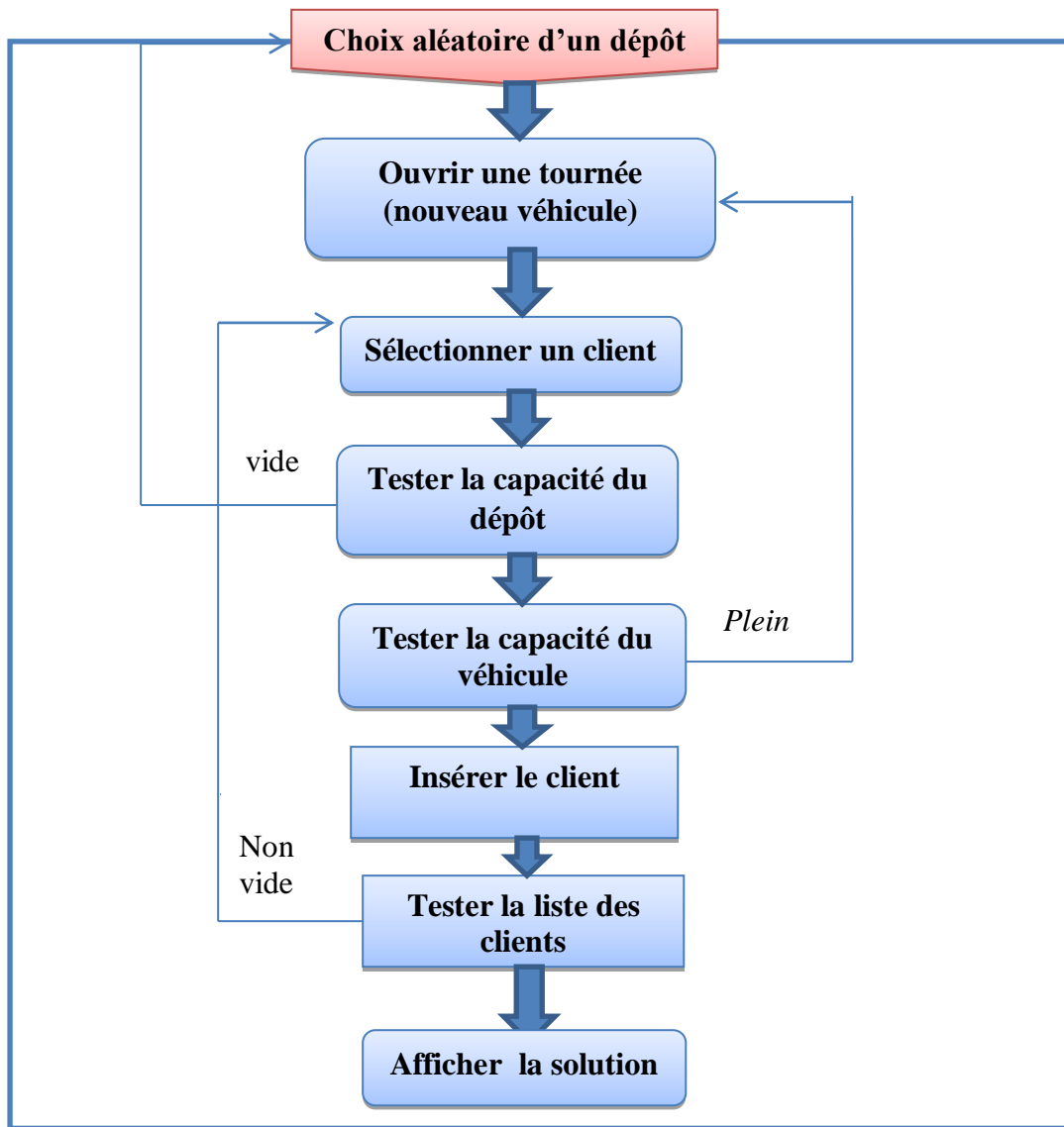


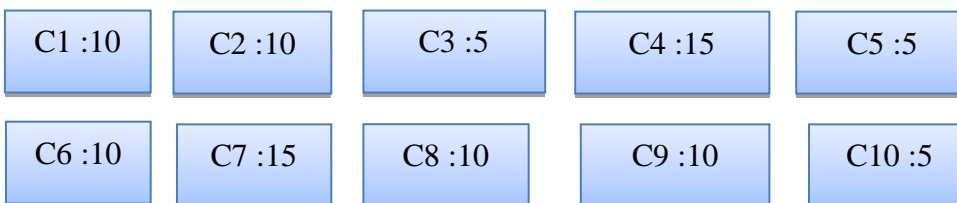
Figure 3.4 : Construction d'une seule solution (un élément de la population initiale)

• **Exemple :**

Nous avons 4 dépôts chaque dépôt a une capacité égale à 50.



Nous avons 10 clients chaque client a une demande



En appliquons l'algorithme nous obtenons la solution suivante :



3.3 Fonction d'évaluation

Le fonctionnement d'un algorithme génétique et le passage entre les générations sont basés sur une fonction d'évaluation qui quantifie la qualité des différentes solutions. Dans notre problème le coût d'une solution est calculé en utilisant la distance parcourue entre les différents points d'une solution (client/dépôt) sans oublier le coût d'ouverture des dépôts et d'utilisation des véhicules. (Voir chapitre2).

L'objectif alors est de Minimiser Z :

$$\text{Min } z = \sum_{i \in I} O_i * y_i + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} F * x_{ij}^k + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ij} x_{ij}^k \quad (2.15)$$

La description textuelle de cette fonction d'évaluation est :

Cout (solution)= \sum des coûts d'ouverture des dépôts + \sum distance parcourue entre les différents points de la solution (client/dépôt) dans les différentes tournées. + \sum cout d'utilisation d'un véhicule

Figure 3.4 : Fonction d'évaluation

3.4 Opération de croisement

Le croisement est l'opération responsable de la production des différentes populations du problème. Dans notre approche nous optons pour un croisement simple basé sur le choix d'un nombre de dépôt d'une solution S₁. Les tournées des dépôts choisis sont fusionnées pour former le début d'un enfant (une nouvelle solution) dans l'ordre d'apparition de ces derniers dans la solution S₁. Pour le reste des clients qui ne figurent pas encore dans la nouvelle solution une liste des clients non servis est formée à partir de la deuxième solution S₂ dans l'ordre d'apparition de ces clients dans la solution S₂. Nous illustrons l'opération de croisement par l'exemple suivant :

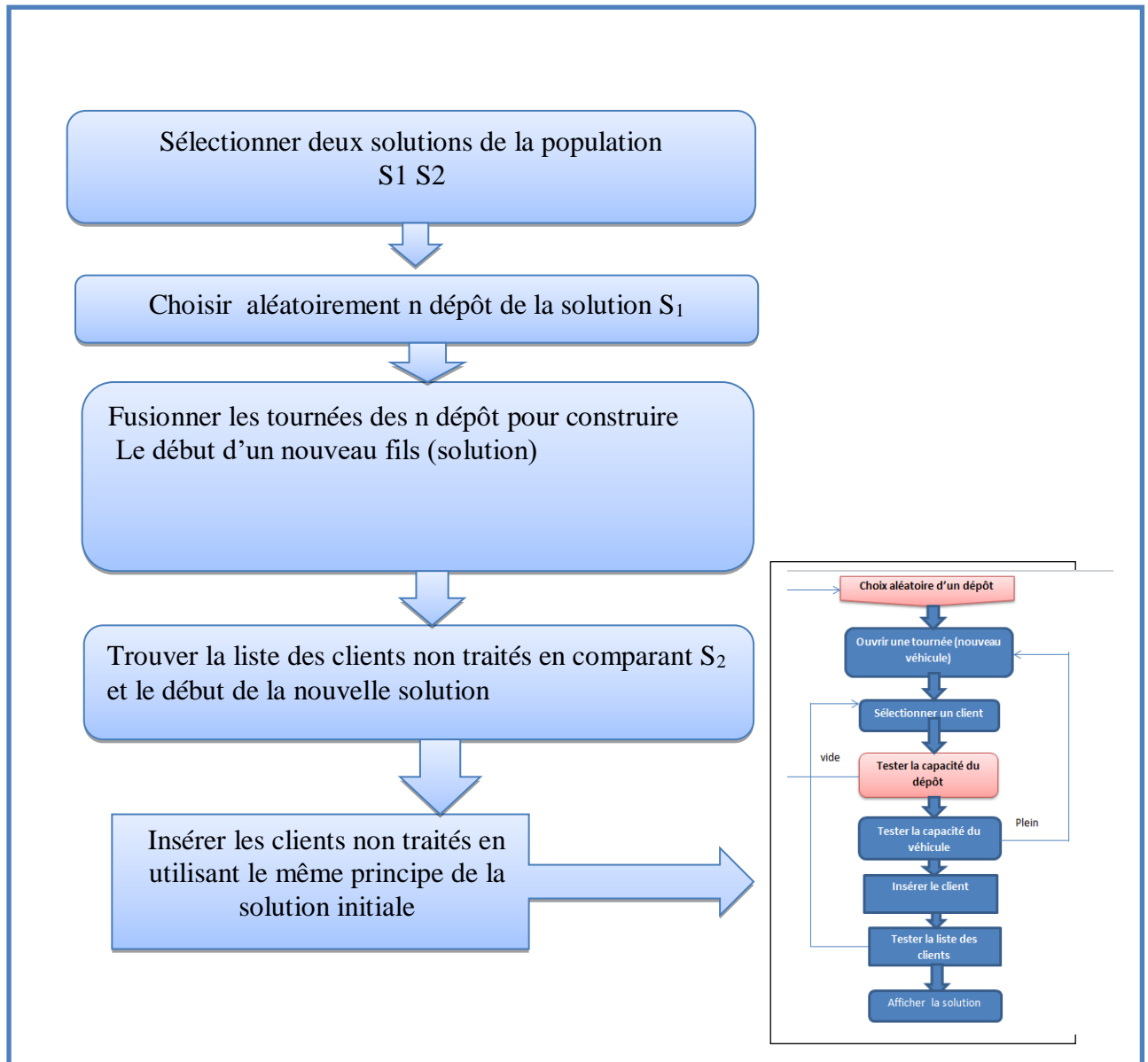
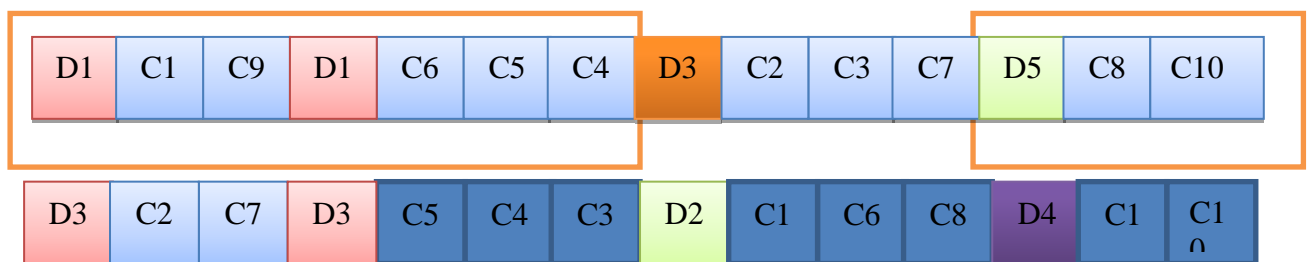


Figure 3.5 : Opération de croisement

• Exemple de croisement



Dans la solution S_1 nous choisissons les deux dépôts D_1 et D_5 et nous fusionnons les tournées de ces deux dépôts. Le début du nouvel enfant (solution) et le suivant :



Pour avoir une solution complète il faut trouver la liste des clients qui n'appartient pas à la nouvelle solution en comparant la solution S_2 et la nouvelle solution pour avoir la liste suivante des clients non affectés dans l'ordre.



Pour insérer ces clients dans notre nouvelle solution nous utilisons le même principe utilisé dans la création de la solution initiale. Nous ouvrons un (ou plusieurs) dépôt pour servir les clients restants.

Dans notre exemple nous ouvrons par exemple le dépôt 3 et nous obtenons la solution complète suivante :

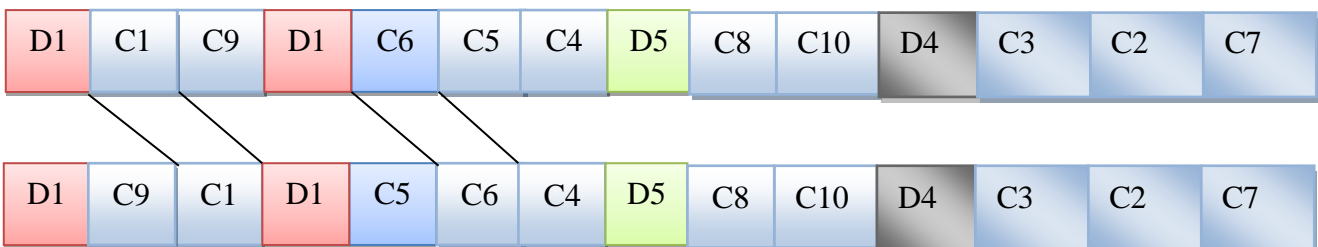


3.5 Opération de mutation

Pour assurer une variété entre les différentes solutions de la même population nous proposons un opérateur de mutation de base (classique)

Dans cet opérateur nous changeons la place d'un nombre de client dans la même tournée. Pour ne pas avoir un grand changement (mutation aléatoire), un petit nombre de client seulement est concerné par cette mutation.

Exemple : Si nous appliquons l'opérateur de mutation sur le client C1 et C6 nous obtenons la solution suivante :



L'opération de mutation dans notre problème assure un changement de la zone de recherche dans le grand espace des solutions.

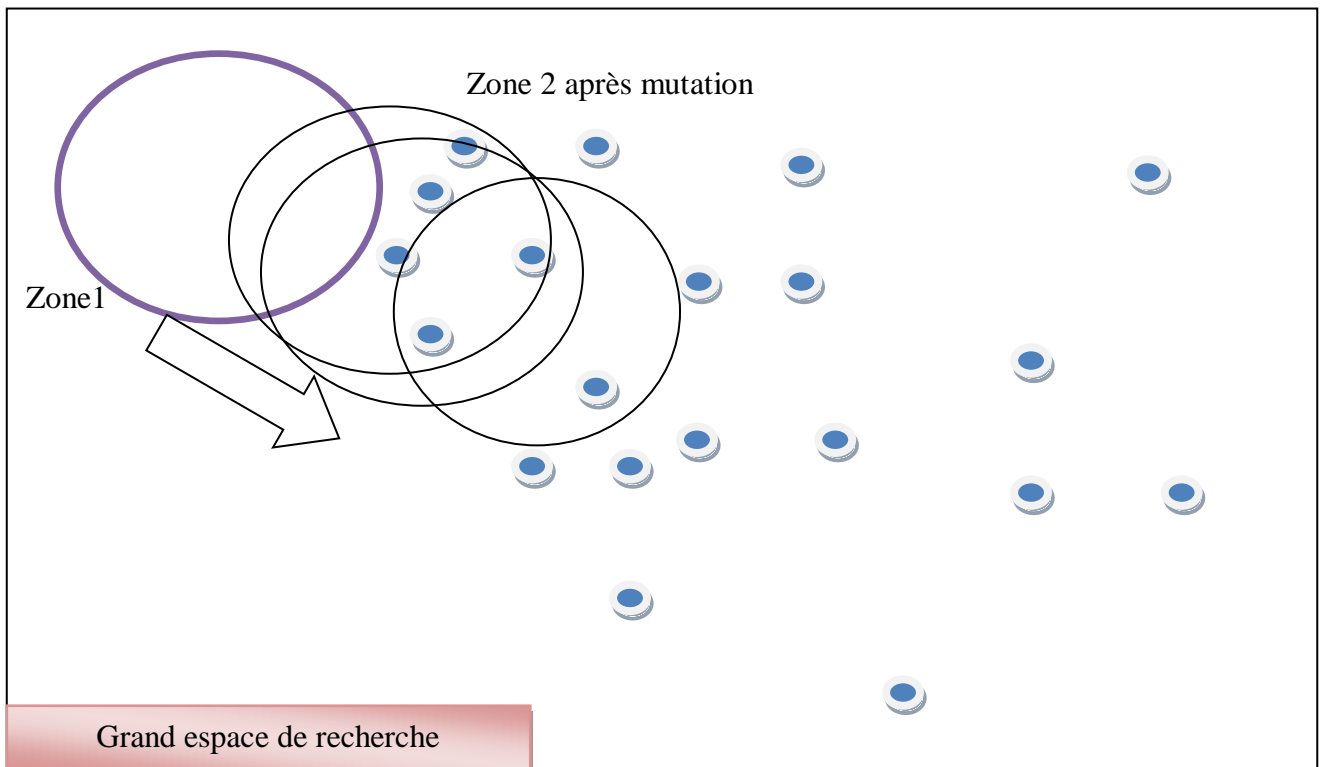


Figure 3.5 : changement de la zone de recherche par mutation

3.6 Opération de sélection

- L'opération de croisement est précédée toujours par une opération de sélection, dans notre cas nous choisissons l'utilisation de la Sélection par rang : Cette technique de sélection choisit toujours les individus possédant les meilleurs couts d'évaluation

4. Conclusion

Dans ce troisième chapitre nous avons expliqué les différents concepts utilisés dans notre approche génétique à savoir : le codage des solutions, les opérateurs de croisement et de mutation ainsi que la méthode adoptée pour la génération de la population initiale. Dans le prochain chapitre nous présentons les résultats de l'implémentation de notre approche sur des jeux de données standards.

Chapitre 4 :

Implémentation et tests

1. Introduction

Dans ce dernier chapitre nous présentons les résultats de l'implémentation des différents concepts de notre approche génétique. L'implémentation est réalisée en utilisant le langage java. Des tests sur des jeux de données standards sont aussi présentés.

2. Langage de programmation

Pour programmer notre approche génétique nous avons utilisé le langage de programmation java sous l'environnement netbeans. Java propose une structure d'organisation en classe des différents principes de notre approche.

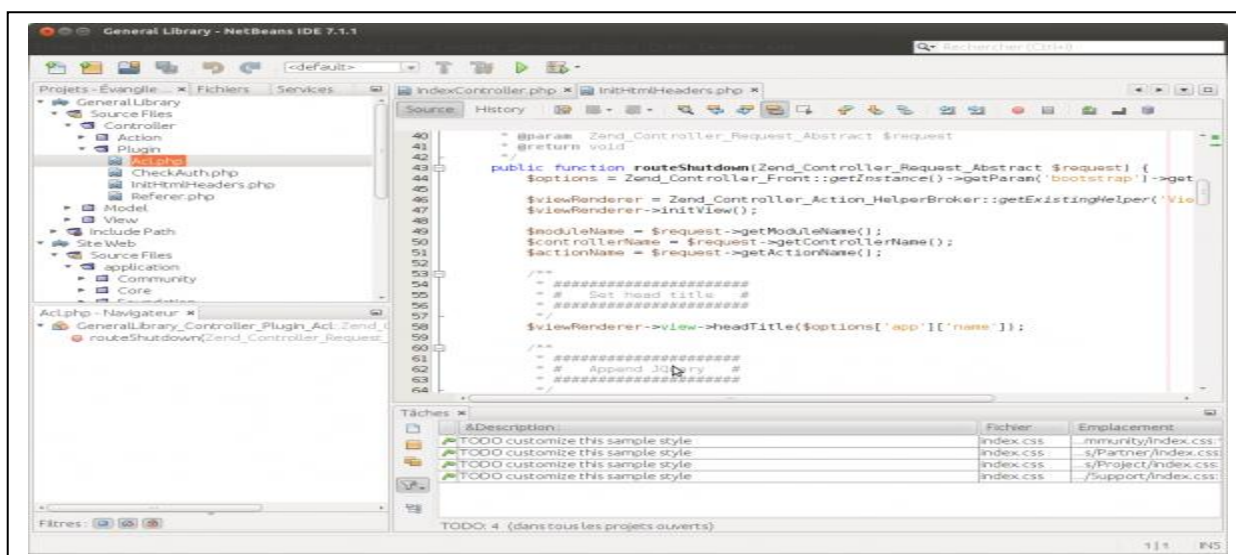


Figure 4.1. Netbeans

3. Matériel et environnements

Notre approche est implémentée sur un poste de travail qui a les caractéristiques suivantes :

- Processeur core i5.
- Une RAM de 4 G.
- Système d'exploitation windows 64 bits.

4. Jeux de données utilisés

4.1 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 :

http://prodhonc.free.fr/Instances/instances_us.htm

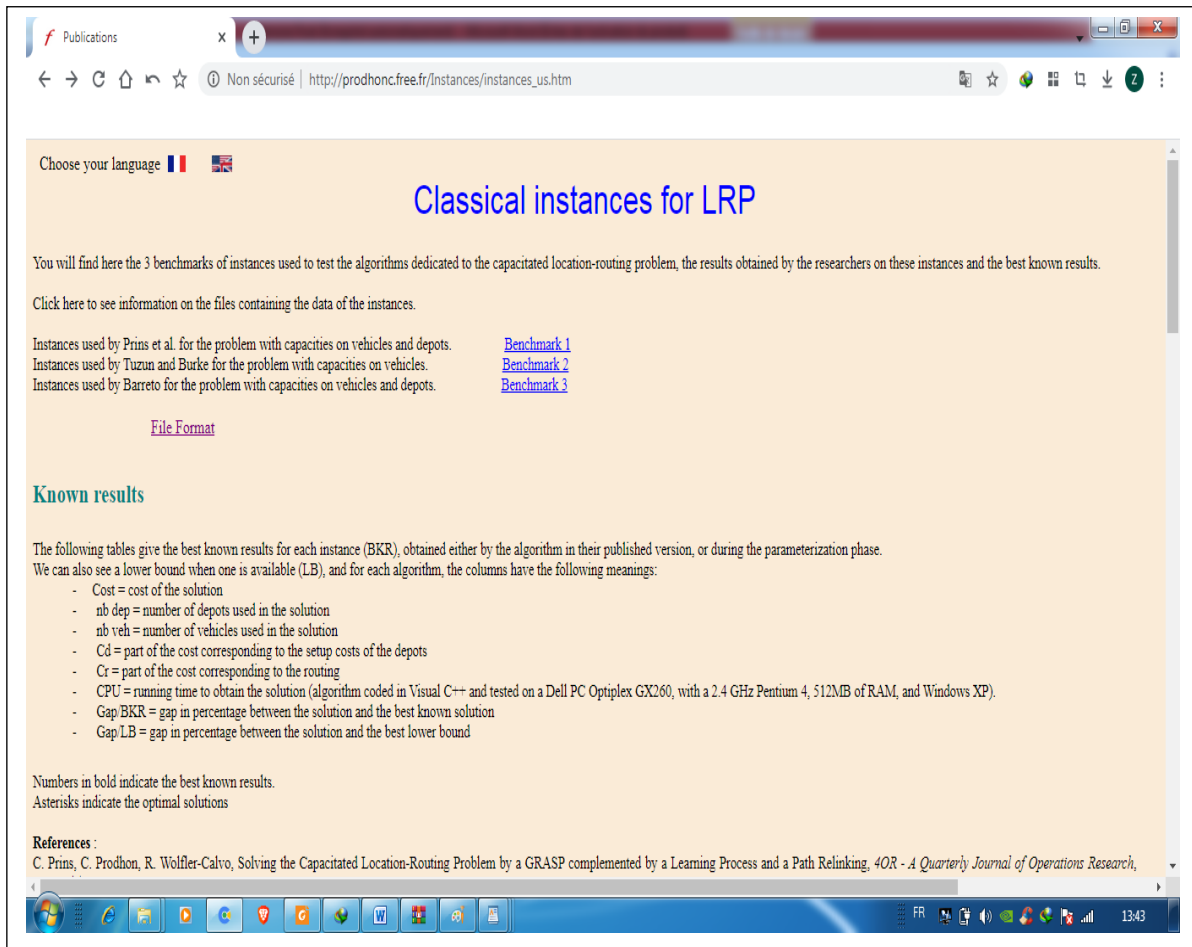


Figure 4.1 : page source des instances

Barreto a proposé un ensemble de fichiers de test composé de 14 éléments (fichier). Chaque fichier propose un contexte différent. 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.

coordChrist50.dat	767	296	Fichier DAT
coordChrist75.dat	1 128	415	Fichier DAT
coordChrist100....	1 589	532	Fichier DAT
coordDas88.dat	3 609	789	Fichier DAT
coordDas150.dat	6 405	1 571	Fichier DAT
coordGaspelle.dat	493	182	Fichier DAT
coordGaspelle2....	491	191	Fichier DAT
coordGaspelle3....	609	214	Fichier DAT
coordGaspelle4....	653	243	Fichier DAT
coordGaspelle5....	654	243	Fichier DAT
coordGaspelle6....	565	128	Fichier DAT
coordMin27.dat	1 018	263	Fichier DAT
coordMin134.dat	3 959	929	Fichier DAT
coordOr117.dat	5 391	1 119	Fichier DAT

Figure 4.2 : liste de fichiers de test proposés par Barreto

3.2 Structure des fichiers de test

Les fichiers de test proposés sont des fichiers texte simples. Ils ont la même structure. (Voir annexe1)

50		32
5		5
10	49	264 435
20	30	323 413
5	25	285 427
54	17	278 424
43	53	266 422
37	52	
9	49	298 427
2	64	309 445
0	26	307 464
0	30	336 475
1	47	320 439
7	63	321 437
1	62	322 437
2	33	323 433
51	21	324 433
42	41	323 429
31	32	314 435
5	25	311 442
12	42	304 427
36	16	293 421
52	41	296 418
27	23	261 384
17	33	297 410
13	13	315 407
57	58	314 406
62	42	321 391
42	57	321 398
		314 394

Figure 4.3 : un fichier de test

Chaque fichier de test est structuré d'une manière organisée pour faciliter la lecture et l'alimentation des algorithmes programmés.

Barreto a défini dans le fichier texte suivant la structure de tous ses jeux de données ainsi que la

méthode utilisée pour calculer la distance entre les points.

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:

$$\text{sqrt}((x_A-x_B)^2 + (y_A-y_B)^2)$$

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 4.4 : description de la structure des fichiers de test de barreto.

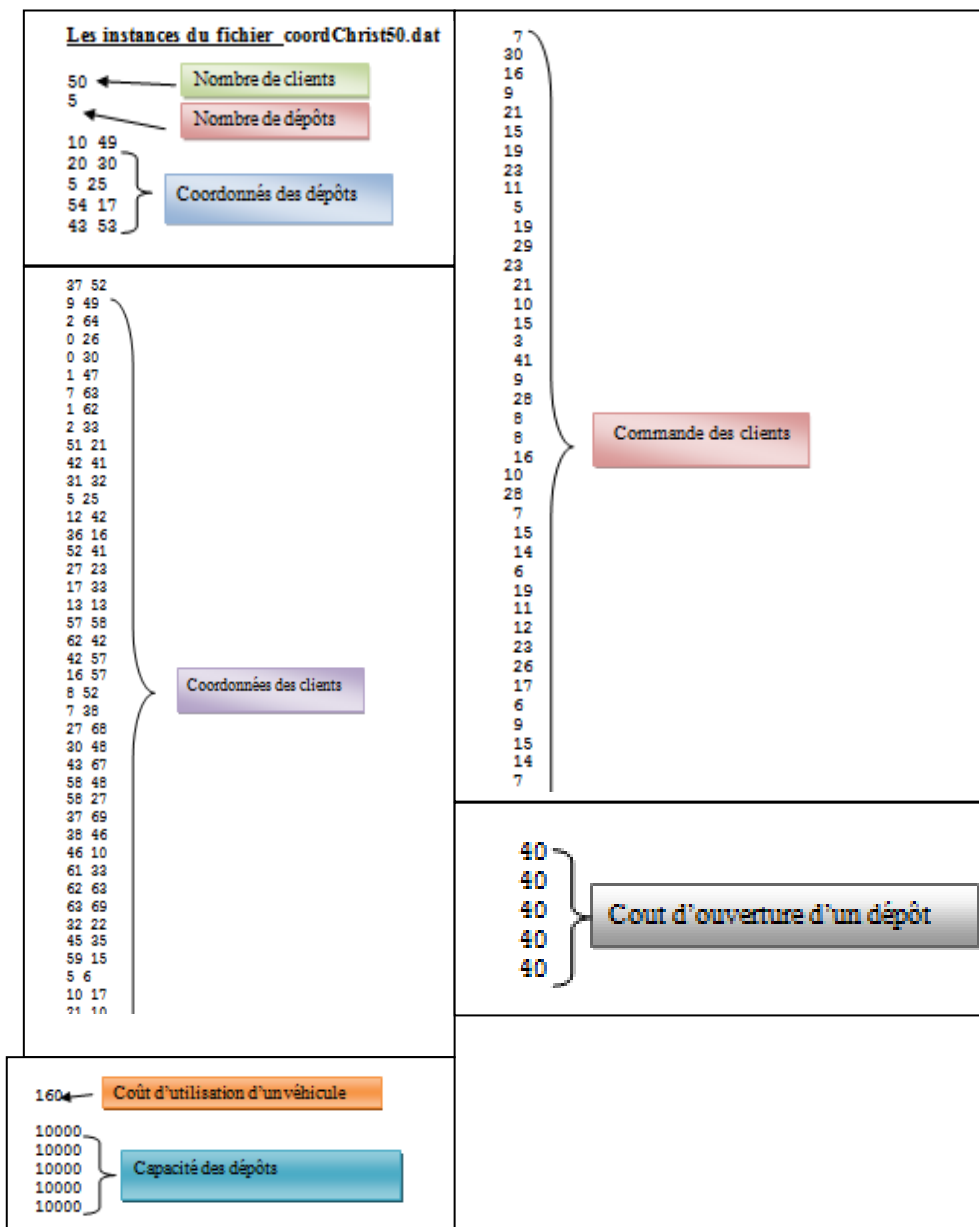


Figure 4.3 structure d'un fichier de test.

5. Interfaces de l'application



Figure 4.5 : Interface Principale

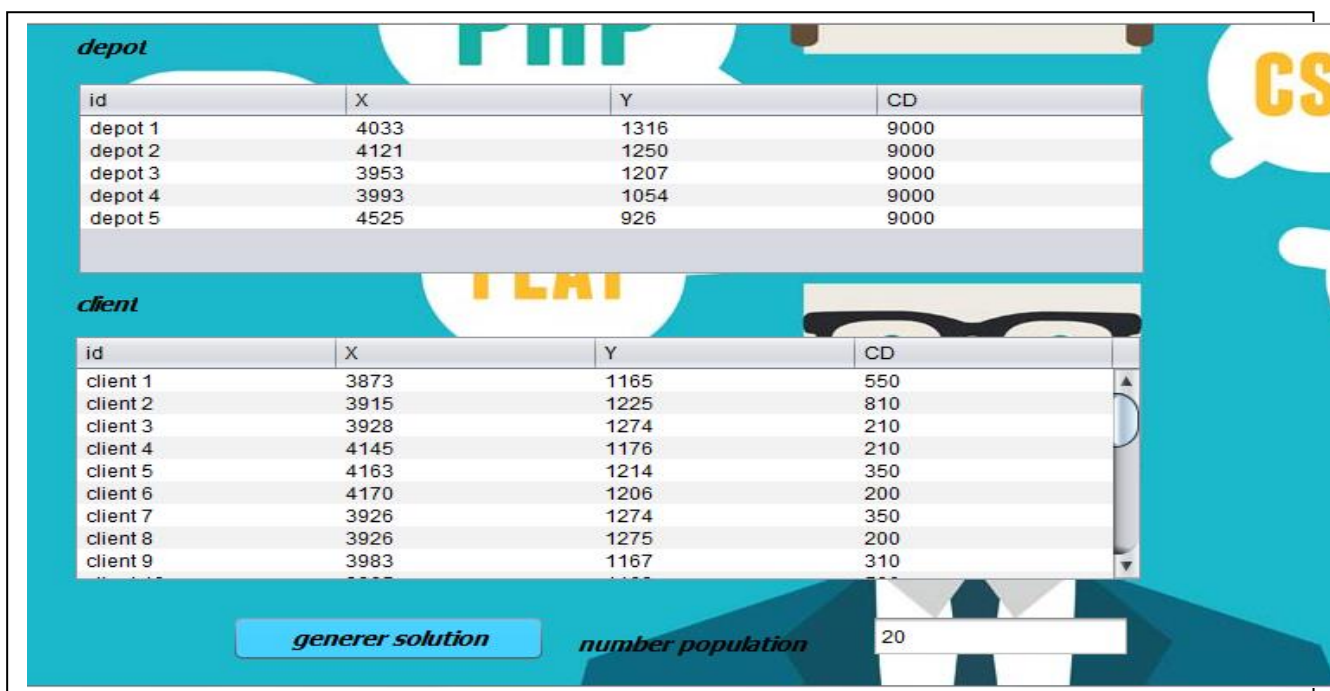


Figure 4.6 : Interface des liste clients et dépôts

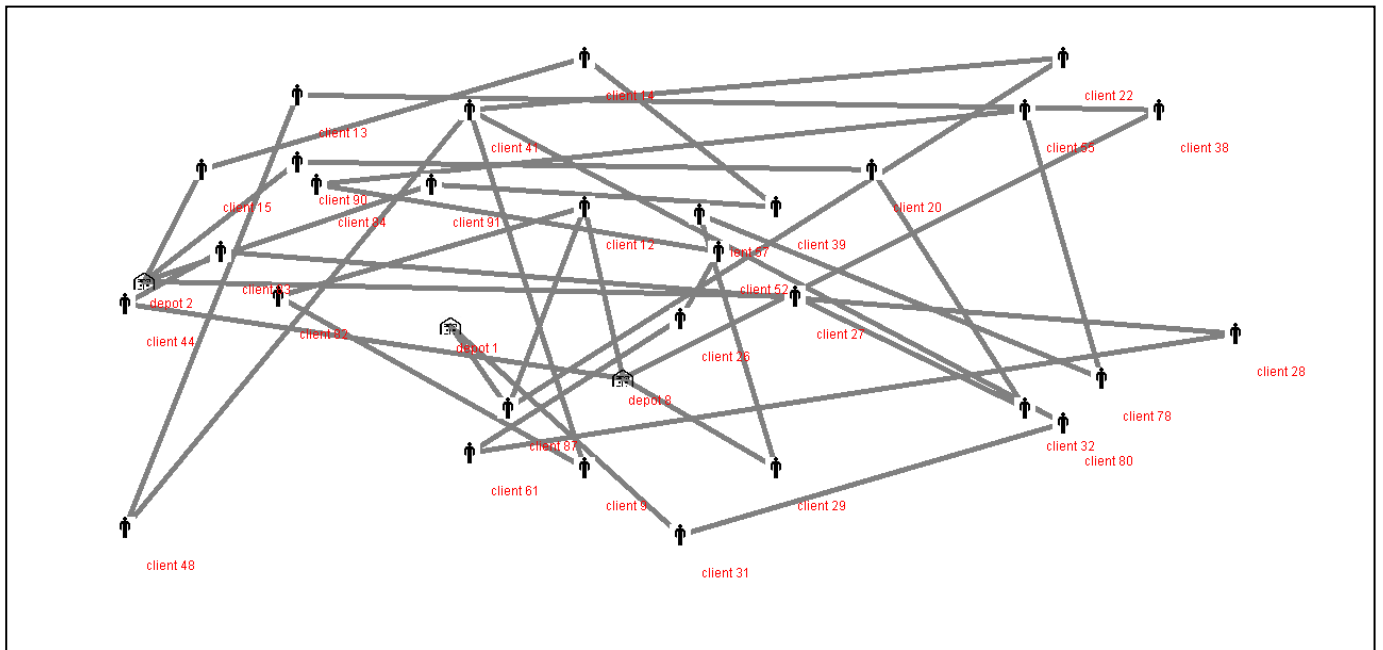


Figure 4.9 : graphe d'une solution

Tableau depot

solution 16 **cout total** 1171.4735706576107

depot	cout	N vicule	N client
depot 0	271.3507110379558	3	7
depot 1	210.73267876938192	3	8
depot 2	689.3901808502728	2	18

graph

Figure 4.8 : détail des solutions

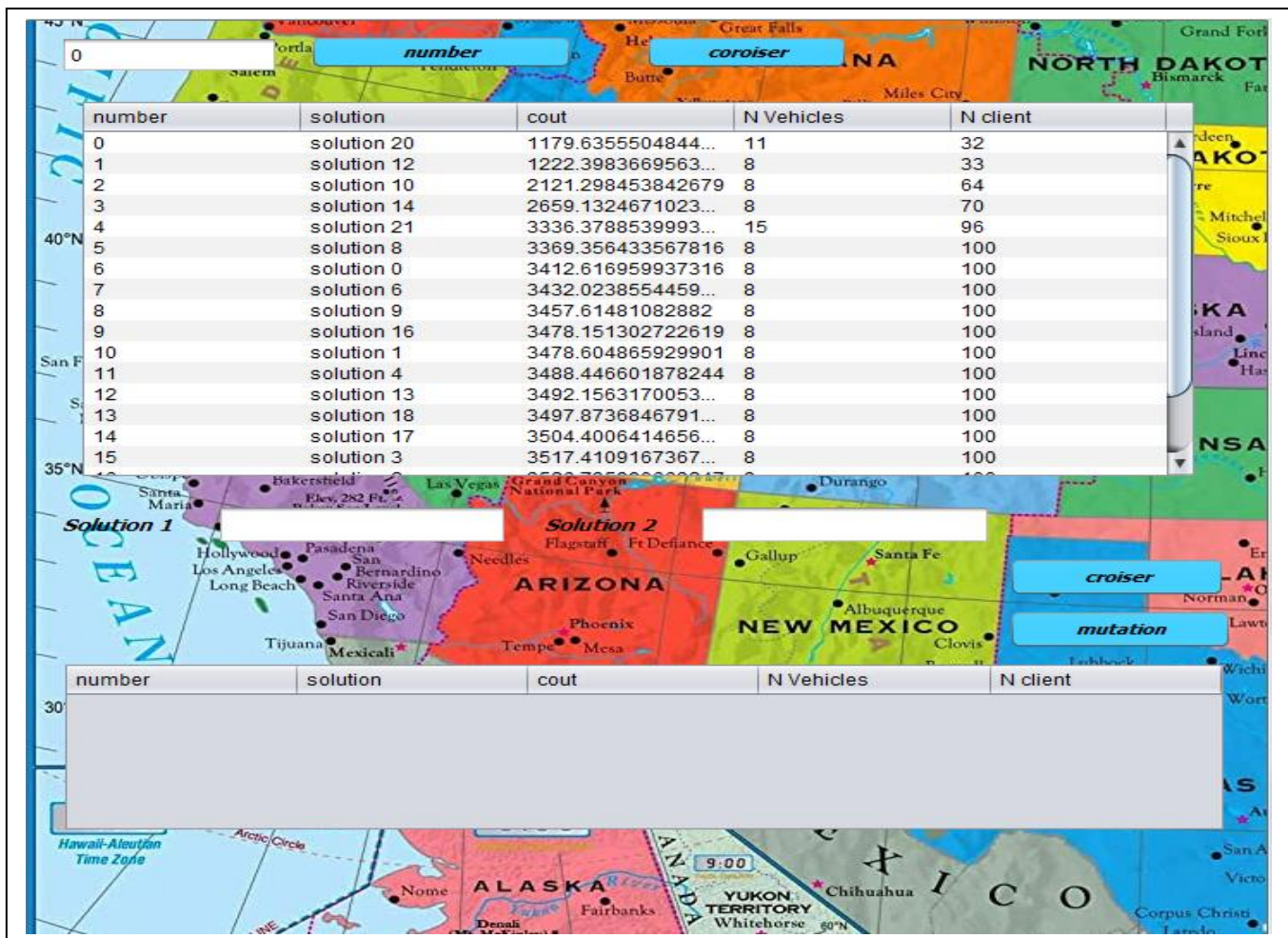


Figure 4.8 : résultat du fichier 2

6 Conclusion

Dans ce dernier chapitre nous avons présenté les résultats d'implémentation de notre approche génétique en utilisant le langage de programmation java. Les principes de nos approches ont été testés sur des jeux de données standard utilisés dans la majorité des articles qui traitent le même problème.

Conclusion Générale

L'objectif de ce travail de master était de proposer et d'implémenter une approche pour la résolution d'un sous type de problèmes de tournées de véhicules (CLRP). 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 coûts engendrés par les opérations de livraisons.

Les deux premiers chapitres de ce mémoire sont dédiés au positionnement du problème étudié (CLRP) par rapport au problème général d'optimisation et par rapport aux différents types des problèmes de tournées de véhicules. Une description mathématique du problème est présentée ainsi qu'une explication textuelle des différentes contraintes des problèmes. Ces deux chapitres sont clôturés par une étude comparative de quatre méthodes récentes proposées dans des articles pour résoudre ce même problème. Les deux derniers chapitres sont consacrés à la proposition et l'implémentation d'une approche génétique inspirée des méthodes étudiées dans les chapitres précédents.

Notre approche est basée sur un codage simple de solution sous forme d'une suite d'éléments (clients/dépôt). Deux types d'opérations de croisement et de mutation sont proposés pour assurer la génération des différentes populations (générations).

L'opérateur de croisement assure un héritage des tournées de quelques dépôts entre les différentes générations et l'opérateur de mutation assure un passage facile et efficace dans le grand espace de recherche. Pour la génération de la population initiale un algorithme d'insertion aléatoire des clients et des dépôts est proposé.

La majorité des opérateurs proposés dans notre approche sont inspirés des concepts décrits dans des articles de recherche récents.

Les principes de notre méthode sont implémentés et testés sur des jeux de données standards (les jeux de données de baretto)

Les perspectives futures de ce travail concernent notamment l'extension de l'approche par de nouveaux opérateurs de croisement et pourquoi pas une hybridation de notre approche génétique avec les principes de la recherche taboue en se basant sur des opérateurs de voisinages efficaces. Nous envisageons aussi une comparaison de nos résultats de test avec ceux des autres approches

Annexe 1

Les instances du fichier coordChrist50.dat

50 ← Nombre de clients
5 ← Nombre de dépôts
10 49
20 30
5 25
54 17
43 53

Coordonnés des dépôts

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
38 46
46 10
61 33
62 63
63 69
32 22
45 35
59 15
5 6
10 17
21 10
5 64
30 15
39 10
32 39

Coordonnées des clients

25 32
25 55
48 28
56 37

160

Coût d'utilisation d'un véhicule

10000
10000
10000
10000
10000

Capacité des dépôts

7
30
16
9
21
15
19
23
11
5
19
29
23
21
10
15
3
41
9
28
8
8
16
10
28
7
15
14
6
19
11
12
23
26
17
6
9
15
14
7
27
13
11
16

Commande des clients

10
5
25
17
18
10

40
40
40
40
40



Cout d'ouverture d'un dépôt

0

1

Les instances du fichier coordChrist75.dat

75
10

44 41
56 7
40 72
62 12
46 5
25 75
69 22
69 61
51 67
33 73

2 22
6 26
1 45
5 35
5 20
3 34
0 50
5 45
6 59
40 66
55 65
35 51
62 35
62 57
62 24
21 36
33 44
9 56
62 48
66 14
44 13
26 13

11 28
7 43
17 64
41 46
55 34
35 16
52 26
43 26
31 76
22 53
26 29
50 40
55 50
54 10
60 15
47 66
30 60
30 50
12 17
15 14
16 19
21 48
50 30
51 42
50 15
48 21
12 38
15 56
29 39
54 38
55 57
67 41
10 70
6 25
65 27
40 60
70 64
64 4
36 6
30 20
20 30
15 5
50 70
57 72
45 42
38 33
50 4
66 8
59 5
35 60
27 24
40 20
40 37

160

10000

10000
10000
10000
10000
10000
10000
10000
10000
10000

18
26
11
30
21
19
15
16
29
26
37
16
12
31
8
19
20
13
15
22
28
12
6
27
14
18
17
29
13
22
25
28
27
19
10
12
14
24
16
33
15
11
18
17
21
27
19

20
5
22
12
19
22
16
7
26
14
21
24
13
15
18
11
28
9
37
30
10
8
11
3
1
6
10
20

40
40
40
40
40
40
40
40
40
40

Bibliographies

- [1]
- TADUNFOCK TETI Bernard et FOTSO Laure Pauline « Heuristiques du problème du voyageur de commerce », CARI06- Volume 1 -2006, www.cari-info.org/actes2006/144.pdf.
- [2]
- J. GREFENSTETTE, R. GOPAL, B. ROSIMAITA, D. V. GUCHT, «Genetic Algorithms for the Traveling Salesman Problem », Proceedings of an International Conference on Genetic Algorithms and their Applications, Carnegie Mellon publishers, 160-168, 1985.
- [3]
- F. GLOVER, « Tabu Search-Part II», ORSA Journal on Computing, no 2(1) :4-32, 1990.
- [4]
- KELD HELSGAUN, «An effective implementation of the Lin-Kernighan traveling salesman heuristic », European Journal of Operations Research, no 12 :106-130, 2000.
- [5]
- DAVID S. JOHNSON, LYLE A. MCGEOCH, E.E. ROTHBERG, « Asymptotic Experimental Analysis for the Held-Karp Traveling Salesman Bound », in Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms, Atlanta, Georgia, January 28-30, 341-350, 1996.
- [6]
- S. KIRKPATRICK, C. D. GELAIT, M.P. VECCHI, « Optimization by Simulated Annealing », Science, no 671-680, 1983.
- [7]
- BENAYACHE Antara et Bouremda tabar « Résolution de Problème de Localisation et de Routage avec Capacité à Base de la méthode GRASP », p 4, Mémoire de fin d'étude pour l'obtention du Diplôme Master en Informatique Université - Jijel, Promotion 2015
- [8]
- GUEMRI W, et SEMOUD R, « Résolution de Problème de Transport à la Demande DARP Multi-objectif à Base de la Méthode de Recherche Taboue », p 66, Mémoire de fin d'étude pour l'obtention du Diplôme Master en Informatique, Université de Jijel, juin 2013.
- [9]
- OSMAN, I H, et LAPORTE G. « Metaheuristics: A bibliography ». Ann Oper Res. Vol 63, N° 5,pp.513-623, 1996.

[10]

GHERBOUDJ A.« Méthodes de résolution de problèmes difficiles académiques», Thèse Pour l'obtention du diplôme de Doctorat 3ème cycle LMD en Informatique, 2013.

[11]

KIRKPATRICK S, GELAIT CD, et VECCHI M P, « Optimization by simulated annealing ». Science, 220:671-680, 1983.

[12]

LUKE S, « Essentials of Metaheuristics», December 2010.

[13]

HANSEN.P «The steepest ascent mildest descent heuristic for combinatorial programming » présenté au Congress on Numerical Methods in Combinatorial Optimization, Capri, Italie, 1986.

[14]

GLOVER F.et LAGUNA M. "Tabu search". Kluwer Academic Publishers, Norwell, MA, second edition, 1997.

[15]

Feo T, Resende M, et Smith S, A greedy randomized adaptive search procedure for maximum independent set». Operations Research, 8,67-71,1989.

[16]

Maria J. Blesa, Christian Blum, Paola Festa - 2013 - "Hybrid Metaheuristics": 8th International Workshop, HM 2013, Ischia, Italy.

[17]

GLOVER F. "Principles and Strategies of Tabu Search", OptTek Systems, Boulder, Colorado (USA)

[18]

Lozano, M., Glover, F., Garcia-Martinez, C., Rodriguez, F., Marti R. (2014) "Tabu Search with SO for the Quadratic Minimum Spanning Tree," IIE Transactions 46 (4), pp. 414-428.

[19]

Wu, Q., Y. Wang and F. Glover "Advanced algorithms for bipartite boolean quadratic programs guided by tabu search, strategic oscillation and path relinking," submitted to the INFORMS Journal on Computing (2016).

[20]

Oualid, G. "Multi-Flux Strategic Oscillation Method for the Uncapacitated Facility Location Problem," 18th Annual Congress of the French Society for Operational Research and Decision Support, Metz, France (2017).

[21]

Ruiz R. and Stützle T. "A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem," European Journal of Operational Research, 177(3), pp. 2033-2049(2008).

[22]

Saïd Hanafi, Oualid Guemri, Fred Glover, Igor Crévits. Simple Multi-Wave Algorithm for the uncapacitated Facility Location Problem. Research report November 2016, LAMIH CNRS UMR 8201, France.

[23]

Burke E K, Newall J P, et Wear RF, « A Simple Heuristically Guided Search for the Timetable Problem », proceeding of the international ICSC Symposium on Engineering og Intelligent systems, pp. 574-579 1998.

[24]

Colomi A, Dorigo M, Maniezzo V, et Trubian, M. « Distributed Optimization by Ant Colonies ». Proceedings of the 1st European Conference on Artificial Life. pp 134-142, Elsevier Publishing, 1991.

[25]

Gherboudj A. « Méthodes de résolution de problèmes difficiles académiques », Thèse Pour l'obtention du diplôme de Doctorat 3ème cycle LMD en Informatique, 2013.

[26]

<http://fr.wikipedia.org/wiki/Métaheuristique>.

[27]

Taillard E. « Principes d'implémentation des métaheuristiques », Optimisation approchée en recherche opérationnelle. Hermès, 2002.

[28]

Belkhelladi K, « Stratégies d'échange d'informations dans un système de calcul distribué pour l'optimisation des problèmes combinatoires», thèse pour obtenir le titre de docteur en sciences, Université d'Angers, 2010.

[29]

Bard F, Huang L, Dror M, et Jaillet P, « A branch and cut algorithm for the vrp with satellite facilities ». *IIE Transactions*, 30: 821 - 834, 1998.

[30]

Kechmane.L, Benayad.N, Baalal.A , «A Memetic Algorithm for the Capacitated Location-Routing Problem» Article in *International Journal of Advanced Computer Science and Applications* · July 2016

[31]

Christophe.D, Philippe L, Christian.P, Caroline. « A GRASP×ELS approach for the capacitated location-routing problem “ article in *ScienceDirect Computers & Operations Research* july 2009.

[32] Rui Borges.L , Carlos.F, Beatriz.S «A simple and effective evolutionary algorithm for the capacitated location–routing problem “ article in *ScienceDirect Computers & Operations Research* june 2019.

[33]

Abbas J & Rashed S. A « Heuristic Based approach to Solve a Capacitated Location-routing Problem»

[34]

C. prodhon « LE PROBLÈME DE LOCALISATION-ROUTAGE » Thèse de doctorat de l'Université de Technologie de Troyes. 2006

[35] P. Toth and D. Vigo, editors. *The vehicle routing problem*. Society for Industrial and Applied Mathematics, 2001.

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

[37]

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