



HAL
open science

Prise en compte d'objectifs de stabilité pour l'organisation de collectes de déchets

Frédérique Baniel

► **To cite this version:**

Frédérique Baniel. Prise en compte d'objectifs de stabilité pour l'organisation de collectes de déchets. Génie des procédés. Institut National Polytechnique de Toulouse - INPT, 2009. Français. NNT : 2009INPT042H . tel-01067163v2

HAL Id: tel-01067163

<https://theses.hal.science/tel-01067163v2>

Submitted on 17 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par *Institut National Polytechnique de Toulouse*
Discipline ou spécialité : *Systèmes Industriels*

Présentée et soutenue par *Frédérique BANIEL*
Le 12 Novembre 2009

Titre : *Prise en compte d'objectifs de stabilité pour l'organisation de collectes de déchets*

JURY

Gilles GONCALVES Professeur à l'Université d'Artois
Philippe LACOMME Maître de Conférence à l'Université de Clermont-Ferrand
Dominique FEILLET Professeur à l'École des Mines de Saint-Etienne
Florence PELLETIER Directrice du pôle Environnement de la CAM
Bernard GRABOT Professeur à l'ENIT

Ecole doctorale : *EDSYS*
Unité de recherche : *LGP/ENIT - MOGISA/LAAS*
Directeur(s) de Thèse : *Marie-José HUGUET / Thierry VIDAL*
Rapporteurs : *Gilles GONCALVES / Philippe LACOMME*

Table des matières

1	Les Problèmes de Tournées de Véhicules	13
1.1	Introduction	13
1.2	Formulation du problème	14
1.3	Les extensions du VRP	16
1.3.1	Le Problème de Tournées de Véhicules avec Fenêtres de Temps	16
1.3.2	Le Problème de Tournées de Véhicules Périodique	17
1.3.3	Problèmes de Tournées de véhicules avec livraison et retour chargé	18
1.3.4	Problème Général de Collecte et de Livraison	18
1.3.5	Problème de Trajet à la Demande	18
1.3.6	Problème de Tournées de Véhicules avec collecte partagée	19
1.3.7	Problème de Routage sur un Horizon	19
1.3.8	Problème de Tournées de Véhicules avec Flotte Hétérogène	19
1.4	Le Problème de VRP en environnement incertain	19
1.5	Le Problème de Tournées avec capacité sur les Arcs	21
1.6	Résolution des problèmes de VRP	22
1.6.1	Optimisation difficile	22
1.6.2	Les méthodes exactes pour le VRP	23
1.6.3	Méthodes heuristiques pour le VRP	24
1.7	Problèmes de décision multi-objectifs	36
1.7.1	Définition	36
1.7.2	VRP et multi-objectif	39
1.8	Conclusion	40
2	Collecte des déchets	41
2.1	Types de collecte	41
2.2	État de l'art sur la collecte des déchets	42
2.3	Contexte de l'étude	49
2.3.1	Partenariat	49
2.3.2	Fonctionnement général de la collectivité locale étudiée	49
2.3.3	Objectifs pour l'organisation des collectes	52

2.3.4	La stabilité des tournées de collecte	52
2.3.5	Deux problématiques de construction de tournées	53
2.4	Conclusion	54
3	Étude bi-objective de la stabilité hebdomadaire	56
3.1	Description du problème	56
3.2	Modèle mathématique	58
3.2.1	Fréquence de collecte	58
3.2.2	Modélisation de la stabilité	59
3.3	Hypothèse 1 : $ Z_{c_1} < Z_{c_2} $	61
3.3.1	Méthodes proposées	61
3.3.2	Expérimentations	68
3.3.3	Résultats et conclusion	69
3.3.4	Métaheuristiques multi-objectifs	80
3.4	Hypothèse 2 : $ Z_{c_1} > Z_{c_2} $	88
3.4.1	Principe de résolution	88
3.4.2	Expérimentations	91
3.4.3	Résultats et conclusion	92
3.5	Conclusion	101
4	Étude bi-objective de la stabilité dynamique	103
4.1	Description du problème	103
4.2	Modèle mathématique	104
4.3	Étude de l'insertion de nouveaux points de collecte	105
4.3.1	Méthodes d'insertion	105
4.3.2	Expérimentations	109
4.3.3	Résultats et conclusion	110
4.4	Augmentation de la charge à collecter	120
4.4.1	Méthode d'absorption	120
4.4.2	Expérimentations	121
4.4.3	Résultats et conclusion	122
4.5	Discussion sur l'évolution continue des tournées	132
4.6	Conclusion	133
A	Résultats pour la stabilité hebdomadaire	148
A.1	Résultats hypothèse 1 : $ Z_{c_1} < Z_{c_2} $	148
A.1.1	Résultats en termes de coût	148
A.1.2	Résultats en termes de stabilité globale	149
A.1.3	Résultats en termes de stabilité du point de vue usagers	151
A.1.4	Résultats en termes de stabilité du point de vue employés	152

A.2	Résultats hypothèse 2 : $ Z_{c_1} > Z_{c_2} $	154
A.2.1	Résultats en termes de coût	154
A.2.2	Résultats en termes de stabilité globale	154
A.2.3	Résultats en termes de stabilité du point de vue usagers	155
A.2.4	Résultats en termes de stabilité du point de vue employés	155
B	Résultats pour la stabilité dynamique	157
B.1	Résultats Insertion de nœuds :	157
B.1.1	Résultats en termes de coût	157
B.1.2	Résultats en termes de stabilité globale	158
B.1.3	Résultats en termes de stabilité du point de vue usagers	159
B.1.4	Résultats en termes de stabilité du point de vue employés	159
B.2	Résultats Augmentation de la quantité de nœuds :	161
B.2.1	Résultats en termes de coût	161
B.2.2	Résultats en termes de stabilité globale	161
B.2.3	Résultats en termes de stabilité du point de vue usagers	162
B.2.4	Résultats en termes de stabilité du point de vue employés	163

Table des figures

1.1	Un problème de tournées de véhicules	16
1.2	Méthode avec économies	25
1.3	Méthode d'insertion de Mole et al.	26
1.4	Voisinages agissant sur plusieurs tournées	28
1.5	Mimima locaux et minimum global	29
1.6	Récapitulatif des méthodes de résolution	35
1.7	Front Pareto	36
3.1	Schéma représentant la collecte si $Z_{c_1} < Z_{c_2}$	61
3.2	Suppression des nœuds collectés en c_1	62
3.3	Transfert des noeuds sur tournées restantes	63
3.4	Déplacement d'un nœud dans la méthode 3	64
3.5	Construction d'une tournée avec les nœuds en c_1	66
3.6	Méthode 5 pour $Z_{c_1} < Z_{c_2}$	67
3.7	Instances de Solomon à 100 clients	68
3.8	Comparaison des méthodes en termes de coût par type d'instances	70
3.9	Comparaison des méthodes en termes de stabilité globale par type d'instances	72
3.10	Comparaison des méthodes en termes de stabilité globale par type d'instances	73
3.11	Comparaison des méthodes en termes de stabilité globale par type d'instances	74
3.12	Comparaison des méthodes en termes de stabilité "usagers" par type d'instances	75
3.13	Comparaison des méthodes en termes de stabilité "employés" par type d'instances	77
3.14	Comparaison des méthodes en termes de stabilité "employés" par type d'instances	78
3.15	Comparaison des méthodes en termes de stabilité "employée" par type d'instances	79
3.16	Fronts Pareto obtenus par une recherche tabou et un recuit simulé sur l'instance C101	83
3.17	Résultats comparatifs heuristiques-métaheuristiques	85
3.18	Résultats comparatifs heuristiques-métaheuristiques	86
3.19	Résultats comparatifs heuristiques-métaheuristiques	87
3.20	schéma représentant la collecte si $Z_{c_1} > Z_{c_2}$	88
3.21	Méthode 1 pour $Z_{c_1} > Z_{c_2}$	89
3.22	Méthode 2 pour $Z_{c_1} > Z_{c_2}$	90

3.23	Comparaison des méthodes en termes de coût par type d'instances	93
3.24	Comparaison des méthodes en termes de stabilité globale par type d'instances	94
3.25	Comparaison des méthodes en termes de stabilité globale par type d'instances	95
3.26	Comparaison des méthodes en termes de stabilité globale par type d'instances	96
3.27	Comparaison des méthodes en termes de stabilité "usagers" par type d'instances	97
3.28	Comparaison des méthodes en termes de stabilité "employés" par type d'instances	98
3.29	Comparaison des méthodes en termes de stabilité "employés" par type d'instances	99
3.30	Comparaison des méthodes en termes de stabilité "employés" par type d'instances	100
4.1	Méthode I1	106
4.2	Comparaison des méthodes en termes de coût par type d'instances	111
4.3	Comparaison des méthodes en termes de stabilité globale par type d'instances	112
4.4	Comparaison des méthodes en termes de stabilité globale par type d'instances	113
4.5	Comparaison des méthodes en termes de stabilité globale par type d'instances	114
4.6	Comparaison des méthodes en termes de stabilité "usagers" par type d'instances	116
4.7	Comparaison des méthodes en termes de stabilité "employés" par type d'instances	117
4.8	Comparaison des méthodes en termes de stabilité "employés" par type d'instances	118
4.9	Comparaison des méthodes en termes de stabilité "employés" par type d'instances	119
4.10	Méthode A1	120
4.11	Comparaison des méthodes en termes de coût par type d'instances	123
4.12	Comparaison des méthodes en termes de stabilité globale par type d'instances	124
4.13	Comparaison des méthodes en termes de stabilité globale par type d'instances	125
4.14	Comparaison des méthodes en termes de stabilité globale par type d'instances	126
4.15	Comparaison des méthodes en termes de stabilité "usagers" par type d'instances	128
4.16	Comparaison des méthodes en termes de stabilité "employés" par type d'instances	129
4.17	Comparaison des méthodes en termes de stabilité "employée" par type d'instances	130
4.18	Comparaison des méthodes en termes de stabilité "employés" par type d'instances	131

Liste des tableaux

2.1	Récapitulatif de quelques exemples de collecte des déchets	46
2.2	Récapitulatif de quelques exemples de collecte des déchets	47
2.3	Récapitulatif de quelques exemples de collecte des déchets	48
2.4	Les différentes communes de la CAM	50
A.1	Résultats des méthodes en termes de coût par type d’instances	148
A.2	Résultats des méthodes en termes de stabilité globale pour les instances de type 1	149
A.3	Résultats des méthodes en termes de stabilité globale pour les instances de type 2	150
A.4	Résultats des méthodes en termes de stabilité “usagers” par type d’instances	151
A.5	Résultats des méthodes en termes de stabilité “employés” pour les instances C1 et R1 . .	152
A.6	Résultats des méthodes en termes de stabilité “employés” pour les instances RC1 et C2 .	153
A.7	Résultats des méthodes en termes de stabilité “employés” pour les instances R2 et RC2 .	153
A.8	Résultats des méthodes en termes de coût par type d’instances	154
A.9	Résultats des méthodes en termes de stabilité globale pour les instances de type 1	154
A.10	Résultats des méthodes en termes de stabilité globale pour les instances de type 2	154
A.11	Résultats des méthodes en termes de stabilité “usagers” par type d’instances	155
A.12	Résultats des méthodes en termes de stabilité “employés” pour les instances C1 et R1 . .	155
A.13	Résultats des méthodes en termes de stabilité “employés” pour les instances RC1 et C2 .	155
A.14	Résultats des méthodes en termes de stabilité “employés” pour les instances R2 et RC2 .	156
B.1	Résultats des méthodes en termes de coût par type d’instances	157
B.2	Résultats des méthodes en termes de stabilité globale pour les instances de type 1	158
B.3	Résultats des méthodes en termes de stabilité globale pour les instances de type 2	158
B.4	Résultats des méthodes en termes de stabilité “usagers” par type d’instances	159
B.5	Résultats des méthodes en termes de stabilité “employés” pour les instances C1 et R1 . .	159
B.6	Résultats des méthodes en termes de stabilité “employés” pour les instances RC1 et C2 .	160
B.7	Résultats des méthodes en termes de stabilité “employés” pour les instances R2 et RC2 .	160
B.8	Résultats des méthodes en termes de coût par type d’instances	161
B.9	Résultats des méthodes en termes de stabilité globale pour les instances de type 1	161
B.10	Résultats des méthodes en termes de stabilité globale pour les instances de type 2	162
B.11	Résultats des méthodes en termes de stabilité “usagers” par type d’instances	162

B.12 Résultats des méthodes en termes de stabilité “employés” pour les instances C1 et R1	163
B.13 Résultats des méthodes en termes de stabilité “employés” pour les instances RC1 et C2	163
B.14 Résultats des méthodes en termes de stabilité “employés” pour les instances R2 et RC2	164

Introduction

L'étude menée dans cette thèse porte sur la logistique multi-objectifs des déchets ménagers. Elle s'est déroulée en partenariat avec une collectivité locale, la Communauté d'Agglomération du Muretain (CAM) située près de Toulouse qui nous a soumis ses problématiques. Cette thèse s'est déroulée au sein de deux laboratoires, dans l'équipe de Production Automatisé (PA) du Laboratoire de Génie de Production (LGP) de Tarbes ainsi que dans le groupe de Modélisation d'Optimisation et de Gestion Intégrée de Systèmes d'Activités (MOGISA) du Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) de Toulouse.

Il y a une trentaine d'années, le traitement des déchets se résumait à une simple mise en décharge. Depuis une conscience collective des problématiques environnementales a placé le traitement des déchets au cœur des préoccupations publiques. La gestion qui est faite des déchets ménagers constitue un enjeu environnemental, économique, social et de citoyenneté majeure, directement lié aux modes de consommation et à la société plus généralement. En France, les municipalités ou les collectivités locales, regroupant plusieurs municipalités, ont en charge de collecter et de convoier les déchets vers leur lieu de traitement. Cette chaîne logistique de gestion globale des déchets couvrent plusieurs domaines d'études et de nombreux métiers, comme par exemple le transport, le tri ou encore le recyclage des produits rejetés. C'est une problématique globale complexe qui touche tous les pays. Il existe plusieurs filières de collecte et de traitement des déchets. Il existe par exemple des filières de déchets dits "ménagers" ou des filières de déchets dits "recyclables" dans lesquelles les déchets sont collectés chez les particuliers selon le traitement spécifique à leur appliquer en aval. Les déchets ménagers sont destinés à être détruits tandis que les déchets recyclables sont destinés à être ré-utilisés dans d'autres industries. Il existe également des filières, telles que les déchets "industriels", qui sont collectées séparément des déchets ménagers et assimilés, mais dont les modalités et les conditions de traitement sont les mêmes ; ces déchets ne présentent pas de caractère toxique ou dangereux et la manutention ou le stockage ne nécessite pas de précaution particulière. Les déchets dangereux eux suivent une autre gestion et doivent subir d'autres trajets de collecte et de traitement [39]. D'autres déchets sont collectés en point d'apport volontaire, c'est-à-dire que les usagers regroupent sur des points de rassemblement ce type de déchets comme par exemple le verre ou le carton dans certaines communes. Les collectes sont par la suite organisées selon les points d'apport volontaire en place.

Dans l'étude proposée, nous allons nous intéresser à la partie logistique, et plus précisément à la collecte et au transport des déchets ménagers et assimilés jusqu'à leur point de décharge (déchetteries,

centre de transfert ou centre de traitement directement). Le but étant d'optimiser le transport de déchets tout en gardant une bonne qualité de service. Les enjeux mis en relief pour cette problématique sont les enjeux classiques imposés par toute étude visant à améliorer un service publique : un enjeu économique classique (minimiser les coûts de collecte et de transport), un enjeu environnemental évident (diminuer l'impact environnemental), un enjeu de qualité (satisfaire au mieux les usagers) et enfin un enjeu social (satisfaire les conditions de travail des employés).

Dans cette thèse nous nous concentrons sur l'aspect opérationnel journalier de la gestion globale : la collecte et le transport des déchets. Cet aspect a déjà fait l'objet de nombreuses études et de nombreux développements de modèles mathématiques en recherche opérationnelle. Les modèles utilisés pour ce type de problématique sont basés sur le très connu problème du voyageur de commerce (ou Travelling Salesman Problem : TSP). Ce problème est le premier à avoir été étudié dans la famille des problèmes de transport. Il consiste à organiser un trajet de longueur minimal pour un véhicule en passant une seule fois chez chaque client. Le problème est plus complexe qu'il n'y paraît. En effet, si on analyse tous les parcours possibles, pour n villes, le nombre de possibilités est de $(n-1)!$. Pour 6 villes nous avons 120 possibilités, pour 10 villes plus de 362 000, etc. Le TSP est connu depuis le 19^{ème} siècle mais a fait l'objet de nombreuses études surtout depuis le développement de l'informatique. Il sert de base pour d'autres problèmes typiques de transport. Lorsque celui-ci concerne une planification de trajet pour plusieurs véhicules en même temps qui doivent chacun faire leur circuit en livrant une certaine quantité de produit à chaque client cela se nomme le problème de tournées de véhicules (Vehicle Routing Problem). Ce problème pratique est souvent étudié pour des cas réels et se complexifie avec l'ajout de nombreux paramètres et contraintes, comme par exemple des contraintes d'horaire de passage sur certaines parties du réseau ; ou encore par l'ajout de plusieurs objectifs à optimiser en même temps, comme par exemple les objectifs représentant les coûts, ceux représentant l'équilibrage de tournées, etc.

Notre application étant basée sur une étude de cas, nous avons fait ressortir deux types d'objectifs que nous allons étudier dans cette thèse : l'objectif de coût et l'objectif de stabilité. L'objectif de coût est un objectif classique utilisé dans la plupart des études. L'objectif de stabilité est plus original et va caractériser des nouvelles solutions en les définissant comme plus ou moins "proches" des anciennes solutions. Cet objectif peut être conflictuel avec l'objectif de coût. Une diminution des coûts va impliquer des changements dans les solutions et inversement, une conservation de solutions relativement similaires va impliquer une augmentation des coûts. La modélisation de notre problème se base sur celle utilisée pour les problèmes de tournées de véhicules avec comme objectifs de minimiser les coûts et de maintenir la stabilité des solutions. Le but de notre travail est de développer des méthodes dédiées qui permettent à une collectivité en charge de la gestion des déchets de définir un ensemble de tournées à réaliser pour chaque jour afin de satisfaire au mieux l'objectif de coût et l'objectif de stabilité que nous allons définir et développer dans cette thèse.

Ce mémoire de thèse est articulé en quatre grands chapitres : un premier chapitre est consacré à l'état de l'art, un deuxième chapitre définit plus précisément le problème au cœur de ce mémoire et les chapitres trois et quatre sont dédiés respectivement à deux problèmes spécifiques mêlant les objectifs

de coût et de stabilité.

Le premier chapitre est consacré à l'état de l'art sur les problèmes de tournées de véhicules en général. Il est articulé en deux grandes parties : la première partie traite des problèmes de VRP dit classiques et de différentes extensions en abordant également les méthodes de résolution de ce problème et la seconde partie aborde les problèmes multi-objectifs en se focalisant plus précisément sur les problèmes de VRP multi-objectifs.

Le deuxième chapitre s'intéresse à un état de l'art sur les problèmes de collecte des déchets ménagers et notamment à des cas pratiques rencontrés dans la littérature. Une première partie de ce chapitre énumère un ensemble d'études menées dans différents pays en mettant en avant la modélisation et les méthodes de résolution utilisées. Une seconde partie de ce chapitre donne le contexte de ce travail en précisant les problèmes d'organisation des collectes posés par la CAM et les aboutissements souhaités. Dans cette seconde partie, nous exposons notre définition de l'objectif de stabilité.

Le troisième chapitre porte sur l'étude d'un problème de collecte hebdomadaire avec contraintes de coût et de stabilité. Dans la plupart des collectivités, la collecte est étalée sur plusieurs jours et les points de passage sont collectés une à plusieurs fois par semaine. Ces fréquences de collecte doivent être prises en compte et peuvent être différentes d'un point à l'autre. Dans notre cas, nous allons étudier le fait que certains points sont collectés une fois et d'autres deux fois par semaine. L'objet de ce chapitre est donc de proposer des méthodes d'élaboration des tournées pour le début et la fin de semaine de manière à ce que les tournées ne soient pas trop éloignées les unes des autres. Nous proposons de comparer différentes méthodes heuristiques et métaheuristiques en les comparant selon les différents objectifs définis pour en extraire celles qui semblent les plus pertinentes.

Le quatrième chapitre traite d'un problème de collecte dynamique avec contraintes de coût et de stabilité. Dans cette étude, les informations de collecte évoluent au cours du temps (par exemple, le nombre de points de passage peut augmenter ou encore la quantité de déchets à collecter peut varier). Le but est de proposer des méthodes d'intégration de ces nouvelles informations tout en ne s'éloignant pas trop des tournées déjà mises en place. Nous proposons différentes méthodes heuristiques et métaheuristiques en les comparant selon les objectifs définis pour en extraire celles qui nous semblent les plus pertinentes.

Et enfin nous terminerons ce mémoire en présentant une conclusion des travaux menés ainsi que des perspectives d'extension envisagées.

Chapitre 1

Les Problèmes de Tournées de Véhicules

1.1 Introduction

Le Problème de Tournées de Véhicules plus connu sous son nom anglais Vehicle Routing Problem (VRP) est un problème qui a fait l'objet de nombreuses études. Il a été introduit en 1959 par Dantzig [37] et permet de modéliser ainsi que de résoudre des problèmes de transport opérationnel. Il est l'intersection de deux problèmes difficiles également bien connus :

- le problème du voyageur de commerce, ou TSP pour Travelling Salesman Problem qui consiste à organiser un circuit pour visiter une et une seule fois un nombre prédéfini de clients de manière optimale, c'est-à-dire généralement en minimisant les coûts [126], représentant généralement la distance de parcours du circuit,
- et le problème du bin-packing où il s'agit de trouver le rangement le plus économique possible d'un ensemble d'articles dans des boîtes [43].

Le but de la résolution d'un VRP consiste à concevoir un ensemble de tournées, partant et revenant d'un même dépôt et en passant par un ensemble de points (souvent nommés clients) prédéfinis. Chaque point de passage doit être visité mais chacun d'entre eux ne le sera que par une seule tournée. Le critère d'optimisation concerne généralement la distance totale parcourue par les véhicules, mais on peut également rencontrer : le nombre de véhicules utilisés, le temps total de parcours, l'équilibrage de charge entre les véhicules etc. Le problème de VRP se modélise généralement à l'aide d'un graphe dans lequel les arcs représentent les axes routiers et les nœuds les points de passage des véhicules. Ce problème est connu pour être *NP*-difficile [95] c'est à dire qu'il n'existe pas d'algorithme en temps polynomial pour le résoudre.

On associe généralement à chaque point de passage une quantité (à collecter ou à livrer) et/ou un temps de service. De plus les véhicules sont souvent limités en termes de capacité. La capacité d'un véhicule représente la quantité de produit maximum pouvant être contenu dans le véhicule. Ce problème se nomme problème de tournées de véhicules avec contrainte de capacité (Capacited Vehicle Routing Problem ou CVRP) et consiste à livrer (ou collecter) une charge sur les points de passage à l'aide d'une flotte de véhicules dont la capacité est limitée. Ce type de problème est souvent confondu avec

le problème de base qui est celui du VRP. En effet, dans les cas pratiques, les tournées correspondent à une livraison (ou une collecte) avec une certaine demande sur chaque point de passage et doivent donc prendre en compte la contrainte de capacité à respecter par chaque véhicule, on rencontre très rarement des problèmes sans contrainte de capacité.

Ainsi les contraintes usuelles de base du VRP ou CVRP sont les suivantes :

- tous les clients doivent être visités,
- un client ne sera servi qu'une unique fois,
- un véhicule qui arrive chez un client doit en repartir,
- la capacité des véhicules doit être respectée.

Sur ce problème de base viennent se greffer d'autres contraintes permettant de prendre en compte des extensions à ce premier problème et donc modéliser différents problèmes d'optimisation. Nous verrons dans la suite de ce chapitre quelques unes de ces extensions.

1.2 Formulation du problème

Une formulation mathématique du problème de CVRP a été proposée par Fisher et al. [55, 56]. Elle se base sur un graphe : $G = (Z, A)$ où :

- $Z = (z_0, z_1, z_2, \dots, z_{n-1})$ est l'ensemble des n sommets (nœuds) du graphe représentant l'ensemble des clients, avec z_0 représentant le dépôt,
- $A = \{(z_i, z_j) | z_i, z_j \in Z; i \neq j\}$ est l'ensemble des arcs orientés représentant le trajet entre deux sommets.

Soient

- $K = \{1, \dots, m\}$: l'ensemble des véhicules disponibles ;
- Q : la capacité maximale des véhicules ; elle est identique pour tous les véhicules (véhicule homogène) ;
- q_i : la capacité à collecter sur un nœud z_i ;
- d_{ij} : la distance de parcours de l'arc (z_i, z_j) .

Les variables de décision utilisées dans cette formulation sont :

- $y_i^k = 1$ si le nœud z_i est visité par le véhicule k et 0 sinon.
- $x_{ij}^k = 1$ si l'arc (z_i, z_j) est dans la tournée du véhicule k et 0 sinon.

Les contraintes du CVRP s'expriment alors de la façon suivante :

$$\min(\sum_{i \in Z} \sum_{j \in Z} \sum_{k \in K} x_{ij}^k \cdot d_{ij}) \quad (1.1)$$

$$\sum_{i=1}^n q_i y_i^k \leq Q \quad (k = 1, \dots, m) \quad (1.2)$$

$$\sum_{k \in K} y_i^k = 1 \quad (i = 1, \dots, n - 1) \quad (1.3)$$

$$\sum_{k=1}^m y_i^k = m \quad (i = 0) \quad (1.4)$$

$$\sum_{i \in Z} x_{ij}^k = y_j^k \quad (j = 0, \dots, n-1); \quad (k = 1, \dots, m) \quad (1.5)$$

$$\sum_{j \in Z} x_{ij}^k = y_i^k \quad (i = 0, \dots, n-1); \quad (k = 1, \dots, m) \quad (1.6)$$

$$\sum_{i,j \in X} x_{ij}^k \leq |X| - 1 \quad (\forall X \subset Z, \quad 2 \leq |X| \leq n-2); \quad (k = 1, \dots, m) \quad (1.7)$$

$$x_{ij}^k \in \{0, 1\} \quad (i, j = 0, \dots, n-1; \quad i \neq j); \quad (k = 1, \dots, m) \quad (1.8)$$

$$y_i^k \in \{0, 1\} \quad (i = 0, \dots, n-1); \quad (k = 1, \dots, m) \quad (1.9)$$

Ce modèle mathématique exprime le fait que l'on cherche à déterminer un ensemble de tournées tout en minimisant la distance totale parcourue (1.1). Les contraintes (1.2) expriment le fait que la capacité des véhicules est limitée. Les contraintes (1.3 et 1.4) établissent que chacun des clients n'est desservi qu'une seule fois (ie. par un seul véhicule) sauf pour le sommet dépôt par lequel tous les véhicules doivent passer. Les deux contraintes suivantes (1.5 et 1.6) établissent la cohérence du parcours : effectuer un trajet d'un nœud à un autre à l'aide d'un véhicule donné revient à dire que ce véhicule est utilisé pour collecter les nœuds en question. Les contraintes (1.7) permettent d'éliminer des sous-tours c'est à dire d'éliminer des solutions ne passant pas par l'ensemble des clients à visiter. La binarité des variables de décision est donnée par les contraintes (1.8 et 1.9). Une solution d'un VRP est représentée par la figure (1.1).

D'autres contraintes peuvent venir s'ajouter à ce premier modèle. En effet, la littérature dénombre de nombreuses extensions à ce problème de base comme l'ajout de fenêtre de temps ou l'utilisation de véhicules de capacité différente. Chacune de ces extensions représente un problème particulier provenant souvent d'un problème pratique. Dans certains cas, il faut cumuler plusieurs extensions pour avoir la modélisation complète du problème considéré et ainsi étudier des problèmes plus complexes puisque prenant en compte un plus grand nombre de contraintes. Par exemple, l'ajout de fenêtres de temps n'est pas rare dans une application, ainsi que l'utilisation de véhicules avec des capacités différentes.

Un autre type d'extension apparaît lorsque les informations associées au problème de tournées évoluent dans le temps.

Dans la suite de ce chapitre, nous allons voir plus en détails certaines extensions. Dans un premier temps les extensions correspondant à des ajouts de contraintes au modèle de base puis dans un second temps les extensions considérant la dynamique des informations.

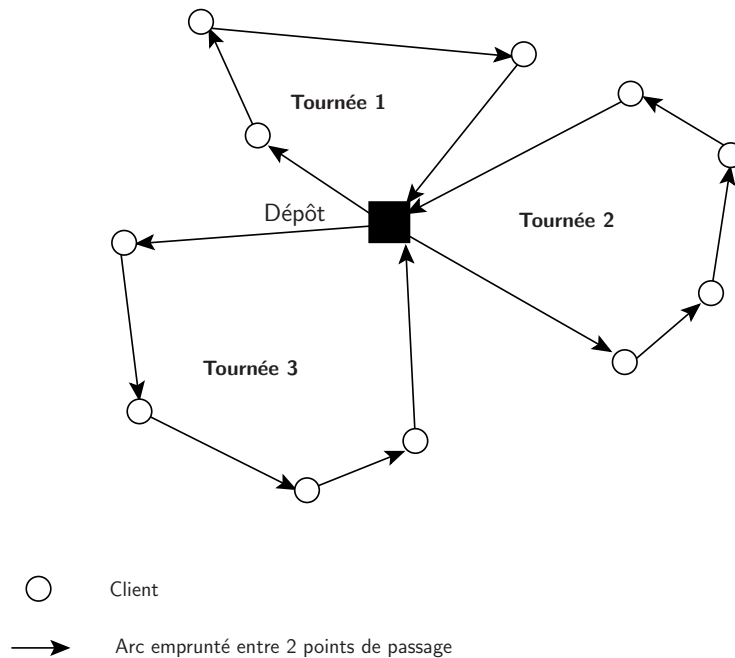


FIG. 1.1 – Un problème de tournées de véhicules

1.3 Les extensions du VRP

1.3.1 Le Problème de Tournées de Véhicules avec Fenêtres de Temps

Plus connu sous son nom anglais, le Vehicle Routing Problem with Time Windows (VRPTW) est l'une des extensions les plus étudiées. Le problème de base est celui du VRP auquel on ajoute une contrainte sur l'horaire de passage à certains nœuds. Ces contraintes horaires traduisent des limites sur la date d'arrivée au plus tôt et sur la date de départ au plus tard en certains nœuds du réseau. Cette extension représente de nombreux cas concrets de gestion de tournées de véhicules. Dans la littérature, ce problème est traité de deux manières : soit en considérant que les fenêtres temporelles sont strictes et doivent donc être absolument respectées, soit en considérant que les fenêtres temporelles sont souples et peuvent donc ne pas être respectées. Des "pénalités de retard" sont alors généralement associées à la fonction coût traduisant le non respect de ces fenêtres.

La formulation d'un VRPTW a été introduite par Solomon [129] et fait intervenir les notations suivantes (en plus de celles utilisées dans le modèle de base de CVRP) :

- $[a_i, b_i]$: la fenêtre de temps associée au nœud z_i avec a_i représentant la date d'arrivée au plus tôt dans le nœud z_i et b_i représentant la date de fin de service au plus tard pour le nœud z_i ;
- t_i : le temps de service du nœud z_i ;
- t_{ij} : le temps de transport du nœud z_i au nœud z_j ;
- u_i^k : la variable de décision temporelle représentant l'heure d'arrivée du véhicule k dans le nœud z_i ;
- M : une grande valeur.

Les contraintes temporelles à ajouter au modèle du CVRP sont les suivantes :

$$a_i y_i^k \leq u_i^k \leq b_i y_i^k \quad (i = 0, \dots, n-1); \quad (k = 1, \dots, m) \quad (1.10)$$

$$u_i^k + t_i + t_{ij} - M(1 - x_{ij}^k) \leq u_j^k \quad (i = 0, \dots, n-1); \quad (k = 1, \dots, m) \quad (1.11)$$

La contrainte (1.10) permet de vérifier que les nœuds sont collectés dans leur fenêtre de temps respective pour un véhicule donné. Lorsqu'un véhicule ne collecte pas un nœud, ces contraintes forcent la variable temporelle à être nulle. Les contraintes (1.11) traduisent la succession entre la collecte de deux sommets z_i et z_j : si z_j est collecté après z_i par un même véhicule k alors le début de la collecte de z_j ne peut se faire tant que la collecte de z_i ne s'est pas terminée et tant que le trajet entre ces deux nœuds n'a pas été effectué. Dans ces contraintes, la valeur M représente un grand nombre et permet d'assurer le respect de la contrainte lorsque le sommet z_j ne succède pas au sommet z_i dans la tournée du véhicule k .

La prise en compte d'une limite sur la durée de chaque tournée (ie. un temps de travail), notée T peut se faire en introduisant les contraintes ci-dessous (1.12) :

$$\sum_{i=1}^n t_i y_i^k + \sum_{i=1}^n \sum_{j \neq i, j=1}^n x_{ij}^k t_{ij} \leq T \quad (k = 1, \dots, m) \quad (1.12)$$

1.3.2 Le Problème de Tournées de Véhicules Périodique

Le Periodic Vehicle Routing Problem (PVRP) généralise le VRP en considérant que les points de passage peuvent être desservis plusieurs fois sur un horizon de planification. Pour chaque point de passage, on dispose d'une fréquence de collecte et d'un ensemble de combinaisons possibles de collecte. Pour résoudre un PVRP, il faut déterminer simultanément une combinaison de jour de collecte pour chaque sommet et l'ensemble des tournées pour chacun des jours de l'horizon de planification. Ces deux étapes peuvent être faites simultanément ou l'une après l'autre. Une formulation du problème a été proposée par Beltrami et al. [15] et Russell et al. [125]. Ce type de problème se rencontre fréquemment pour les collectes des déchets.

La modélisation du PVRP utilise les paramètres complémentaires suivants :

- f_i : la fréquence de collecte du sommet z_i ,
- R_i : la combinaison des jours possibles pour le passage sur le sommet z_i .

Plusieurs classifications des problèmes de PVRP ont été proposées par Mourgaya et al. [104]. Dans cet article, les auteurs classent les problèmes de PVRP selon les objectifs, comme la distance de trajet, le temps de trajet, le nombre de véhicules, l'équilibrage des charges, la qualité de service ou encore la régionalisation des trajets (regroupement des points de passage en cluster). Ils proposent de classer également selon les différentes contraintes ajoutées, c'est-à-dire en ce qui concerne (i) la planification des visites (visite unique ou multiple), (ii) le type de demande (identique ou cumulative en fonction du temps ou d'un taux) ou encore (iii) la disponibilité des véhicules. Ils mettent en évidence le fait que ce

problème de tournées de véhicules est tactique (choisir les jours de collecte) et opérationnel (construire les tournées). Francis et al. [57] introduisent dans leur étude un niveau de décision stratégique consistant à prendre en compte la fréquence de collecte comme une variable de décision.

Un exemple de PVRPTW (Periodic Vehicle Routing Problem with Time Windows) est donné dans l'article de Cordeau et al. [36]. Dans cet article, les auteurs proposent de nouvelles instances pour traiter des problèmes de tournées de véhicules et notamment celui de PVRPTW en définissant pour chaque client une fréquence de passage et les jours possibles de visite.

Les résolutions pour ce type de problème sont soit séparées en deux niveaux (affectation des jours de collecte et création des circuits), soit elles sont intégrées (l'affectation des jours et la création des circuits sont établies simultanément).

1.3.3 Problèmes de Tournées de véhicules avec livraison et retour chargé

Ce problème appelé Vehicle Routing Problem with Backauls (VRPB) est une extension du VRP incluant deux types de nœuds : des nœuds de livraison et des nœuds de collecte. Dans un VRPB les points de livraison vont recevoir une quantité de produit provenant du dépôt et les nœuds de collecte vont prélever une quantité de produit ramenée au dépôt.

Le point critique pour ce problème est d'organiser en premier les livraisons pour "vider" le contenu du véhicule pour pouvoir collecter les nouveaux produits. La somme des produits à livrer ainsi que la somme des produits à collecter ne doit pas excéder séparément la capacité du véhicule utilisé. La résolution de ce problème consiste à commencer les tournées par les clients à livrer et à les terminer par ceux à collecter. Cette contrainte s'explique par le fait qu'il est très difficile de réarranger les charges vu que les véhicules partent du dépôt chargé au maximum et peuvent vite être en surcharge, c'est ainsi qu'en pratique, les clients à livrer ont une plus forte priorité que ceux à collecter [138].

1.3.4 Problème Général de Collecte et de Livraison

Le problème général de collecte et de livraison (Pickup and Delivery Problem ou PDP) est à rapprocher du problème de VRPB. La différence réside dans le fait que certains produits vont être collectés et redistribués dans la même tournée [40] sans passer par le dépôt. Pour cela, la capacité du véhicule doit être connue, ainsi que les quantités de produit à redistribuer et les points d'origine et de livraison de chaque produit. Le VRP classique peut être donc considéré comme un PDP dans lequel l'ensemble des origines se situe au dépôt ou bien l'ensemble des livraisons, selon le problème traité.

Ce problème peut présenter certaines contraintes additionnelles, comme par exemple la contrainte LIFO, c'est-à-dire que le produit collecté en dernier doit être redistribué en premier.

1.3.5 Problème de Trajet à la Demande

Le DARP (Dial a Ride Problem) est une extension du PDP (pick up and delivery problem) où les marchandises sont des personnes à prendre et à déposer [35]. Dans ce type de problème les usagers fixent

leur point de ramassage et leur point de dépôt. Les usagers peuvent également spécifier un créneau horaire pour le ramassage et/ou le dépôt. En général ils n'en spécifient qu'un des deux. Ce type de problèmes est de plus en plus étudié en raison de la croissance du transport à la demande mis en place dans certaines agglomérations. Il est souvent étendu à un problème en ligne qui permet de gérer les demandes des usagers en dynamique [84].

1.3.6 Problème de Tournées de Véhicules avec collecte partagée

Ce problème, autrement appelé Split Delivery Vehicle Routing Problem (SDVRP) permet de satisfaire une demande d'un client par plusieurs tournées. Il a été introduit par Dror [47]. Plusieurs véhicules correspondant à des tournées différentes peuvent passer par un même nœud ; ainsi la demande d'un client peut être fractionnée et un client peut donc être visité par plusieurs véhicules.

1.3.7 Problème de Routage sur un Horizon

Depuis une vingtaine d'années, ce problème (Inventory Routing Problem ou IRP) suscite aussi beaucoup d'intérêt. Il s'agit d'une approche plus globale que celle du PVRP. Le problème consiste à distribuer un produit unique à partir d'un dépôt central vers plusieurs clients sur un horizon de planification de plusieurs périodes. Chaque client consomme le produit suivant un taux de consommation donné et une capacité d'entreposer localement le produit dans un réservoir. La distribution se fait au moyen d'un parc de véhicules identiques. Le problème consiste à déterminer, pour chaque période de l'horizon de planification, les clients à visiter, la quantité à livrer et le trajet à suivre pour chaque camion de façon à minimiser les coûts de distribution. Ceci est typiquement adapté au problème de transport de fuel.

1.3.8 Problème de Tournées de Véhicules avec Flotte Hétérogène

Le problème de tournées de véhicules avec flotte hétérogène (Heterogeneous fleet Vehicle Routing Problem ou HVRP) est une extension du CVRP : à chaque véhicule k correspond une capacité de collecte notée Q_k . Les premières études de ce problème ont été menées par Golden et al. [69]. Pour résoudre ce problème, en plus de la construction des tournées il est nécessaire de déterminer le nombre de véhicules de chaque type à utiliser. Deux variantes de ce problème ont été traitées dans la littérature : le HVRP avec flotte de véhicules limitée et le HVRP avec flotte de véhicules illimitée.

1.4 Le Problème de VRP en environnement incertain

Le problème de tournées de véhicules dynamique ou Dynamic Vehicle Routing Problem (DVRP) est une des extensions au problème de VRP des plus récentes. Elle se différencie des extensions précédentes par le fait qu'elle n'introduit pas de nouveaux types de contraintes mais fait évoluer la modélisation du problème.

Les avancées récentes en matière de télécommunications et des technologies d'information associées donnent la possibilité de pouvoir accéder à de nombreuses informations très rapidement, d'où l'émergence actuelle des problèmes de tournées dynamiques. Ces problèmes concernent l'évolution des données au cours du temps. Plusieurs degrés de dynamisme ont été définis par Larsen [93] dans sa thèse. Ces degrés dépendent du nombre d'évènements dynamiques qui vont apparaître ou bien être modifiés et de leur distribution temporelle. La distribution temporelle correspond au temps de réaction associé à l'apparition d'un nouvel évènement à intégrer aux tournées déjà en places. Cela peut se résumer en deux grandes familles de DVRP :

- soit un ou plusieurs évènements apparaissent au cours du temps une fois les tournées déjà en place, ils vont ainsi ajouter de nouvelles variables à au modèle de base,
- soit l'information sur un ou plusieurs évènements évolue une fois les tournées déjà en place, c'est-à-dire que les valeurs de certains paramètres du modèle de base changent.

Une classification des systèmes dynamiques peut se faire en trois échelons : faible, modéré et fort. Pour qualifier les problèmes selon ces différents échelons, Larsen [93] propose plusieurs mesures qui permettent d'évaluer le dynamisme d'un système. Ces mesures sont divisées en trois catégories :

- un dynamisme qualifié "sans fenêtres de temps" : dépendant du nombre de clients dynamiques et du nombre de clients total,
- un dynamisme qualifié "effectif" : dépendant des instants où les évènements ont lieu réellement,
- et un dynamisme qualifié "avec fenêtre de temps" : dépendant des fenêtres de temps associés aux clients.

Récemment Larsen et al. [94] ont peaufiné leur classification en trois échelons en la basant non seulement sur les degrés de dynamisme évoqués ci-dessus mais également sur les différents objectifs pris en compte. En effet, les objectifs peuvent être différents et nécessiter différents temps de réaction. Par exemple, les services d'urgence n'ont pas les mêmes besoins de réactivité que les services de distribution de gasoil. Housroum [72], dans sa thèse, répertorie des exemples types pour chaque échelon de systèmes dynamiques.

La formulation de ce type de problème est la même que le VRP classique, seules les données du problème évoluent au cours du temps et sont ainsi remises à jour.

Ce type de problème est à rapprocher des problèmes de tournées de véhicules Stochastiques. En effet, le VRP est dit "stochastique" (**Stochastic Vehicle Routing Problem** ou SVRP) quand une ou plusieurs composantes du problème ne sont pas connues précisément à l'avance afin de refléter l'incertitude qu'il peut exister en pratique sur les valeurs des paramètres. Selon Laporte et al. [90] ainsi que Gendreau et al. [63] le SVRP peut se classer en trois catégories :

- le Problème de Tournées de Véhicules avec clients stochastiques (Vehicle Routing Problem with Stochastic Customers) : pour cela une probabilité est associée à la présence ou non du client,
- le Problème de Tournées de Véhicules avec temps de trajet stochastique (Vehicle Routing Problem with Stochastic Travel Time) : dans ce cas les temps de déplacement (ou de service) sont des

variables aléatoires,

- le Problème de Tournées de Véhicules avec demandes stochastiques (Vehicle Routing Problem with Stochastic Demands) : les demandes des clients correspondent à des variables aléatoires.

Ce dernier est l'un des problèmes stochastiques les plus étudiés, il a été développé par exemple pour la livraison de pétrole, la collecte d'argent dans les banques etc.

1.5 Le Problème de Tournées avec capacité sur les Arcs

Ce type de problème est à rapprocher du VRP puisqu'il adresse également des problèmes de tournées de véhicules. La différence se situe dans le type de tournées effectuées. Pour un VRP la livraison (ou la collecte) s'effectue sur les nœuds, alors que pour le CARP (Capacited Arc Routing Problem) la livraison (ou la collecte) se fait sur les arcs du réseau routier.

L'étude du CARP est plus récente que celle du VRP, les premiers travaux datant des années 1980. Ce problème consiste, à l'aide d'un ensemble de véhicules partant d'un dépôt unique, à parcourir un ensemble d'arêtes d'un réseau représentant le réseau routier avant de revenir à ce même dépôt. Certaines des arêtes du réseau doivent être prises en charge par l'ensemble des véhicules affectés aux tournées. Les demandes des clients sont alors associées aux arêtes. Du point de vue de la modélisation, cela signifie que les clients sont répartis le long des rues où il y a un ensemble de services à réaliser.

Les critères d'optimisation sont les mêmes que ceux utilisés dans le VRP. Comme le VRP, le problème de CARP est connu pour être NP-difficile [48].

Cette modélisation permet de représenter le réseau routier et les trajets effectifs des différents véhicules de manière plus précise que celle basée sur le VRP. Dans les années 2000, différentes extensions du CARP ont été étudiées. La communauté de chercheurs s'est intéressée à pouvoir prendre en compte des contraintes réelles rencontrées notamment dans le cas de réseaux routiers : voies à double sens ou non, sens interdit, interdiction de tourner etc. Ces travaux ont conduit au ECARP (Extended CARP) ou MCARP (Mixed CARP) [87, 14]. Par ailleurs, une extension prenant en compte l'aspect périodique des tournées sur arcs a également été proposée par W. Ramdane-Cherif dans sa thèse et dans Lacomme et al. [120, 88]. Cette extension, appliquée notamment pour de la collecte de déchets, consiste comme pour le cas du VRP périodique, à prendre en compte un horizon de collecte et des fréquences de collecte associées aux différentes rues pour déterminer une affectation des rues aux jours de collecte tout en minimisant le coût total de l'ensemble des tournées sur l'ensemble des périodes étudiées. Une extension récente s'intéresse à l'intégration de contraintes temporelles limitant les horaires de passage sur certaines rues [121]. Pour ces différentes variantes du CARP, les validations expérimentales traitent des problèmes ayant quelques centaines de sommets et d'arêtes.

Des cas pratiques de CARP se rencontrent pour la collecte de déchets, l'entretien de rues ou de réseaux ferrés, la distribution de courriers, l'inspection de lignes électriques, etc.

1.6 Résolution des problèmes de VRP

Dans Golden et al. [68], les auteurs présentent le problème de VRP comme étant facile à expliquer mais difficile résoudre. Le VRP fait partie de la classe des problèmes dit "NP-difficile". Pour ce type de problème, on distingue généralement deux types de méthodes de résolution :

- les méthodes exactes : permettant d'obtenir une solution optimale au problème posé,
- les méthodes approchées : permettant d'obtenir une solution au problème posé sans pouvoir affirmer qu'elle est optimale.

Dans cette partie, nous allons présenter les problèmes d'optimisation en général, puis les types de méthodes de résolution en se focalisant plus particulièrement sur les méthodes dédiées à la résolution de problèmes de VRP.

1.6.1 Optimisation difficile

Optimisation difficile

Les problèmes d'optimisation dont il est possible de trouver une solution en un temps polynomial font partie de la classe P.

L'optimisation difficile traite quant à elle des problèmes de la classe dites "NP" (Non-déterministes Polynomiaux). Ces problèmes sont qualifiés de NP-difficile et sont parmi les problèmes les plus étudiés, tels le VRP [18]. Ceci parce que beaucoup de problèmes applicatifs sont NP-difficiles et que l'on ne sait pas résoudre un problème NP-difficile efficacement. Concrètement, il est "facile" d'évaluer une solution à un tel problème, mais l'exploration de l'espace de recherche pour la résolution du problème croît de manière exponentielle avec la taille des instances. Il n'existe pas, à ce jour, d'algorithme polynomial pour la résolution de ces problèmes. L'utilisation d'algorithmes permettant de trouver une solution approchée en un temps raisonnable est donc courante.

Problème d'optimisation

Un problème d'optimisation au sens général comprend un ensemble de solutions possibles X , dont la qualité peut être décrite par une fonction objectif f . On cherche alors à trouver la solution x^* possédant la meilleure qualité $f(x^*)$ (en minimisant ou en maximisant $f(x)$ selon l'objectif défini). Un problème d'optimisation peut présenter des contraintes sur x , être dynamique si $f(x)$ change avec le temps ou encore multi-objectif si plusieurs fonctions objectifs doivent être optimisées. Il existe des méthodes déterministes dites "exactes" permettant de résoudre certains problèmes en un temps fini de manière optimale, et des méthodes "heuristiques" permettant une optimisation dite "approchée".

Certains problèmes d'optimisation demeurent cependant hors de portée des méthodes exactes. C'est le cas notamment lorsque le temps de calcul pour une méthode exacte croît de manière exponentielle en fonction de la taille des données. Ainsi, à partir d'une certaine taille des données (dépendant du type de problème et de la méthode de résolution) le temps de calcul nécessaire devient prohibitif et ne permet

plus de résoudre de manière exacte le problème posé. Dans de tels cas, le problème d'optimisation est dit "difficile", car aucune méthode exacte n'est capable de le résoudre en un temps "raisonnable". Pour trouver malgré tout une solution, on devra alors faire appel aux méthodes "heuristiques". Cet ensemble de méthodes ne garantit en rien de trouver la meilleure solution, mais a pour avantage de trouver une solution réalisable dans un temps correct. Il existe plusieurs types de méthodes heuristiques et donc plusieurs qualifications possibles de ces méthodes, décrites ci-dessous.

Le problème de VRP a fait l'objet de nombreuses recherches en ce qui concerne les méthodes dédiées pour sa résolution de manière exacte et de manière approchée. Pour le résoudre de manière exacte, les instances traitées ne doivent pas dépasser une centaine de nœuds à traiter, par contre en ce qui concernent les méthodes approchées, les instances considérées peuvent être de grande taille (des dizaines de milliers de nœuds).

1.6.2 Les méthodes exactes pour le VRP

Ces méthodes permettent d'obtenir la solution exacte à un problème donné, c'est-à-dire qu'elles fournissent la solution optimale d'un problème. Selon Laporte et al. [91] les méthodes exactes peuvent être divisées en 3 grandes catégories :

Procédure de séparation et d'évaluation

L'algorithme de séparation et d'évaluation, plus connu sous le nom de "Branch and bound" se base sur l'énumération et l'évaluation progressive de différentes solutions possibles. Le principe de cette méthode consiste à d'une part construire l'arbre de recherche et d'une autre de couper les branches qui ne conduiront pas au résultat optimal [138]. Pour cela, une borne est calculée. Cette borne représente la valeur maximale atteignable si la branche est poursuivie. Une attention particulière doit être portée pour le calcul de la borne, en effet plus cette borne est pertinente, moins les branches sont poursuivies et donc moins il y a de calcul. Au pire des cas, la méthode va énumérer l'ensemble des solutions possibles au problème de départ.

En général, seuls les problèmes de VRP de petite taille peuvent être résolus avec cette méthode (moins de 100 nœuds). Nous pouvons citer comme exemple l'utilisation de la méthode "K-Trees" [54] qui permet de résoudre des problèmes de 25 à 71 nœuds.

Procédure de coupes et de séparation

Cette technique est connue également sous le terme anglais Branch and Cut. L'idée générale des méthodes de coupes est de résoudre un programme en nombres entiers comme une séquence de programmes linéaires.

Un algorithme de coupes et branchements (Branch-and-Cut algorithm) est une technique de séparation et évaluation dans laquelle on applique un algorithme de coupes pour calculer la borne de chaque sous-problème. Cette méthode introduite par Padberg et al. [111] pour le problème du voyageur de

commerce s'est avérée très efficace, elle est maintenant largement utilisée pour résoudre d'une manière exacte des problèmes d'optimisation combinatoire.

En ce qui concerne le VRP, Naddef et al. [117] résolvent des problèmes allant jusqu'à 135 nœuds [138]. La difficulté pour ce problème est la gestion du nombre de contraintes de sous-tours que donnent les relaxations successives.

La programmation dynamique

Cette technique est basée sur le principe dit de Bellman "dans une séquence optimale, chaque sous-séquence doit aussi être optimale". La solution optimale d'un problème de taille n s'exprime en fonction de la solution optimale de problèmes de taille inférieure à n . L'idée est d'éviter de calculer plusieurs fois la solution pour un même sous-problème. Cette technique est limitée puisqu'elle permet de ne résoudre qu'un très petit nombre de problèmes combinatoires. Pour le VRP cela représente des instances de 10 à 20 clients [7].

1.6.3 Méthodes heuristiques pour le VRP

Les méthodes heuristiques nous permettent de trouver des solutions "approchées" pour des problèmes d'optimisation. Ces solutions ne sont pas forcément optimales mais peuvent être obtenues dans un temps relativement correct. En général, ces méthodes permettent de réaliser un bon compromis entre le temps de calcul et la qualité de la solution.

Nous allons, dans ce qui suit, présenter quelques heuristiques classiques pour le VRP.

Heuristiques de résolution

Les heuristiques classiques de résolution du VRP sont divisées en deux grandes familles : les heuristiques de construction et les heuristiques en deux-phases.

1. Heuristiques de construction ou heuristiques gloutonnes :

Ces heuristiques permettent de générer rapidement une première solution à un problème de tournées de véhicules. Deux heuristiques connues sont : les heuristiques avec calcul d'économies (savings) et les heuristiques d'insertion.

- **Les heuristiques avec économies (savings)** : Cette méthode a été introduite par Clarke et al. [31] pour le CVRP et est l'une des plus connues. L'algorithme débute en créant un aller-retour pour chaque nœud i à visiter. A chaque itération une économie s_{ij} est calculée entre deux sommets de la manière suivante : $s_{ij} = d_{i0} + d_{0j} - d_{ij}$ où d_{ij} représente le coût entre le sommet i et le sommet j comme illustré sur la figure (1.2). Cette économie est calculée entre deux trajets possibles $(0, \dots, i, 0)$ et $(0, j, \dots, 0)$ pouvant être regroupés en un seul $(0, \dots, i, j, \dots, 0)$. On choisit de fusionner les deux trajets dont l'économie est la plus grande.

L'application de cet algorithme peut se faire deux manières : soit de manière séquentielle en augmentant une seule route à la fois, soit de manière parallèle en augmentant toutes les routes si

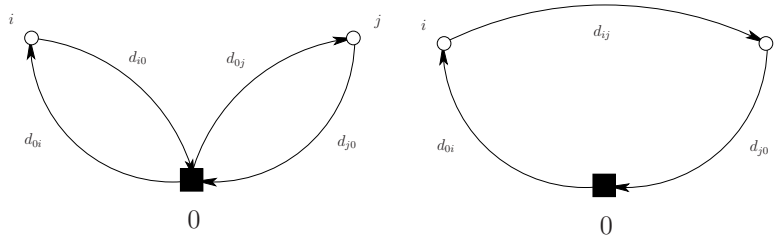


FIG. 1.2 – Méthode avec économies

possible (dans l'ordre décroissant des savings). Toth et al. [138] comparent les deux utilisations de cet algorithme et montrent que l'application de la méthode parallèle donne de meilleurs résultats que la méthode séquentielle en termes de distance totale parcourue.

Cette méthode a fait l'objet de nombreuses adaptations. Certains chercheurs ont proposé des modifications du calcul des économies afin de la généraliser. Gaskell [59] et Yellow [145] ont proposé d'ajouter un paramètre λ au calcul de l'économie tel que, en faisant varier λ , on donne plus ou moins d'importance à la distance entre les deux clients et différentes solutions vont pouvoir être obtenues. D'autres adaptations [4, 41] permettent de calculer les économies de façon à réorganiser la route affectée par le changement, sans préserver l'ordre des deux routes qui sont regroupées. Une autre adaptation a été proposée par Wark et al. [143] et consiste à chaque itération à résoudre un problème de sélection de clients à regrouper. Ce dernier algorithme a montré de très bonnes performances et permet d'atteindre les meilleurs résultats connus pour certaines instances comme répertorié dans Toth et al. [138].

- **les heuristiques d'insertion** : A l'inverse des méthodes utilisant les économies, qui construisent une route pour chaque client, les méthodes d'insertion intègrent au fur et à mesure les clients qui ne font pas encore partie d'un itinéraire.

Les méthodes d'insertion introduites par Mole et al. [101], permettent de sélectionner un nœud puis de l'insérer à la tournée en cours de réalisation de la meilleure façon possible. Pour cela elle utilise deux critères :

- un premier critère $\alpha(i, k, j) = d_{ik} + d_{kj} - \lambda d_{ij}$ permet de savoir où s'insère chaque nœud (non affecté à une route) de manière minimale dans la route en construction,
- un second critère $\beta(i, k, j) = \mu d_{0k} - \alpha(i, k, j)$ permet de choisir le meilleur nœud à insérer dans la route en construction.

La figure (1.3) décrit le processus d'insertion.

Le choix des valeurs pour les paramètres λ et μ changent les politiques d'insertion. Par exemple si $\lambda = 1$ et $\mu = 0$ alors l'algorithme va choisir d'insérer le nœud qui donne la plus petite distance d'ajout. Tandis que si $\lambda = 0$ et $\mu = 0$ alors la méthode va choisir le nœud qui va donner la plus petite distance entre deux nœuds voisins. Cette méthode a été appliquée de manière séquentielle, c'est-à-dire qu'elle construit une route après une autre.

Christofides et al. [30] propose une méthode d'insertion un peu plus sophistiquée qui se déroule également en deux temps et qui utilise également les deux paramètres λ et μ , mais elle peut

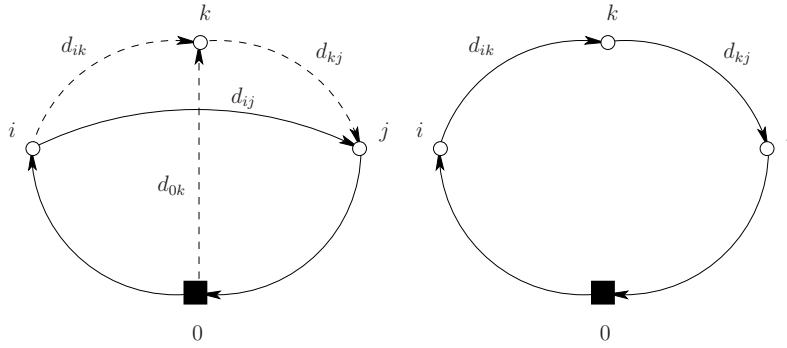


FIG. 1.3 – Méthode d'insertion de Mole et al.

être utilisée en séquentielle ou en parallèle. La méthode parallèle suppose d'avoir une première phase qui évalue le nombre de tournées à construire au total. La méthode de Christofides donne de meilleurs résultats que celle de Mole ainsi que celle de Clarke et al. [92].

Solomon [129] propose différentes méthodes d'insertion pour résoudre des problèmes de VRPTW. Une de ces méthodes utilise également deux critères d'insertion à chaque itération c_1 et c_2 .

Nous allons voir plus en détails cette méthode. Le choix du nouveau nœud à intégrer à une tournée se déroule en deux temps : un premier temps qui consiste à chercher la meilleure insertion dans les différentes tournées pour chaque nœud non encore inséré, puis de choisir celui qui s'insèrera le mieux selon des critères c_1 et c_2 définis en fonction de la distance et du temps de la manière suivante.

$$c_{11} = d_{iu} + d_{uj} - \mu d_{ij}, \mu \geq 0 \quad (1.13)$$

$$c_{12} = b_{ju} - b_j \quad (1.14)$$

$$c_1 = \alpha_1 c_{11} + \alpha_2 c_{12}, \alpha_1 + \alpha_2 = 1, \alpha_1 \text{ et } \alpha_2 \geq 0 \quad (1.15)$$

$$c_2 = \lambda d_{0u} - c_1, \lambda \geq 0 \quad (1.16)$$

L'équation (1.13) représente la distance en plus liée à l'insertion d'un nœud u entre les nœuds i et j . Tandis que le critère (1.14) nous donne l'écart sur le temps de début de service du nœud j si un nœud u est inséré en amont dans le circuit. Ces deux éléments (1.13 et 1.14) forment le critère 1, qui établit pour chaque nœud restant à insérer le meilleur compromis entre la nouvelle distance et le nouveau temps de service. Ensuite le deuxième critère (1.16) nous permet de déterminer, toujours pour chaque nœud, l'éloignement du nouveau nœud à insérer par rapport au nœud dépôt. Une fois calculé le critère 2 pour chaque nœud, il suffit de choisir le nœud qui va s'insérer au mieux dans le circuit, c'est-à-dire celui qui donne le maximum du critère 2. Les paramètres de la méthode (α_1 , α_2 , λ et μ) permettent de définir différentes stratégies

d'insertion de nœud. Par exemple si $\mu = \alpha_1 = \lambda = 1$ et $\alpha_2 = 0$ cela correspond à l'économie qui est faite pour servir le client u sur la même route que i et j à la place de le collecter sur une autre tournée.

Cette méthode peut être utilisée de deux manières possibles :

- de manière séquentielle : on construit les tournées les unes après les autres. La difficulté se pose pour choisir le point de départ de la tournée en cours de construction.
- de manière parallèle : on construit les tournées en même temps. La difficulté se pose pour déterminer le nombre de tournées et les points de départ de chaque tournée.

Mais tandis que Solomon l'utilise de manière séquentielle, Potvin et al. [115] utilise cet algorithme pour définir le nombre de véhicules puis utilise une mesure de regret pour insérer les différents clients.

2. Heuristiques à deux-phases :

Pour ces heuristiques, la construction d'une première solution est divisée en deux étapes consistant à regrouper des nœuds au sein de clusters et à calculer un trajet entre les nœuds d'un même cluster. Selon l'ordre d'application de ces étapes, il existe des heuristiques de type cluster-first, route-second ou bien de type route-first, cluster-second.

- **cluster-first, route-second** : cette heuristique commence par partitionner les nœuds du réseau entre les véhicules et construit par la suite pour chaque véhicule la tournée correspondante. Dans cette catégorie d'heuristiques nous pouvons retrouver, entre autres, deux algorithmes très connus, le "sweep algorithm" développé par Gilett et al. [64] et le "petal algorithm" dont le meilleur exemple est celui développé par Renaud et al. [123]. Le "sweep algorithm" consiste à faire tourner une ligne imaginaire provenant du dépôt, autour de ce dépôt pour former des clusters dans lesquels viennent s'intégrer les nœuds à collecter. Cette ligne balaie les nœuds en faisant un tour complet et chaque nœud balayé par cette ligne sera intégré au cluster en construction pour former par la suite une tournée. Le cluster cesse de grossir lorsque la valeur limite de capacité, de temps ou de distance est atteinte. Un autre cluster est alors entamé. Le "petal algorithm" est une extension de la méthode "sweep" qui consiste à générer des itinéraires appelés "pétales" au sein du cluster en construction et par la suite choisir le meilleur.
- **route-first, cluster-second** : à l'inverse des heuristiques cluster-first, route-second, cette heuristique construit un "tour géant" (en relaxant la contrainte de capacité) passant par tous les nœuds du réseau puis découpe ensuite ce tour en tournées selon le nombre de véhicules nécessaire [12].

Les méthodes de descente :

Les méthodes de descente partent d'une solution initiale et tente d'améliorer cette première solution en explorant son voisinage. Cette exploration recherche une solution de meilleure qualité parmi les voisins de la solution initiale. Cette nouvelle solution devient alors la solution initiale et une nouvelle exploration de voisinage se déroule. La recherche se termine lorsque l'exploration du voisinage n'améliore plus la

qualité de la solution courante. Dans le cadre du VRP, il existe deux types de voisinage : un voisinage modifiant l'ordre des nœuds au sein d'une seule tournée ou un voisinage défini en déplaçant des nœuds entre des tournées différentes.

- **voisinages agissant sur une seule tournée** : ces voisinages sont ceux utilisés pour un problème de voyageur de commerce classique. Ils visent à améliorer la qualité de chaque tournée indépendamment des autres. Ces voisinages se basent, pour la plupart, sur celui développé par Lin [96] la très célèbre méthode $\lambda - opt$, qui consiste à enlever λ nœud(s) dans une tournée et à reconnecter ces λ nœuds de toutes les manières possibles dans la tournée étudiée. En se basant sur cette méthode Or [108] développe le voisinage $Or - opt$ qui consiste à déplacer de 1 à 3 nœuds consécutifs dans une même tournée. Nous pouvons également citer la méthode GENIUS développée par Gendreau et al. [62] qui améliore également une tournée à la fois.
- **voisinages agissant sur plusieurs tournées** : ces voisinages visent à améliorer simultanément la qualité globale de l'ensemble des tournées. Pour cela, Thompson et al. [137] décrivent un cadre général "b-cyclic, k-transfer" dans lequel selon une permutation circulaire de b tournées, k clients seront intégrés à la tournée suivante selon la permutation considérée. Breedam [20], en 1994 donne une classification des différents voisinages multi-tournées en 4 catégories : "String-cross", "String-exchange", "String-relocation" et "String-mix" qui concernent toutes l'échanges de nœuds entre deux tournées [92].

Le voisinage "String-cross" consiste à échanger deux suites de nœuds entre deux tournées en échangeant deux arrêtes de chaque tournée. Le voisinage "String-exchange" échange au plus k sommets entre deux tournées. Le voisinage "String-relocation" transfère au plus k sommets d'une tournée à une autre (sans échange). Et enfin le voisinage "String-mix" consiste à choisir la meilleure amélioration entre celle donnée par la méthode "String-exchange" et celle donnée par la méthode "String-relocation". Ces voisinages sont schématisés dans la figure 1.4.

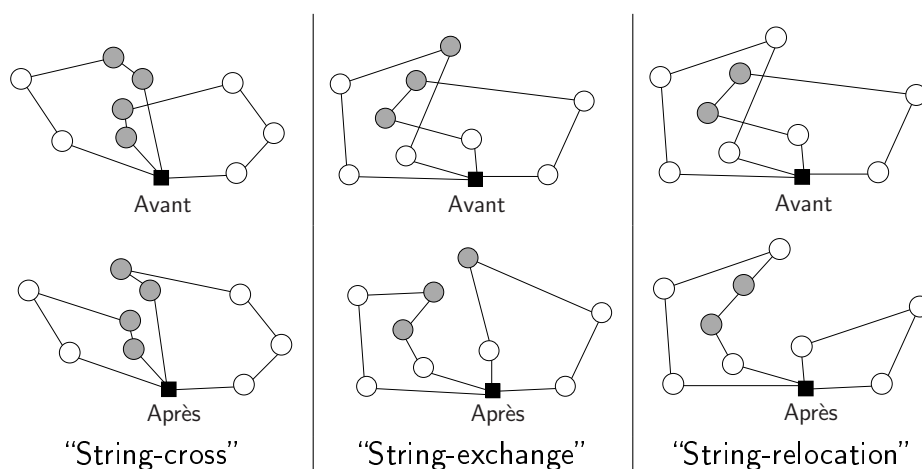


FIG. 1.4 – Voisinages agissant sur plusieurs tournées

Kinderwater et al. [81] définissent des opérations similaires qui améliorent les résultats pour le VRPTW. Dans [146], les auteurs proposent d'utiliser ces voisinages mais en se basant sur différentes

méthodes de sélection des nœuds à changer de place dans les tournées.

Les méthodes de descente se caractérisent par l'arrêt de la recherche de solutions lorsque le voisinage défini ne permet plus de trouver une solution de meilleure qualité. La recherche risque donc de rester "bloquée" dans un optimum local. Pour pouvoir dépasser cet optimum local de nombreuses métaheuristiques ont été développées. Dans la figure (1.5), nous avons illustré les minima locaux et le minimum global d'une fonction $F(x)$ à minimiser.

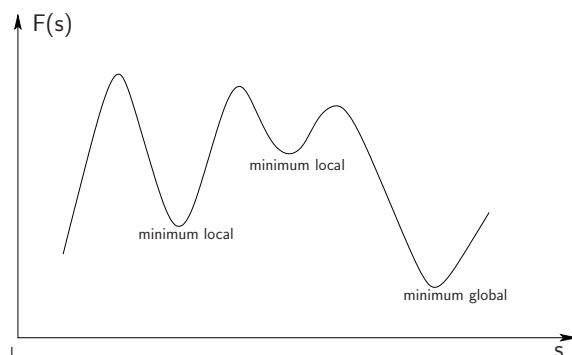


FIG. 1.5 – Mimina locaux et minimum global

Les métaheuristiques

Le terme méta(-) sert à construire des néologismes signifiant ce qui dépasse ou bien ce qui englobe. Certains chercheurs ont tenté de donner une définition à ce terme [110], [142] et [130], mais aucune n'est reconnue universellement.

Les métaheuristiques tirent en particulier leur intérêt dans leur capacité à éviter les optima locaux, soit en acceptant une dégradation de la fonction objectif au cours de leur progression, soit en utilisant une population de solutions.

Par le biais des définitions proposées dans la littérature, nous pouvons en tirer certaines propriétés fondamentales communes à toutes ces méthodes :

- les métaheuristiques sont des stratégies qui permettent de guider la recherche d'une solution optimale ;
- le but visé par les métaheuristiques est d'explorer l'espace de recherche efficacement afin de déterminer des solutions de bonne qualité ;
- les techniques qui constituent des algorithmes de type métaheuristique vont de la simple procédure de recherche locale à des processus d'apprentissage complexes ;
- les métaheuristiques sont en général non-déterministes et ne donnent aucune garantie d'optimalité ;
- les métaheuristiques peuvent contenir des mécanismes qui permettent d'éviter d'être bloqué dans des régions de l'espace de recherche ;
- les concepts de base des métaheuristiques peuvent être décrits de manière abstraite, sans faire appel à un problème spécifique ;

- les métaheuristiques peuvent faire référence à des heuristiques qui tiennent compte de la spécificité du problème traité, mais ces heuristiques sont contrôlées par une stratégie de niveau supérieur ;
- les métaheuristiques peuvent faire usage de l'expérience accumulée durant la recherche de l'optimum, pour mieux guider la suite du processus de recherche.

Souvent inspirées d'analogies avec la réalité (physique, biologie, éthologie, . . .), elles sont généralement conçues au départ pour des problèmes discrets, mais peuvent faire l'objet d'adaptations à des problèmes continus. Les métaheuristiques, du fait de leur capacité à être utilisées sur un grand nombre de problèmes différents, se prêtent facilement à des extensions. Par exemple :

- l'optimisation multi-objectifs (dites aussi multi-critères) [33], où il faut optimiser plusieurs objectifs qui peuvent être contradictoires. La recherche vise alors non pas à trouver un optimum global, mais un ensemble d'optima "au sens de Pareto" formant la "surface de compromis" du problème ;
- l'optimisation dynamique où la fonction objectif varie dans le temps. Il faut alors approcher au mieux l'optimum à chaque pas de temps ;
- l'hybridation, qui vise à tirer parti des avantages respectifs de métaheuristiques différentes en les combinant [132],...

Du fait du foisonnement de la recherche, un grand nombre de méta-heuristiques existent, nous présentons certaines de ces métaheuristiques dans la suite de ce chapitre, celles qui sont le plus couramment utilisées pour le VRP. Nous les avons classés par type de métaheuristiques : celles à solution unique et celles à population de solutions.

Métaheuristiques à solution unique

1. Le recuit simulé

La méthode du recuit simulé a été mise au point par Cerny [26]. Elle est inspirée d'un processus utilisé en métallurgie. En faisant fondre un alliage, lors de sa solidification des cristaux se forment à partir des impuretés présentes dans le métal en fusion : ce sont les optima locaux dans un problème d'optimisation combinatoire. La forme et la taille de ces cristaux conditionneront les caractéristiques mécaniques de la pièce solide. Pour modifier les caractéristiques mécaniques de la pièce et diminuer ces "défauts", on effectue souvent une opération que l'on nomme recuit. C'est à dire que l'on chauffe à nouveau la pièce métallique proche du point de fusion, puis on refroidit à nouveau plus lentement. Les anciens cristaux métalliques, sont en partie remplacés par d'autres, des cristaux formés par les impuretés disparaissent, ce qui constitue donc un nouvel arrangement de la matière, et on obtient une structure mieux ordonnée, assimilable à une tendance de se rapprocher de l'optimum global.

Par analogie avec le processus physique, la fonction à minimiser deviendra l'énergie E du système. On introduit également un paramètre fictif, la température T du système. Cette procédure a pour caractéristique principale d'intégrer des éléments probabilistes pour être guidé, ceci implique une difficulté de paramétrage de ces éléments.

L'algorithme de recuit simulé part d'une première solution (soit calculée, soit obtenue de manière

aléatoire), et choisit aléatoirement dans son voisinage une autre solution. Soit celle-ci améliore le critère que l'on cherche à optimiser, on dit alors qu'on a fait baisser l'énergie du système, soit celle-ci le dégrade. Si on accepte une solution améliorant le critère, on tend ainsi à chercher l'optimum dans le voisinage de la solution de départ. L'acceptation d'une solution de moins bonne qualité permet de ne pas s'enfermer dans un optimum local. La difficulté pour ce type d'algorithme est de déterminer les valeurs pour les paramètres en entrée et notamment la température T . Cette température décroît au cours de l'algorithme (et tendre vers 0) et correspond à la probabilité d'accepter une moins bonne solution (plus T sera élevée plus la probabilité d'accepter une moins bonne solution sera forte). Déterminer T et sa loi de décroissance peut se révéler compliqué puisque si T refroidit trop rapidement l'algorithme va se retrouver "bloqué" dans un optimum local et si elle est trop lente, l'algorithme va être trop gourmand en temps de calcul. Contrairement à d'autres métaheuristiques, il a été montré que le recuit simulé converge asymptotiquement vers l'optimum global [67].

En ce qui concerne les problèmes de VRP et ses dérivés, les premiers recuits simulés ont été développés par Alfa et al. [3] mais ces développements n'ont pas donné de très bons résultats [138]. D'autres chercheurs, par la suite, ont utilisé cette technique avec succès pour l'appliquer au VRP comme par exemple [109], [21] ou [29]. Plus récemment les mécanismes utilisés pour les recuits simulés sont utilisés de manière combinée avec une autre métaheuristique, comme par exemple dans [28] dans lequel les auteurs utilisent deux métaheuristiques différentes pour garder les avantages de chacune.

2. La recherche tabou

La recherche tabou a été initiée au début des années 80. Elle a été introduite par Glover [65]. Cette méthode est une métaheuristique itérative fondée sur l'exploration d'un voisinage. C'est l'une des rares métaheuristiques qui soit déterministe.

Partant d'une solution initiale, le processus consiste, à chaque itération, à choisir la meilleure solution dans le voisinage de la solution courante, même si elle n'entraîne pas une amélioration de la fonction objectif (afin de ne pas "tomber" dans un optimum local). Pour éviter alors de boucler sur un même ensemble de solutions, la méthode tabou s'appuie sur une mémoire des solutions explorées. Cette mémoire est appelée liste de solutions "interdites" ou tabou (d'où le nom de la méthode). Cette liste tabou peut contenir par exemple certains attributs des dernières solutions visitées et interdit pendant un certain nombre d'itérations le retour vers l'ensemble des solutions contenues dans cette liste. Lorsque une nouvelle solution est choisie, la liste tabou est mise à jour en remplaçant les attributs de la plus ancienne par ceux de la nouvelle. La méthode tabou stoppe dès qu'un critère d'arrêt est vérifié. Ce critère peut être le temps d'exécution de la méthode, le nombre d'itérations, un seuil minimal à atteindre pour l'objectif de la solution,... La meilleure solution faisable de toutes celles qui ont été explorées est bien sûr celle qui sera conservée (ce n'est pas forcément la dernière solution explorée).

En ce qui concerne le VRP, et ses extensions (comme par exemple le VRPTW), il existe de

nombreuses applications utilisant une recherche tabou. Dans [23], les auteurs exposent certaines méthodes déjà développées pour le problème de VRPTW. Nous pouvons, par exemple, citer [131] ou bien [124] qui utilise la recherche tabou couplée à une “mémoire adaptative” qui a pour but de modifier les différentes solutions de départ à l'aide de combinaisons de tournées extraites de bonne solution déjà trouvées et peuvent mêmes dans certains cas nous donner les meilleures solutions. La recherche tabou développée par Gendreau et al. [60] permet de trouver des résultats de très bonne qualité voir même égales aux meilleures solutions connues. Cordeau et al. [36] proposent une méthode qui permet de visiter des solutions ne respectant pas les contraintes du problème tout en associant des pénalités à ces violations de contraintes dans la fonction de coût.

Plus récemment, nous pouvons citer Toth [139] et Tarantilis [135] qui proposent des méthodes tabou qui sont très efficaces et qui produisent des résultats de très bonne qualité. Toth et al. [139] prétraitent le problème pour améliorer les résultats. Tandis que Tarantilis et al. [135] utilisent la notion de mémoire adaptative.

En ce qui concerne le DVRP, plusieurs études ont été menées, notamment par Rego et al. [122] et Ichoua et al. [74].

3. Variable Neighborhood Search (VNS)

Cette métaheuristique introduite récemment par Mladenović et al. [100] est basée sur l'exploration successive de plusieurs voisinages. Pour mettre en œuvre cette méthode, il faut disposer d'un ensemble ordonné de voisinages qui sont explorés successivement.

Cette méthode part d'une solution initiale s et elle se déroule ainsi :

- (a) une solution s' est choisie aléatoirement dans un voisinage donné de s ;
- (b) le voisinage de s' est ensuite exploré pour en extraire la meilleure solution s'' ;
- (c) Si s'' n'améliore pas la meilleure solution connue, on passe au voisinage suivant ;
- (d) Si en revanche s'' améliore la meilleure solution connue, on la mémorise puis on remplace s par s'' et on retourne à l'étape (a).

A l'inverse des autres métaheuristiques, cette méthode utilise peu ou pas de paramètres, mais nécessite un ensemble de voisinages avec un ordre entre eux. Elle a déjà prouvé son efficacité pour certain problème de VRP, comme le VRPTW [22], et même le CARP [114].

4. Large Neighborhood Search (LNS)

Cette métaheuristique a été présentée par Shaw [127]. Elle se base sur les mêmes principes que ceux de VNS. La différence s'établit lors de l'utilisation des voisinages. Ceux-ci ne sont pas appliqués sur l'ensemble du problème mais sur une partie des variables du problème et peuvent donc être combinés entre eux sur un même problème. Ceci permet de visiter le voisinage d'une solution de manière plus large.

Une étude récente proposée par Pisinger et al. [113] utilise une extension du LNS : Adaptive Large Neighborhood Search (ALNS) qui utilise une méthode de recherche spécifique pour trouver

le voisinage, et consiste à détruire une partie de la solution pour ensuite la reconstruire à l'aide de méthodes heuristiques spécifiques. Cette méthode a été testée sur plusieurs types de problèmes types de VRP, et notamment ceux avec fenêtres de temps et donne pour certains problèmes de meilleurs résultats que ceux déjà connus.

Métaheuristiques à population de solutions

1. Les algorithmes évolutionnaires

Les algorithmes évolutionnaires sont des algorithmes d'optimisation qui s'appuient sur des techniques inspirées de la génétique et de l'évolution naturelle [38] : sélection, croisement, mutation, ... C'est au début des années 60 que John Holland a commencé à s'intéresser à ce qui allait devenir les algorithmes évolutionnaires. Ses travaux trouvent un premier aboutissement en 1975 [73]. L'ouvrage de Goldberg [66] a également largement contribué à les vulgariser. Les quatre éléments fondamentaux des algorithmes évolutionnaires sont :

- l'évaluation du niveau d'adaptation d'un individu (évaluation de la fonction objectif à optimiser),
- la sélection : représentant le choix des individus en fonction de leur niveau d'adaptation,
- le croisement : correspond au mélange des bagages génétiques entre individus,
- la mutation : traduisant une modification abrupte du bagage génétique d'un individu.

Les algorithmes évolutionnaires servent à simuler le processus d'évolution d'une population. A partir, d'une population de n individus (n solutions d'un problème donné), des opérateurs de sélection, croisement et mutation sont appliqués à l'ensemble de ces individus pour en définir des nouveaux. La sélection a pour but de favoriser les meilleurs éléments de la population, le croisement et la mutation assurent une large exploration de l'espace de recherche de part en diversifiant la population d'individus. De nouveaux individus vont être évalués et vont venir remplacer certains plus anciens ou moins bons, ... Ainsi la population des n individus évolue au cours du temps et contient des individus de mieux en mieux adaptés au problème. Elle se dirige donc vers l'optimum. Les critères d'arrêt de la méthode sont un nombre fixé de générations, une limite de convergence de la population, ou une population qui n'évolue plus suffisamment, etc. Les algorithmes évolutionnaires diffèrent des algorithmes classiques d'optimisation et de recherche essentiellement en quatre points fondamentaux :

- Ils utilisent un codage des éléments de l'espace de recherche et non pas les éléments eux-mêmes,
- Ils recherchent une solution à partir d'une population de solutions et non pas à partir d'une seule solution,
- Ils n'imposent aucune régularité sur la fonction étudiée (continuité, dérivabilité, convexité. . .),
- Ils ne sont pas déterministes, et utilisent des règles de transition probabilistes (comme le recuit simulé).

Les algorithmes évolutionnaires ont fait l'objet de nombreuses études et notamment pour le VRP et ses extensions. Des algorithmes évolutionnaires pour le VRP classique ont été proposés récemment contrairement au VRPTW [61, 24, 133]. C'est en 2003 que Berger et al. [16] proposent

un algorithme génétique compétitif pour le problème de CVRP basé sur l'évolution parallèle de deux populations de solutions afin de minimiser la distance totale de parcours. Ils s'inspirent des techniques connues en tournées de véhicules pour faire évoluer les populations. En intensifiant la recherche dans des parties de l'espace de solution qui semblent intéressantes de part l'utilisation d'algorithmes évolutionnaires, ils diversifient la recherche, et en utilisant les techniques connues, ils intensifient la recherche dans certaines parties. Prins [116] propose également un algorithme évolutionnaire qui permet de résoudre efficacement le problème de VRP et des algorithmes pour résoudre le CARP.

L'aspect dynamique du VRP a également fait l'objet, récemment, d'études [71] et [70]. Dans ces articles, ce sont des problèmes de temps réel qui sont étudiés.

2. Les colonies de fourmis

C'est une métaheuristique des plus récentes. Les colonies de fourmis proviennent des travaux de Coloni [34], elles ont été créées à partir de la constatation suivante : les fourmis résolvent naturellement des problèmes souvent complexes. Les biologistes ont étudié comment les fourmis arrivent collectivement à résoudre des problèmes trop complexes pour un seul individu, notamment les problèmes de choix lors de la recherche de sources de nourriture.

Leur technique se base sur la pose de marqueurs chimiques, les phéromones, déposées sur les trajets parcourus et qui pourront être suivis par leurs congénères. Les fourmis utilisent les pistes de phéromones pour marquer leur trajet, par exemple entre le nid et une source de nourriture. Une colonie est ainsi capable de choisir (sous certaines conditions) le plus court chemin vers une source à exploiter, et ceci sans que les individus aient une vision globale du trajet. En effet, les fourmis qui reviennent le plus rapidement au nid (celles qui auront emprunté le chemin le plus court) auront déposé sur leur trajet deux fois plus de phéromones. Les fourmis suivantes ayant tendance à emprunter les chemins les plus marqués, une solution va progressivement se dessiner. Les premiers algorithmes de colonies de fourmis développés [45, 44, 46] concernaient le problème du Traveling Salesman Problem. Cette méthode a été, par la suite, testée et adaptée à nombreux types de problèmes. Mais les auteurs ayant remarqué que les colonies de fourmis n'explorent pas assez l'espace de recherche, ils les ont souvent couplées avec des méthodes de descente.

Malgré que cette approche soit assez récente, elle a déjà été avec succès pour certaines applications, et notamment pour des problèmes de VRP ou certaines de ces extensions [25][58]. Avec une heuristique 2-opt ajoutée pour améliorer les tournées individuellement obtenues par les colonies, cette approche donne des résultats légèrement inférieurs à ceux donnés par la recherche tabou. Plus récemment, on peut noter l'algorithme développé par Tan et al. [134]. L'aspect DVRP a été étudié par Montemanni et al. [102].

Le schéma suivant (1.6) résume les méthodes de résolution vues dans cette partie.

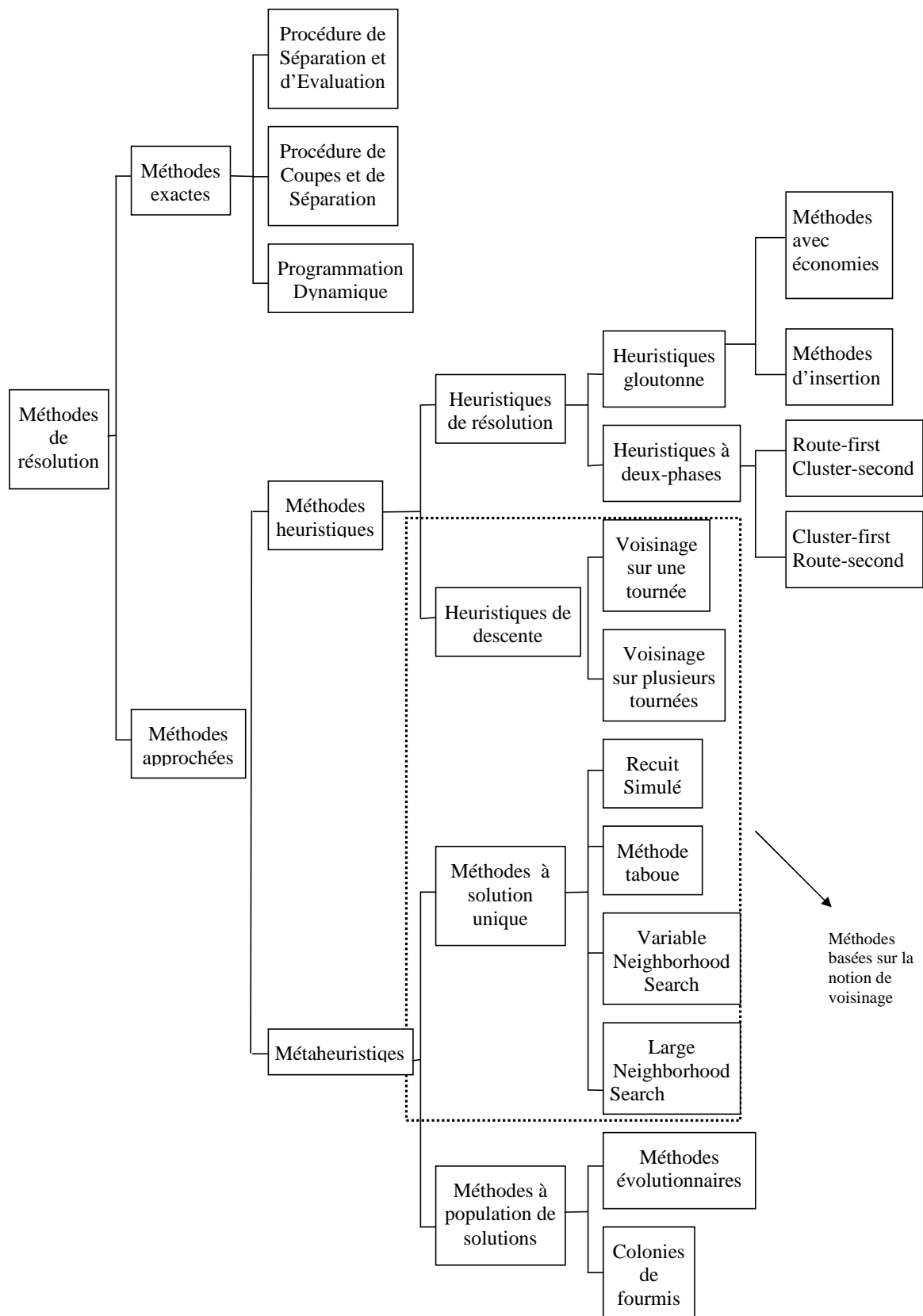


FIG. 1.6 – Récapitulatif des méthodes de résolution

1.7 Problèmes de décision multi-objectifs

Les problèmes d'optimisation combinatoire provenant de cas réels sont, pour la plupart, par nature multi-objectifs. Plusieurs critères, qui peuvent être contradictoires, sont à considérer simultanément pour résoudre le problème. L'optimisation multi-objectifs tire ses racines en économie par les travaux de Edgeworth [49] et de Pareto [112]. Malgré l'intérêt de l'étude de ce type de problèmes, il va falloir attendre les années 80-90 pour voir émerger des méthodes de résolution de problèmes multi-objectifs.

1.7.1 Définition

Un problème multi-objectif (PMO) peut être décrit de la manière suivante :

$$F(x) = (f_1(x), f_2(x), \dots, f_n(x)) \quad (1.17)$$

$$x \in D \quad (1.18)$$

où n représente le nombre d'objectifs à atteindre ($n \geq 2$) et $x = x_1, x_2, \dots, x_k$ est le vecteur représentant l'ensemble des variables de décision appartenant à l'ensemble D des solutions réalisables. L'équation (1.17) représente l'ensemble des fonctions objectifs à optimiser.

L'ensemble des solutions d'un problème multi-objectifs dites non-dominées peuvent être réparties sur un "Front Pareto". Les solutions non-dominées sont des solutions pour lesquelles une amélioration d'un objectif provoquerait la dégradation d'un autre objectif. Le schéma (1.7) illustre ce Front Pareto pour la minimisation de deux objectifs en décrivant des solutions non-dominées et les solutions dominées.

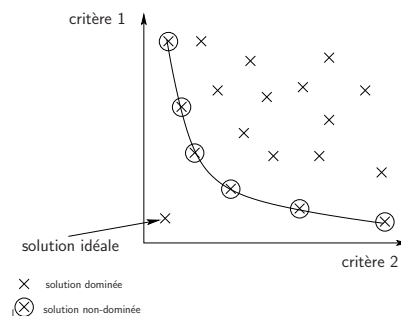


FIG. 1.7 – Front Pareto

Lors de la résolution d'un problème multi-objectifs, une difficulté est de s'approcher au mieux du Front Pareto. Une autre difficulté est d'obtenir les solutions bien réparties sur le Front Pareto sur l'ensemble du Front Pareto de manière à proposer des compromis différents sur les objectifs.

Les premières études sur les problèmes multi-objectifs qui ont été menées transformaient ces problèmes en un problème mono-objectif. Pour cela une approche consiste à donner un poids différent à chaque objectif selon son degré d'importance dans une seule fonction objectif. Une autre approche

consiste à résoudre une succession de problèmes mono-objectif en fonction de l'importance donnée à chacun des objectifs : un objectif est optimisé si cela ne dégrade pas les valeurs pour les objectifs prioritaires.

Cependant en pratique, il est souvent très difficile de hiérarchiser les critères ou de définir des pondérations pour chacun d'eux. Dans cette situation nous allons plutôt parler de "compromis" entre plusieurs objectifs pour un utilisateur. L'ensemble de ces compromis correspond à l'ensemble des solutions dites "Pareto Optimales". Chaque solution de cet ensemble est optimale, dans le sens où on ne peut pas améliorer la solution selon un objectif sans en dégrader au moins un autre. Pour résoudre ce type de problème, il est indispensable de faire intervenir l'humain à travers le décideur pour le choix de la solution à conserver au final [42]. Par exemple, sur la figure (1.7) chaque point extrême du Front Pareto représente la minimisation stricte selon un objectif, c'est le résultat que l'on obtiendrait en optimisant un seul des deux objectifs. Les autres solutions appartenant au Front Pareto représentent des compromis entre les deux objectifs.

Ceci nous amène à classer les méthodes selon le rôle joué par le décideur [33]. Nous avons les familles suivantes :

- les méthodes à priori : ces méthodes consistent à modéliser un compromis sur les objectifs avant l'exécution de la méthode. Cela permet d'obtenir une méthode mono-objectif qui nous donne une unique solution et évite de générer différentes solutions du Front Pareto. L'avantage de ces méthodes est de faire une seule recherche de solution, l'inconvénient est de devoir modéliser le compromis avant d'obtenir un résultat et donc il est possible qu'après cette exécution, le décideur ne soit pas satisfait et doive réitérer la méthode avec un nouveau compromis.
- les méthodes progressives : ces méthodes consistent à réajuster au cours de l'exécution de la méthode le compromis posé avant cette exécution. Elle génère une solution intermédiaire pouvant être qualifiée par rapport aux objectifs. L'avantage de ces méthodes est de pouvoir réorienter les solutions vers le compromis réel souhaité. L'inconvénient est de nécessiter l'intervention du décideur tout au long de la recherche de solution.
- les méthodes à posteriori : ces méthodes consistent à chercher l'ensemble des solutions de manière à ce qu'elles soient réparties au mieux sur le Front Pareto. Les avantages de cette méthode sont de ne pas avoir à fixer le compromis entre les objectifs avant l'exécution et de proposer un nombre suffisant de solutions au décideur pour qu'il puisse choisir le compromis qu'il souhaite. L'inconvénient majeur est le temps d'exécution.

Plusieurs états de l'art et livres sont consacrés à ce problème et proposent des études plus complètes ([32, 33, 50, 51]). Par la suite, nous allons résumer quelques unes des méthodes multi-objectifs classiques rencontrées dans ces différentes références.

Méthodes scalaires

Ce sont des méthodes qui "transforment" le problème de base multi-objectifs en un problème mono-objectif. Ces méthodes consistent à connaître à priori les préférences du décideur et fixe ainsi avant

l'exécution de la méthode le compromis que l'on désire entre les différents objectifs.

Dans ces méthodes, nous pouvons citer la **méthode par agrégation**. Cette méthode est la plus évidente. En effet, elle consiste à sommer l'ensemble des objectifs dans une fonction objectif globale. Chaque objectif peut avoir un certain poids, c'est-à-dire que selon le coefficient que l'on choisit pour un objectif, celui-ci peut se trouver favorisé ou non par rapport aux autres. L'avantage de cette méthode est de pouvoir utiliser par la suite toutes les méthodes de résolution de problèmes combinatoires mono-objectif. L'inconvénient majeur est de devoir déterminer les importances relatives des différents objectifs, ce qui nécessite de bien connaître le problème. Si ce n'est pas le cas, le décideur peut ne pas être satisfait du résultat et relancer la recherche de solution avec d'autres poids et cela tant qu'il ne sera pas satisfait. L'avantage du temps d'exécution relativement rapide est donc perdu.

La **méthode \in *-contraintes*** fait également partie de cette catégorie de méthode. Elle consiste à optimiser un objectif en considérant tous les autres comme contraintes à respecter. Pour trouver plusieurs solutions à un même problème, on fait varier l'objectif à résoudre. L'inconvénient de cette méthode est que l'on optimise un seul objectif, les autres doivent juste être respectés. La méthode doit donc être utilisée plusieurs fois considérant successivement différents objectifs à optimiser pour trouver plusieurs solutions de compromis.

Une autre méthode connue est la **méthode lexicographique**. Cette méthode consiste à optimiser le problème selon un premier objectif en posant les autres en contraintes. Ceci nous permet d'obtenir une première solution. Ensuite ce premier objectif est transformé en contrainte d'égalité et l'objectif suivant est optimisé. Cette méthode est ensuite itérée sur l'ensemble des objectifs. La solution obtenue dépend de la hiérarchisation établie sur les différents objectifs. Il est donc important de bien connaître le problème à priori.

Méthodes métaheuristiques

Les principales métaheuristiques sont le recuit simulé, la recherche tabou et les algorithmes évolutionnaires. Lorsqu'elles sont utilisées en multi-objectifs, elles peuvent être soit couplées avec les méthodes multi-objectifs (les algorithmes évolutionnaires) soit découplées (le recuit simulé ou la recherche tabou), c'est-à-dire que la méthode multi-objectifs utilisée est soit intégrée dans les propriétés de la métaheuristique, soit elle est utilisée en tant que guide au sein de la métaheuristique.

Les difficultés pour ces méthodes se posent pour :

- l'évaluation des solutions obtenues selon les objectifs : soit par l'utilisation d'une méthode multi-objectifs (méthodes découplées), soit en utilisant directement les propriétés de la métaheuristique utilisée (méthodes couplées).
- les systèmes d'exploration des solutions : diversifier l'exploration des solutions au maximum.

Par la suite chaque solution non-dominée au sens Pareto doit être conservée pour être au final proposée au décideur.

Une méthode connue est la **méthode MOSA** [33]. Cette méthode est basée sur le recuit simulé. La probabilité d'acceptation de mauvaises solutions est calculée pour chaque objectif. Une fois le calcul de

ces probabilités fait, une agrégation est faite de ces probabilités. Un ordre sur les objectifs est donc posé. La différence avec une méthode d'agrégation classique est de proposer une probabilité d'acceptation pour chaque objectif, elle prend donc plus en compte l'aspect multi-objectifs que les méthodes de pondération classiques.

Une autre méthode très connue est la **méthode VEGA** [33]. Cette méthode se base sur les algorithmes évolutionnaires. Elle considère une population de n individus. Ces n individus sont répartis en k groupes (ici k représente le nombre d'objectifs) de n/k individus. A chaque groupe, on associe une fonction objectif qui évalue chaque individu du groupe. Puis, lors de la sélection les individus sont mélangés et les croisements se font selon l'efficacité de chaque individu. Les nouveaux individus vont faire partie d'un groupe selon l'évaluation de son efficacité suivant l'objectif affecté. Cette méthode propose donc un ensemble de solutions.

1.7.2 VRP et multi-objectif

Les problèmes de tournées de véhicules académiques nécessitent généralement d'être adaptés lorsque l'on se trouve en présence d'un cas réel. Ces adaptations sont souvent des contraintes que l'on rajoute au modèle de départ [138], comme notamment les extensions vues précédemment.

De plus, généralement dans les cas réels, il ne suffit pas d'optimiser un seul objectif, mais d'en prendre en compte plusieurs qui peuvent être contradictoires. Les objectifs les plus communs sont de minimiser la distance totale, le temps total de parcours, le coût global, la taille de la flotte, l'équilibrage des tournées, de maximiser la qualité de service ou le profit collecté, etc. De manière générale, Jozefowicz [76] propose dans sa thèse de classer les objectifs selon leur appartenance à une composante du problème : la tournée (coût, travail,...), les nœuds ou les arcs (fenêtres de temps, service du client,...) et les ressources (taille de la flotte, marchandises,...). Chaque objectif sera ainsi défini par son implication dans l'une de ses catégories.

Toujours selon Jozefowicz [77], l'étude des problèmes de tournées de véhicules multi-objectifs est principalement due à trois motivations : l'extension des problèmes classiques académiques, la résolution de problèmes réels ou bien encore l'étude d'objectifs complémentaires sans abandonner pour autant l'objectif principal classique qui est celui de la minimisation de la distance totale.

Il existe de nombreuses références sur cette problématique en littérature, notamment en étude de problème bi-objectif, dans cette partie nous allons répertorier certaines études de manière non exhaustive.

La plupart des études sur le problème de VRP multi-objectifs concernent des études bi-objectifs [106, 76, 13]. Dans ces études, une façon, relativement commune, de prendre en compte plusieurs objectifs consiste à les agréger dans une fonction objectif finale [118, 97]. Une autre approche est apparue avec les colonies de fourmis. En effet, des recherches ont été menées de façon à utiliser une colonie par objectif comme dans [9, 119]. Il existe de nombreuses autres méthodes de résolution, il semblerait toutefois que les méthodes évolutionnaires aient été le plus utilisées [106, 76, 89] et ont prouvé leur efficacité, comme par exemple la méthode NSGA-II ([78, 144]). En effet, elles semblent

bien se prêter à l'exploration et sont souvent hybridées afin d'intensifier les recherches [17]. Par ailleurs d'autres méthodes peuvent être utilisées comme des heuristiques de construction et d'amélioration [118, 97]. On peut également utiliser d'autres métaheuristiques comme la méthode tabou [13] ou le recuit simulé [53].

En résumé, les méthodes multi-objectifs pour le VRP ont évolué depuis quelques années. Tandis qu'elles se limitaient à des méthodes scalaires, dorénavant elles tendent à exploiter les métaheuristiques. Nous avons également pu observer que de nombreuses études utilisent les algorithmes évolutionnaires, souvent hybridés avec des méthodes heuristiques afin d'explorer de façon plus efficace l'espace de recherche.

1.8 Conclusion

Dans ce chapitre nous avons présenté le problème de base au cœur de ce mémoire, celui des tournées de véhicules (VRP). Nous avons vu la formulation de ce problème et certaines extensions proposées dans la littérature.

Ce problème fait partie des problèmes d'optimisation NP-difficile et utilise donc des méthodes de résolution appropriées. Ces méthodes sont divisées en deux grandes catégories, les méthodes exactes qui permettent d'obtenir la solution optimale et les méthodes approchées qui ne garantissent en rien de trouver la solution optimale mais permettent de résoudre en un temps raisonnable tous les types de problèmes. Les méthodes approchées sont réparties entre les méthodes de descente qui ont l'avantage d'être faciles à mettre en œuvre mais qui ont l'inconvénient de s'arrêter sur des optima locaux ; et les métaheuristiques qui sont plus difficiles à mettre en œuvre mais qui permettent de dépasser en partie cette barrière des optima locaux. Les méthodes exactes ne sont pas adaptées aux problèmes de grande taille en raison de temps de calcul qu'elles nécessitent. Les méthodes approchées par contre sont adaptées à tous les types de problèmes.

Parmi les méthodes approchées, les méthodes heuristiques utilisent des mécanismes relativement simples. Les métaheuristiques donnent en général des solutions de meilleure qualité mais en contre partie sont plus difficiles à implémenter notamment en raison de leur paramétrage qui peut être compliqué et prendre du temps.

Dans une dernière partie, nous avons évoqué les problèmes multi-objectifs plus particulièrement dans le contexte des tournées de véhicules. Nous avons pu constater que la plupart des études pratiques utilisent soit l'agrégation des critères pour avoir au final une unique solution, soit un Front Pareto qui permet de mettre en avant les solutions non-dominées. Cet ensemble est intéressant puisqu'il permet de garder différentes propositions pour le décideur. Ce Front est particulièrement pratique et est bien adapté notamment pour les études bi-objectifs, sur lesquels les solutions non-dominées restent pertinentes et visuelles. Pour obtenir ce Front Pareto, différentes méthodes peuvent être utilisées, nous pouvons constater qu'en pratique par les algorithmes évolutionnaires se sont révélés très efficaces.

Chapitre 2

Collecte des déchets

Ce chapitre s'intéresse plus précisément au problème de tournées qui nous préoccupe dans ce mémoire : celui de la collecte des déchets ménagers et assimilés. Nous allons commencer par un état de l'art des études menées dans ce domaine. La problématique de l'organisation de la collecte des déchets répond à un besoin réel pour les collectivités en charge de cette collecte ; c'est pourquoi ce domaine a suscité ces vingt dernières années de plus en plus d'études scientifiques spécifiques pour gérer des problèmes concrets de collecte des déchets.

Par la suite, nous détaillons le fonctionnement des collectes mises en place par une collectivité locale particulière. Enfin, nous présentons deux problèmes de collecte émergents liés à des caractéristiques de cette collectivité locale. Ces deux problèmes sont au cœur des développements effectués dans ce mémoire et sont étudiés dans les chapitres suivants.

2.1 Types de collecte

Les déchets les plus classiques sont les ordures ménagères (OM). Ces ordures suivent des filières classiques et sont soit incinérées et par la suite enfouies, soit enfouies directement. Depuis quelques années sont apparus massivement les déchets recyclables. Ils concernent les déchets pouvant être retraités et ainsi fournir des nouvelles matières de base aux industries. Chaque type de déchets recyclables suit une filière de traitement spécifique, comme par exemple, le papier, le verre,... Ces déchets sont collectés et traités différemment des OM. Les déchets verts, eux, peuvent être directement traités par les usagers par l'usage de composteur ou faire, selon le volume, l'objet de collecte spécifique et de traitement. Enfin, les collectivités mettent souvent en place des filières de traitement des encombrants pour traiter les déchets de gros volumes. Certains déchets sont également collectés directement en déchetteries, c'est-à-dire que les usagers apportent eux-mêmes leurs déchets dans un lieu de collecte. Ces déchetteries peuvent être vues comme des relais entre les usagers et les filières de traitement. Elles engrangent une forte quantité de déchets et se chargent de les transmettre dans les filières de traitement concernées en aval.

Les collectes mises en place pour des déchets de différentes natures varient. Le premier type de collecte est le porte à porte : il consiste à collecter les déchets des usagers placés dans des bacs individuels

ou collectifs selon les habitations. Ce type de collecte oblige à passer régulièrement devant toutes les habitations, il est utilisé généralement lorsque les quantités de déchets sont importantes, comme par exemple pour les ordures ménagères classiques. Pour les déchets spécifiques et les produits en moindre quantité, il est possible de mettre en place des bornes d'apport volontaire (comme par exemple pour le verre). Les collectes sur ces points d'apport volontaire sont généralement moins régulières que les collectes en porte à porte.

2.2 État de l'art sur la collecte des déchets

La problématique de l'optimisation des collectes des déchets est apparue en recherche depuis quelques années, notamment avec l'émergence de la prise en compte de contraintes environnementales. La question de la gestion des déchets se pose dans tous les types de communes : il faut déterminer, en fonction des types de déchets, les filières de traitement à mettre en place et les modes de collecte à effectuer. Les différents travaux existant visent à permettre aux organismes en charge de cette gestion d'en limiter les coûts tout en garantissant une bonne qualité de service et tout en limitant les impacts environnementaux.

La problématique de l'optimisation de la gestion des déchets est assez récente (environ vingt ans) mais a déjà suscité de nombreuses études [105]. La plupart des articles de la littérature portent sur un problème réel et vise une application concrète à mettre en place au bout de l'étude. Ils ont un grand intérêt pratique pour la mise en œuvre, mais comportent souvent des spécificités qui sont propres à chaque problème étudié.

Les études se basent sur deux types de modélisation : celle basée sur le VRP ou celle basée sur le CARP. S'il semble plus naturel de modéliser la collecte des déchets ménagers en utilisant un graphe de type CARP (puisque la collecte en porte à porte se fait souvent sur une rue). Cette modélisation a été moins développée au profit d'une modélisation plus agrégée de type VRP, qui dans les faits se révèle plus pratique et plus facile à mettre en œuvre en raison notamment des volumes de données à traiter.

Les objectifs classiques de la collecte des déchets sont de minimiser le nombre de véhicules nécessaires à cette collecte, de minimiser le temps total de collecte ou la distance totale parcourue par les véhicules [106]. Sans prétendre à l'exhaustivité, nous présentons ci-après différentes études d'optimisation de collecte dont nous avons connaissance.

En 1990, Ong et al. [107] proposent une heuristique "route first-cluster second" pour optimiser la collecte à Singapour. Par la suite, en 1993, Janssens [75] détermine un modèle mathématique permettant de calculer la taille optimale de la flotte de véhicules dans le cas particulier de la collecte des huiles sales en Belgique. La même année, Alvarez-Valdes et al. [98] proposent d'utiliser une nouvelle stratégie pour les rippers, qui consiste à déplacer les containers à des points de jonction pour minimiser ainsi les trajets effectués par les véhicules. Kulcar [85], en 1996, propose un modèle pour élaborer des stratégies de collecte et notamment l'utilisation de plusieurs dépôts pour les véhicules ou encore l'utilisation de plusieurs modes de transport (transport multi-modal dans un contexte urbain).

En 1997, Eisenstein et al. [52] développent une méthode qui permet d'organiser les dépôts de véhicules de collecte à Chicago, ils utilisent un algorithme de résolution dynamique sur l'horizon d'une semaine qui permet de visiter une ou deux fois par jour le lieu de vidage, selon la capacité du véhicule utilisé lors d'un itinéraire tout en maximisant le niveau de service proposé. Pour cela, cet algorithme dynamique utilise les processus de décision de Markov et propose des tournées variables calculées de manière dynamique (selon la quantité de déchets déjà collectée par un véhicule). Chang et al. [27], cette même année, s'intéressent à comment combiner les fonctions d'un SIG (Système d'Information Géographique) et un modèle analytique multi-objectifs afin d'aider à analyser les alternatives proposées pour les stratégies possibles de collecte des déchets à Taiwan. Toujours en 1997, Smith [82] propose une étude de cas en Afrique du Sud permettant de gérer les collectes des déchets en démontrant également la difficulté de transférer des modèles et des méthodes de résolution pour d'autres cas. Les recommandations proposées s'inscrivent essentiellement pour les problèmes qui ont des particularités similaires à celles des collectes appliquées en Afrique du Sud.

Bommisetty et al. [19] en 1998 s'intéressent à la collecte des déchets recyclables sur un campus et utilise une modélisation de type VRP Périodique. Pour résoudre leur problème ils utilisent une heuristique en 2 phases.

En 2000, Tung et al. [140] basent leur étude, qu'ils modélisent comme un VRPTW, sur le cas particulier de la collecte des déchets à Hanoi. Leur méthode se déroule en deux temps. Tout d'abord, ils utilisent un algorithme de construction en utilisant les principes de l'algorithme d'insertion de Solomon et dans un deuxième temps, ils améliorent cette première solution à l'aide d'algorithmes de descente tels que Or-opt et 2-opt. Ce papier présente des résultats qui comparent la solution courante et celle calculée en terme de distance parcourue. Les résultats améliorent les deux principaux objectifs des collectes : le nombre de véhicules et le temps total des tournées. La même année, Mourao et al. [103], présentent une approche "route first-cluster second" pour un problème de collecte des déchets modélisée à l'aide d'un CARP, qui crée une tournée géante dans un premier temps puis décompose avec des méthodes de bornes inférieures en plusieurs tournées faisables selon la capacité du camion.

Shih et al. [128] travaillent sur un problème de collecte de déchets "dangereux" dans les hôpitaux et ont développé un algorithme en deux phases pour l'organisation et la planification de la collecte. Dans une première phase, l'algorithme crée classiquement des tournées à l'aide d'une méthode de programmation dynamique et dans une seconde phase, un algorithme affecte les jours de collecte aux tournées formées.

Dans Toth et al. [138], les auteurs proposent une analyse de dix cas pratiques de collecte des déchets mis en œuvre dans un contexte urbain. Cette analyse met en avant les caractéristiques et les avancées qui ont pu être faites pour ces problèmes pratiques à l'aide des méthodes développées pour chacun. Cette analyse pointe également que les deux méthodes de modélisation sont utilisées (CARP et VRP) mais que le modèle de CARP est un peu moins utilisé.

En 2002, Baptista et al. [8] étudient un problème de collecte périodique des containers de papier à recycler en proposant une résolution en deux temps. Tout d'abord ils organisent les tournées de

véhicules puis ils affectent ces tournées aux jours possibles. Cette solution est ensuite améliorée par des mouvements locaux comme "String-exchange". Angelelli et al. [6], en 2002, proposent un modèle d'estimation des coûts opérationnels pour la collecte des déchets et montrent l'importance de cette estimation pour choisir le système utilisé pour la collecte des déchets.

En 2003, Minciardi et al. [99] tiennent compte du réseau routier exact, c'est-à-dire qu'ils prennent en compte toutes les routes du réseau. Ils proposent une méthode de résolution pour le processus de décision pour minimiser le temps total de parcours des tournées, basée sur des "quartiers". Cette même année, Labadi [86] étudie les tournées sur arcs et propose un algorithme génétique pour gérer la collecte des déchets avec une demande non fixe sur les arcs (selon le jour de collecte) et elle propose d'utiliser l'extension concernant la gestion des tournées périodiques.

En 2004, Koushki et al. [83] proposent d'étudier l'efficacité d'une collecte réalisée au Koweït. Ils mettent en place des indicateurs et une analyse comparative des coûts. Toujours en 2004, Amponsah et al. [5] proposent une étude pour gérer la collecte des déchets des pays en voie de développement en tenant compte des coûts et des impacts environnementaux. Pour cela, ils ont développé une heuristique constructive de prévision dédiée à ce problème.

Le but de Teixeira et al. [136], 2004, est de concevoir des tournées de collecte planifiées sur un mois afin que tous les types de déchets recyclables soient collectés. Leur problème est résolu en 3 phases : d'abord définir les zones géographiques pour chaque véhicule, puis définir le type de déchets à collecter enfin concevoir les tournées de collecte.

Dans son papier, Kim et al. [79] en 2006, proposent d'étudier l'élaboration de circuits de collecte en se basant sur la méthode d'insertion de Solomon en prenant en compte quatre objectifs : les deux objectifs classiques de la collecte des déchets soit la minimisation du nombre de véhicules et la minimisation du temps total de trajet. Il rajoute aussi un objectif de compacité des tournées et d'équilibre entre ces tournées ; ces derniers objectifs étant traités à l'aide d'un algorithme basé sur le groupement de capacité. Dans cette étude, les auteurs prennent en compte également le temps de pause des chauffeurs-rippers. La même année, Nuortio et al. [105] ont développé une métaheuristique appliquée à la collecte des déchets en Finlande, basée sur des méthodes de recherche locale et d'autres métaheuristicues dans lesquelles des méthodes d'accélération et de réduction de mémoire ont été ajoutées. Lacomme et al. [89], en 2006, proposent d'étudier la collecte des déchets pour la ville de Troyes en utilisant une modélisation sur les arcs (Arc Routing Problem) et résout ce problème à l'aide d'un algorithme évolutionnaire qui prend en considération deux objectifs : la minimisation de la distance totale et la minimisation de la tournée la plus grande.

En 2007, Kim et al. [80] considèrent le problème de façon déterministe et de façon stochastique. Ils ont développé deux versions d'algorithmes pour gérer chaque type de problème et montrent que leurs algorithmes sont performants et améliorent les résultats donnés par des méthodes plus conventionnelles.

En 2008, Bautista et al. [10] poursuivent une étude entamée en 2004 [11], et utilisent une méthode basée sur les colonies de fourmis pour gérer la collecte des déchets dans une partie de Barcelone. Cette méthode est combinée avec une heuristique de construction et une recherche locale.

L'ensemble de ces articles démontrent bien les besoins des collectivités pour améliorer les performances de leur système de collecte et l'effort fourni pour résoudre ces différents problèmes pratiques. Cependant bien que certaines contraintes soient communes et présentes dans la majorité des articles, de nombreuses contraintes sont issues directement du "terrain" et sont très souvent spécifiques à une étude. C'est pourquoi il semble difficile de transposer directement une méthode de résolution d'un cas à un autre et de généraliser ces méthodes.

Nous pouvons noter dans cette bibliographie l'importance donnée aux modélisations plus agrégées de type VRP par rapport à celles de type CARP. En effet, il semble plus simple de modéliser des nœuds regroupant plusieurs points effectifs de collecte pour éviter la gestion d'un volume de données qui devient vite très important s'il s'agit de modéliser l'ensemble du réseau de manière détaillée.

Nous proposons de résumer les caractéristiques des études dans les tableaux (2.1, 2.2 et 2.3), en donnant les problèmes de collecte étudiés, la modélisation utilisée (CARP ou VRP), les objectifs et les méthodes de résolution.

Dans l'ensemble de ces études, nous avons pu constater que l'objectif de coût est prédominant, c'est l'élément essentiel pour construire des tournées. Nous n'avons pas rencontré d'études manipulant un objectif associé à des ensembles de tournées. Par la suite nous allons proposer d'ajouter un objectif, dit de **stabilité** qui va permettre de qualifier des distances entre des ensembles de tournées de manière à avoir une vision de leur éloignement respectif.

Dans notre étude, nous nous intéressons uniquement au problème de collecte des ordures ménagères en porte à porte inspiré par une collectivité locale partenaire. Nous allons voir dans la suite de ce chapitre plus en détails l'organisation de la collectivité partenaire, les objectifs la concernant et les problématiques d'organisation de collecte qu'elle se pose.

2.2. État de l'art sur la collecte des déchets

Référence	Problématique	Modèle utilisé	Objectifs	Méthodes résolution
[107]	Étude des circuits de collecte à Singapour	VRP	minimiser la distance parcourue	heuristique de "route-first cluster-second"
[75]	Dimension de la taille de la flotte pour la collecte des huiles usagers en Belgique	VRP	minimiser la taille de la flotte	modèle mathématique de résolution
[98]	Étude sur l'optimisation possible des trajets pour les rippers en utilisant différentes stratégies et notamment le déplacement de containers	VRP	Minimiser la distance parcourue	Développement d'un système de calcul dédié : ARC
[85]	Étude des différentes possibilité de collecte et de transport des déchets pour la ville de Bruxelles	VRP	Minimiser les coûts de transport	combinaison entre une méthode de recherche opérationnelle et un système d'ingénierie
[52]	Étude dynamique des circuits de collecte à Chicago suivant le nombre de visites au dépôt de transfert	VRP	Maximiser le niveau de service	utilisation d'un modèle de Markov pour un algorithme dynamique qui fait varier le nombre de visites au dépôt de transfert
[27]	Système d'évaluation de différentes alternatives pour la gestion des déchets à Taiwan		Développer un modèle multi-objectif pour intégrer la recherche d'itinéraires au sein d'un système	
[82]	Étude sur l'adaptation des techniques connues pour la gestion des déchets dans un nouveau contexte : pour les pays en voie de développement		Minimiser les distances	

TAB. 2.1 – Récapitulatif de quelques exemples de collecte des déchets

2.2. État de l'art sur la collecte des déchets

Référence	Problématique	Modèle utilisé	Objectifs	Méthodes résolution
[19]	Étude de la collecte des déchets recyclables dans un contexte particulier (dans une université)	VRP	Minimiser les distances	Algorithme en deux-phases : affectation des jours de collecte aux points du réseau et résolution du problème de VRP sur chaque jour
[140]	problème de collecte des déchets à Hanoi	VRSP	Minimiser les distances	Algorithme de construction suivi d'algorithmes d'amélioration
[103]	problème de collecte des déchets dans un quartier de Lisbonne	CARP	Minimiser les coûts	présentation de deux bornes inférieures et d'une heuristique en trois phases pour générer une solution proche de la borne inférieure
[128]	étude sur le traitement des déchets infectés hospitaliers	PVRP	Minimiser les coûts	algorithme en deux phases : une première phase qui crée des circuits de collecte et une seconde phase qui affecte les tournées aux jours dans la semaine
[8]	Étude de la collecte des papiers à recyclés au Portugal	PVRP	Minimiser les coûts	Algorithme en deux-phases : affectation des jours de collecte aux points du réseau et résolution du problème de VRP sur chaque jour
[6]	Étude de trois différentes situations de collecte des déchets : suivant le nombre de rippers demandés	PVRP	Minimiser les coûts	
[99]	Étude de la collecte des déchets en Italie	CARP	Minimiser le temps total des tournées	Algorithme de construction de solution par quartier

TAB. 2.2 – Récapitulatif de quelques exemples de collecte des déchets

2.2. État de l'art sur la collecte des déchets

Référence	Problématique	Modèle utilisé	Objectifs	Méthodes résolution
[5]	Étude de la collecte, du transport et des point de dépôt des déchets dans les pays en voie de développement	VRP	Minimiser les coûts et les impacts environnementaux	Algorithme de construction suivi d'algorithmes d'amélioration
[136]	Étude de la collecte des déchets recyclables au Portugal, les déchets sont de trois types différents	PVRP	Minimiser la distance totale parcourue sur la totalité de l'horizon d'étude	Algorithmes à trois-phases : définition des zones géographiques à collecter par véhicule, définition du type de déchets à collecter, définition des circuits de collecte.
[79]	Étude de la collecte des déchets dans la réalité en tenant compte des différents point de dépôt et du déjeuner des rippers	VRPTW	minimiser le nombre de véhicules, les temps de transport, la compacité des routes et l'équilibre des charges	Algorithme basé sur les "cluster"
[105]	Étude de la collecte des déchets en Finlande	VRP	Minimiser les distances de parcours	Algorithme de construction suivi d'algorithmes d'améliorations
[89]	Étude de la collecte des déchets à Troyes (France)	CARP	Minimiser les distances de parcours et la plus grande tournée	Algorithme génétique multi-objectif
[80]	Étude de la collecte des déchets à Seoul (Corée)	SVRP	Minimiser les coûts	deux types d'algorithmes : un algorithme déterministe et un algorithme stochastique
[10]	Étude de la collecte des déchets dans un quartier de Barcelone (Espagne)	MCARP	Minimiser les coûts	Algorithme de colonie de fourmis

TAB. 2.3 – Récapitulatif de quelques exemples de collecte des déchets

2.3 Contexte de l'étude

2.3.1 Partenariat

Pour mieux appréhender le domaine de la collecte des déchets ménagers, nous nous appuyons sur un partenariat avec une collectivité locale de la région toulousaine : la Communauté d'Agglomération du Muretain (CAM). Cela nous permet d'avoir des problèmes concrets et de mieux cerner la réalité du terrain. Dans un souci de généralisation, nous nous appuyons également sur un soutien de l'ADEME (Agence de l'Environnement et de la Maîtrise de l'Énergie). L'ADEME est une agence publique qui a comme objectif de soutenir et d'accompagner des entreprises ainsi que des collectivités locales en proposant des soutiens au développement durable. L'un des domaines de compétences de l'ADEME est celui des déchets [2]. L'apport de l'ADEME à ce travail est de veiller à la possibilité de développements futurs pour adapter notre approche à d'autres collectivités.

Une collectivité locale regroupe plusieurs communes. Elle sert à mutualiser certaines compétences et proposer à tous les usagers un service homogène sur l'ensemble du territoire concerné. Le domaine d'action des collectivités est vaste et variable (eau, déchets, transport scolaire etc.). En ce qui concerne la collecte et le transport des déchets, le but est de proposer un service de collecte en minimisant les coûts et les impacts environnementaux, tout en respectant les réglementations sur les conditions de travail des salariés et en offrant une bonne qualité de service aux usagers. Le mode de gestion des collectes est une décision stratégique : soit les collectivités choisissent de sous-traiter la collecte et le transport des déchets, soit elles choisissent de gérer les collectes en régie et doivent dans ces cas là posséder une flotte de véhicules, des équipes, et des structures de rassemblement des déchets ainsi que les moyens nécessaires pour mettre en place la gestion quotidienne des déchets.

La CAM regroupe 14 communes situées à la périphérie de Toulouse correspondant en 2007 à environ 70 000 habitants répartis sur 18 200 hectares.

Le tableau (2.4) donne la liste des communes faisant partie de la CAM ainsi que leur nombre d'habitants et leur superficie. Dans le périmètre de la CAM, se retrouve une majorité de communes rurales mais aussi quelques communes urbaines.

2.3.2 Fonctionnement général de la collectivité locale étudiée

La CAM est une collectivité relativement "jeune", les communes se sont regroupées depuis 2004. Elle a de nombreuses responsabilités dont celle de la gestion de la collecte, du transport et du traitement des déchets ménagers pour l'ensemble des villes intégrées à la collectivité.

La CAM a fait le choix de gérer la totalité de ses collectes en régie. Elle met donc en place elle-même l'ensemble des tournées, gère sa propre flotte de véhicules et affecte les employés au ramassage des ordures ménagères.

Le volume total des déchets pris en charge par la CAM en 2007 était d'environ 42 000 t dont 49% de déchets ménagers, 12% de recyclables et 39% en déchetterie. Dans ce travail, nous nous intéresserons

Nom	Nombre d'habitants	Surface
Eaunes	4 241	1 496
Labarthe sur Lèze	4 758	1 003
Labastidette	1 785	627
Lavernose-Lacasse	2 276	1 732
Muret	23 400	5 800
Pinsaguel	2 581	520
Pins-Justaret	4 297	442
Portet sur Garonne	9 523	1 619
Roquettes	3 416	336
St Clar de Rivière	1 093	1 067
St Hilaire	965	633
St Lys	7 463	2 130
Saubens	1 723	601
Villate	754	182

Table 2.4: Les différentes communes de la CAM

uniquement à la collecte des ordures ménagères (OM).

Les communes qui ont une forte population sont divisées en secteurs, les communes plus petites sont soit collectées dans une seule tournée soit regroupées dans une même tournée. Par exemple la commune de Muret est divisé en 7 secteurs, celle de Portet-Sur-Garonne en 3 secteurs et celle de Saint-Lys en 2 secteurs; les communes de Saint-Clar de Rivière et de Labastidette ainsi que celles de Lavernose-Lacasse et de Saint-Hilaire sont collectées respectivement dans une même tournée.

Pour la collecte des OM, les maisons particulières sont équipées de bacs individuels, les logements verticaux disposent quant à eux de bacs collectifs et les collectes sont organisées en porte-à-porte.

L'ensemble du territoire de la CAM est divisé en deux parties collectées séparément à des jours différents. En début de cette étude, les ordures ménagères étaient collectés deux fois par semaine pour chaque commune (cette fréquence de collecte est nommée c_2 : elle a lieu soit le lundi et le jeudi sur la première partie du territoire, soit le mardi et le vendredi sur la seconde partie du territoire) ; seuls certains secteurs de la commune de Saint-Lys étaient collectés une seule fois par semaine pour les ordures ménagères (cette fréquence de collecte est nommée c_1 : elle a lieu en début de semaine, soit le lundi pour la première partie du territoire ou le mardi pour la seconde partie du territoire). Les tournées sont au nombre d'une dizaine par jour de collecte pour les OM. La CAM réfléchit actuellement à généraliser la collecte unique des OM dans la semaine pour les habitations particulières.

Les contraintes considérées pour les collectes des OM sont les suivantes :

- collecter toutes les habitations sachant que certaines d'entre elles sont contraintes par des plages horaires (par exemple pour les établissements scolaires, ou bien pour éviter les zones de trafic dense aux heures de pointe),
- respecter les capacités maximales de chaque véhicule,
- respecter le nombre de véhicules disponibles,
- respecter le temps de travail de chaque équipe par journée de collecte (toutes les équipes doivent

respecter le même temps de travail),

En pratique, les véhicules partent tous du même dépôt et terminent les trajets au même point de vidage des véhicules (centre de transfert des déchets). On n'étudie pas dans cette thèse le problème de vidage intermédiaire des véhicules. Les véhicules retournent tous ensuite du vidage au dépôt. Le réseau routier va être agrégé de manière à conserver des nœuds correspondant à des "quartiers de collecte". Ces quartiers sont issus du réseau routier sur lequel un travail de prétraitement va être effectué pour conserver de manière logique des ensembles de points à collecter en même temps (par exemple, les habitations d'un lotissement sont collectées à la suite par le même véhicule). Le problème de collecte de la CAM correspond alors à un VRPTW. Sur ce VRPTW, il s'agit d'organiser des tournées partant du dépôt, arrivant au point de décharge et passant par l'ensemble des nœuds définis pour la journée de travail, sachant que certains sont contraints par une fenêtre de temps.

Les tournées mises en place ont été héritées de la situation précédant la création de la CAM; elles ont progressivement évoluées depuis 2004 pour coller au mieux aux contraintes du terrain mais n'ont pas, à ce jour, fait l'objet d'une étude d'optimisation.

Une particularité de la collecte des déchets de la CAM est qu'en moyenne la quantité des déchets à collecter en fin de semaine est moins importante qu'en début de semaine et qu'il y a moins de bacs à collecter. Par ailleurs, le taux de présentation des bacs est inférieur en fin de semaine. Les tournées du jeudi et du vendredi sont donc différentes de celles de début de semaine. Ceci permet de diminuer le nombre de camions à sortir en fin de semaine et donne de la flexibilité pour les jours de repos des chauffeurs-rippers.

Le premier objectif de notre travail sera de proposer des tournées de début et de fin de semaine tout en limitant les variations entre ces deux jours de collecte afin de garantir une meilleure qualité de service et une meilleure efficacité dans le déroulement des tournées.

Une autre particularité de la collectivité étudiée est de devoir s'adapter à une augmentation de population relativement importante (+2% par an). Il faut également prendre en compte le fait que, malgré la volonté publique et les actions de sensibilisation, la quantité de déchets augmente en moyenne de 2% par an et par habitant. Ces évolutions combinées de population et de quantité de déchets à collecter se traduisent en pratique par des variations de quantité à collecter ou par l'apparition de nouveaux secteurs de collecte, ce qui nécessite une actualisation régulière des tournées existantes sur un horizon de plusieurs mois. Cette adaptation régulière peut mener à la longue à la mise en place de nouveaux circuits de collecte.

Cependant, dans un souci d'efficacité de la collecte et de qualité de service, la collectivité souhaite non seulement limiter les coûts liés aux collectes mises en place mais également limiter l'impact des variations des informations de collecte. Ces deux objectifs sont primordiaux pour la collectivité. Elle souhaite adopter des tournées de collecte de manière à absorber les nouvelles informations en cherchant un compromis entre les coûts engagés et les variations de tournées impliquées. Par exemple, une solution d'évolution des tournées qui réduirait les coûts mais qui proposerait des variations trop importantes

ne serait pas privilégiée par rapport à une solution qui proposerait un coût correct et des variations moindres.

Le deuxième objectif de notre travail sera de proposer des adaptations des tournées existantes pour faire face aux variations sur les informations de collecte.

Remarque sur la suite de l'étude : nous allons étudier des problèmes de collecte liés à la CAM, mais afin d'évaluer les méthodes proposées, nous utiliserons des instances de la littérature. En effet, les données réelles n'étant pas disponibles, nous n'avons pas pu tester nos méthodes en situation réelle. Nous allons donc devoir adapter au mieux les instances aux problèmes de la CAM.

2.3.3 Objectifs pour l'organisation des collectes

Pour notre collectivité locale, l'enjeu de la gestion quotidienne des déchets est de proposer un service de qualité aux usagers tout en minimisant ses coûts de gestion et en garantissant également de bonnes conditions de travail aux employés. Voyons plus en détails ces enjeux en ce qui concerne la construction des tournées :

- la qualité de service : cette notion se traduit ici par la collecte effective de tous les points nécessitant (notion déjà présente dans le VRP classique),
- la minimisation des coûts : dans cette étude, le coût est considéré proportionnel à la distance parcourue, autrement dit nous allons traduire la minimisation des coûts par la minimisation de la somme des distances de l'ensemble des tournées,
- l'équilibrage des tournées : les tournées doivent être équilibrées en termes de temps de travail et de pénibilité (temps de parcours de chacun des circuits, temps en haut-le-pied¹, quantité à collecter) : cet objectif sera pris en compte dans la conception des méthodes d'élaboration des tournées.

L'enjeu environnemental présent dans toutes les études sur les déchets, se limitera ici à la minimisation de consommation de carburant, ce qui reviendra à minimiser les distances parcourues par l'ensemble des véhicules, c'est-à-dire dans notre cas à la minimisation des coûts.

En résumé, l'objectif de collecte se traduira par la minimisation des distances de parcours dans notre étude.

2.3.4 La stabilité des tournées de collecte

Comme exposé ci-dessus, la collectivité cherche à limiter les variations sur les itinéraires des différents véhicules de collecte quitte à, par exemple, sortir plus de véhicules que la solution la moins coûteuse pour conserver des itinéraires relativement similaires pour les différentes équipes.

Afin d'étudier ce phénomène, nous allons définir la notion de **stabilité**. La stabilité vise à qualifier les solutions en termes de variation : les modifications apparaissant entre de nouvelles et d'anciennes tournées vont être caractérisées à l'aide de cette notion. Comme vu dans l'état de l'art ci-dessus, nous

¹déplacement du véhicule sur le réseau routier sans collecter de déchets

n'avons pas rencontré d'études sur la comparaison d'ensemble de tournées en termes d'éloignement de solutions (calcul des écarts entre deux solutions).

Le but de la collectivité est à la fois d'organiser et de réorganiser (en raison des évolutions de certaines informations) les différentes tournées de collecte sans trop perturber la solution de départ. La solution courante doit être viable pendant un certain temps (environ 1 an) et ne subir que des modifications non fondamentales et ceci notamment pour garder une constance envers les usagers et les employés qui n'accepteraient pas de devoir changer fréquemment leurs habitudes. Ainsi, la stabilité sera un indicateur de la qualité des nouvelles tournées par rapport à celles existantes.

La stabilité est évaluée entre deux solutions, c'est-à-dire entre deux ensembles de tournées. Chaque ensemble de tournées est constitué de certains nœuds et arcs du graphe. Le premier ensemble de tournées représente la solution courante (la solution qui ne doit pas subir trop de perturbations) notée : S_{ref} , et le second ensemble de tournées représente la solution ayant subi des évolutions, notée : S_{evol} .

La stabilité va être caractérisée selon trois niveaux : (1) de **manière globale**, par des écarts sur les caractéristiques générales des deux ensembles de tournées, (2) **au niveau des usagers**, en prenant en compte les perturbations engendrées de leur point de vue et (3) **au niveau des employés** en tenant compte des perturbations engendrées par les nouvelles collectes sur leurs itinéraires quotidiens.

Nous allons affiner chacun de ces niveaux de stabilité :

1. stabilité globale : elle se traduit par
 - l'écart sur le nombre de véhicules nécessaires à la collecte des ensembles S_{ref} et S_{evol} ,
 - l'écart sur les distances totales des tournées de S_{ref} et S_{evol} ,
 - l'écart entre les temps de parcours total des tournées de S_{ref} et S_{evol} .
2. stabilité du point de vue des usagers : elle est exprimée par les écarts sur les horaires effectifs de ramassage des bacs entre les nœuds communs aux ensembles de tournées S_{ref} et S_{evol} .
3. stabilité du point de vue des employés : elle sera exprimée pour chacune des tournées k commune à S_{ref} et S_{evol} par :
 - la variation de distance parcourue par la tournée k ,
 - la variation du temps de travail de la tournée k ,
 - les variations de quantités collectées par la tournée k ,
 - les variations sur les ensembles de points collectés pour chaque tournée k ,
 - les variations sur l'ordre de passage dans les nœuds collectés pour chaque tournée k .

Chacune de ces mesures va permettre de comparer différentes solutions en termes de stabilité des tournées par rapport à une solution courante.

2.3.5 Deux problématiques de construction de tournées

Le premier problème posé par la CAM concerne l'**organisation hebdomadaire des collectes** sachant que certains nœuds ne sont collectés qu'une seule fois (en début de semaine) et d'autres deux

fois (en début et fin de semaine) et que la charge de collecte est moins importante en fin de semaine par rapport à celle en début de semaine. Actuellement le nombre de points de collecte en c_1 (nœuds collectés une fois par semaine) est moins grand que le nombre de points en c_2 (nœuds collectés deux fois par semaine). La collectivité cherchant à réduire la fréquence de collecte des OM, cela signifie qu'à terme le nombre de nœuds collectés en c_1 sera plus important que le nombre de nœuds collectés en c_2 . Selon l'hypothèse émise sur le nombres de nœuds collectés en c_1 par rapport aux nombres de nœuds collectés en c_2 , nous allons proposer différentes méthodes de construction des tournées de collecte hebdomadaire.

Il est important pour la collectivité, que les collectes de début et de fin de semaine ne diffèrent pas trop, nous avons donc défini pour cela un objectif de stabilité que nous nommons ici **stabilité hebdomadaire**.

Le deuxième problème étudié est lié aux évolutions des informations de collecte : la collectivité que nous étudions étant soumise à une augmentation régulière du nombre d'habitants et de la quantité de déchets.

La collectivité souhaite pouvoir piloter des **variations dans l'organisation des collectes** pour répondre aux évolutions des informations tout en limitant les perturbations sur les tournées existantes. Nous allons pour cela introduire un objectif de stabilité que nous nommons **stabilité dynamique**.

Pour étudier la construction de tournées dans un contexte hebdomadaire ou dynamique nous utilisons un objectif de stabilité défini entre deux ensembles de tournées pour mesurer l'écart entre une solution courante de collecte et une nouvelle solution de collecte. Cet objectif de stabilité est défini pour les deux problèmes qui nous intéressent :

- Objectif de stabilité “hebdomadaire” : il doit permettre de minimiser des écarts entre les tournées de début de semaine et celles de fin de semaine. Cette stabilité “hebdomadaire” traduit une stabilité globale sur l'ensemble des tournées et une stabilité du point de vue des usagers et/ou des employés.
- Objectif de stabilité “dynamique” : il vise à minimiser les différences entre les tournées courantes et les nouvelles tournées calculées pour répondre à des évolutions des données. Cette stabilité “dynamique” traduit également une stabilité globale sur l'ensemble des tournées ainsi qu'une stabilité du point de vue des usagers et/ou des employés.

2.4 Conclusion

Dans ce chapitre, nous nous sommes intéressés aux différents types de collecte des déchets. Nous avons effectué un état de l'art de différentes études de cas pratiques de collecte des déchets. Ces études proviennent de cas réels issus de différents pays et s'intéressent à des caractéristiques et à des spécificités dues à chaque cas. Il semble difficile de généraliser ces différentes études et de proposer des méthodes génériques permettant de traiter l'ensemble des problématiques de collecte, notamment

en raison des caractéristiques sociales, environnementales, de sociétés inhérentes à chaque pays. Nous avons pu remarquer que dans l'ensemble de ces études l'objectif principal étudié est celui des coûts engendrés par les collectes. Pour ce qui est des études multi-objectifs, aucune ne porte sur un objectif de stabilité, c'est à dire sur la prise en compte de variations entre deux ensembles de tournées. Ceci nous a amené à développer et proposer une étude plus approfondie sur cet aspect dans l'ensemble de ce mémoire.

Dans une seconde partie de ce chapitre, nous avons mis en relief, le contexte de notre travail, en présentant la collectivité partenaire de ce mémoire et ces caractéristiques.

L'étude des collectes de déchets ménagers d'une collectivité locale et de ces spécificités nous a conduits à définir deux nouveaux problèmes de collecte :

- Un problème de collecte hebdomadaire dans lequel les tournées de début et de fin de semaine doivent être le moins différentes possibles alors que les nœuds à collecter peuvent varier ainsi que les quantités de déchets.
- Un problème de collecte dynamique dans lequel de nouvelles tournées doivent être élaborées pour faire face à des évolutions de données mais tout en évitant de remettre complètement en cause les tournées courantes.

Pour évaluer les variations entre deux ensembles de tournées que ce soit pour un problème hebdomadaire ou dynamique, nous avons proposé d'utiliser un objectif de stabilité couplé avec un objectif usuel de coût de collecte. Nous avons pour cela proposé de caractériser l'objectif de stabilité à différents niveaux (global, usagers et employés).

Les deux problèmes identifiés : collecte hebdomadaire et collecte dynamique sous les objectifs de coût et de stabilité font l'objet des prochains chapitres de ce mémoire.

Chapitre 3

Étude bi-objective de la stabilité hebdomadaire

Dans ce chapitre, nous nous intéressons plus spécifiquement au problème d'organisation hebdomadaire des collectes. Ce problème consiste à déterminer deux ensembles de tournées pour chaque semaine : un premier ensemble associé aux tournées de début de semaine, noté S_{deb} , et un second représentant celles de fin de semaine, noté S_{fin} . Les deux objectifs considérés sont la minimisation des coûts de collecte exprimés en termes de distance totale parcourue et la minimisation des écarts entre les deux ensembles de tournées aussi appelé stabilité hebdomadaire. Nous proposons des méthodes heuristiques intégrant les objectifs de coût et de stabilité pour la création de ces circuits. Les performances de ces méthodes sont évaluées sur des instances de la littérature.

3.1 Description du problème

La collecte de l'ensemble des communes sur la CAM ne peut être faite en un seul jour. En effet, le nombre de points à collecter est trop important par rapport aux ressources de la CAM (nombre de véhicules disponible pour la collecte, nombre d'employés). Il est donc nécessaire d'étaler l'ensemble des collectes sur plusieurs jours. Pour cela, le territoire de la CAM a été partitionné en deux parties : une première partie du territoire dont les habitations sont collectées le lundi et le jeudi, et une seconde partie dont les habitations sont collectées le mardi et le vendredi. Ce partitionnement donne deux "clusters" indépendants et fixes qui correspondent chacun à des zones géographiques recouvrant à eux deux la totalité de la collectivité. Dans notre problème, nous ne cherchons pas à les obtenir, ils sont donnés. Ils correspondent à des contraintes géographiques particulières de la collectivité et sont aussi fonction de la localisation du point de vidage. Dans chacun de ces clusters, il peut exister des points de collecte à ramasser une fois par semaine et des points de collecte à ramasser deux fois par semaine.

Nous allons, par la suite traiter indépendamment chacun de ces clusters. En considérant que dans chaque cluster, il y a deux jours de collecte (un jour de collecte en début de semaine et un jour de collecte en fin de semaine) ce qui correspond au problème de collecte hebdomadaire que nous allons

traiter dans ce chapitre.

Pour le problème de collecte des déchets hebdomadaire, il est nécessaire de disposer d'une première solution qui constitue les circuits de base sur lesquels la construction de la deuxième collecte va se baser. Nous devons également tenir compte des différentes fréquences de collecte. La définition de la fréquence de collecte est un paramètre d'entrée du problème, c'est aux responsables des communes de la collectivité de définir ces fréquences de collecte. Par exemple, en ce qui concerne la collectivité partenaire, il existe deux fréquences de collecte, différentes même en fonction des quartiers des communes. Dans notre cas, certains points doivent être collectés deux fois (c_2) et d'autres une seule fois (c_1). Cette fréquence a de l'importance puisque l'organisation des collectes varie selon le rapport entre le nombre de nœuds collectés une fois par semaine et le nombre de nœuds collectés deux fois par semaine. Si le nombre de nœuds à collecter en c_1 est peu important (autour de 10/20%) alors tous ces nœuds vont être collectés en début de semaine afin de dégager des jours de repos en fin de semaine pour les employés. Par contre si ce nombre est important (de 60 à 70%) alors ces nœuds seront répartis sur l'ensemble de la semaine.

Un autre paramètre est affecté par la fréquence de collecte du nœud considéré, c'est celui de la quantité de déchets à collecter. En effet, d'après l'expérience de la collectivité, la quantité de déchets présentée en fin de semaine est bien inférieure à celle présentée en début de semaine (en raison du week-end). Or si la fréquence de collecte est de une fois par semaine, la quantité à collecter sera plus importante pour cet unique passage.

Le but de ce chapitre est de proposer une étude comparative entre différentes méthodes de construction pour deux ensembles de tournées de véhicules sur une période d'une semaine : celui de début de semaine et celui de fin de semaine.

Dans ce problème nous allons prendre comme hypothèse pour une première partie de l'étude que les nœuds en c_2 sont majoritaires sur le territoire. Puis dans une seconde partie de l'étude, nous allons prendre en compte l'hypothèse inverse, c'est-à-dire que les nœuds en c_1 sont majoritaires sur le territoire.

Ces deux hypothèses ont été induites par la situation actuelle de la collectivité partenaire, qui collecte une majorité d'usagers deux fois par semaine mais dont la volonté est de tendre vers une collecte unique par semaine pour un grand nombre d'usagers.

Pour chaque hypothèse considérée, nous allons proposer différentes méthodes de construction des tournées de début et de fin de semaine. Ces méthodes sont élaborées afin d'obtenir non seulement des solutions dont les coûts sont corrects mais dont la stabilité reste correcte également. En effet, l'étude de ces deux objectifs doit permettre de faire ressortir des solutions dont les caractéristiques peuvent être à la fois différentes et intéressantes en termes de coût et de stabilité. Il est difficile de mettre un ordre sur ces objectifs, on cherche à la fois à trouver des solutions qui ne donnent pas un coût trop important quitte à modifier les tournées, mais également on préfère privilégier une solution qui reste proche de la première quitte à dégrader la fonction coût. L'équilibre entre ces deux objectifs est à déterminer et peut être variable selon les interlocuteurs. Dans le cas d'une collectivité locale qui fonctionne en régie, le nombre d'équipes est fixe par jour, donc le nombre de véhicule est prédéterminé pour le début de

semaine mais peut être plus souple pour la fin de semaine. En effet, la collectivité privilégie certaines fois de sortir plus de véhicules, afin de conserver des itinéraires similaires. Tout ceci a de l'importance sur le fonctionnement et les objectifs envisagés.

3.2 Modèle mathématique

Le modèle décrivant l'organisation de la collectivité partenaire est basé sur celui du "Vehicle Routing Problem with Time Windows (VRPTW)" vu au chapitre 1. Dans cette partie nous allons compléter ce modèle en précisant la fréquence de collecte des nœuds et l'objectif de stabilité.

Par la suite, nous allons adopter la notation suivante :

- Z_{c_1} : l'ensemble des nœuds collectés une fois dans la semaine,
- Z_{c_2} : l'ensemble des nœuds collectés deux fois dans la semaine.

Le modèle de VRPTW est décrit au chapitre 1, nous allons juste rappeler les variables de décision du modèle qui permettent de caractériser une solution :

- $x_{ij}^k = 1$ si l'arc (z_i, z_j) est dans la tournée du véhicule k et 0 sinon.
- $y_j^k = 1$ si le nœud z_j est visité par le véhicule k et 0 sinon.

Une solution S à un problème de VRPTW est caractérisée par l'ensemble de ses variables (x_{ij}^k, y_j^k) .

3.2.1 Fréquence de collecte

Dans notre problème, la fréquence et les combinaisons de jours possibles vont être définies de la manière suivante :

- $f_i = 1$ si $z_i \in Z_{c_1}$ et $f_i = 2$ si $z_i \in Z_{c_2}$.
- R_i : combinaison de passage possible sur le nœud z_i , avec $R_i = \{debut, fin, debut - fin\}$

Avec respectivement *debut* représente le fait que le nœud est collecté en début de semaine (lundi ou mardi), *fin* le fait que le nœud est collecté en fin de semaine (jeudi ou vendredi) et *debut - fin*, le fait que le nœud est collecté en début et en fin de semaine (lundi et jeudi ou mardi et vendredi).

Hypothèse 1 : $|Z_{c_1}| < |Z_{c_2}|$

Cette hypothèse correspond à la situation actuelle de la CAM : il y a plus de nœuds à collecter deux fois par semaine que de nœuds à collecter une seule fois. Le choix a été fait de collecter tous les nœuds en c_1 en début de semaine, il ne reste alors comme combinaison que les possibilités suivantes : $R_i = \{debut, debut - fin\} \forall z_i \in (Z_{c_1} \cup Z_{c_2})$ avec $R_i = \{debut\}$ pour $z_i \in Z_{c_1}$ et $R_i = \{debut - fin\}$ pour $z_i \in Z_{c_2}$.

Ainsi, dans notre situation, l'appartenance ou non d'un nœud à une collecte au début ou de fin de semaine est déjà définie. Nous ne sommes donc pas en présence d'un PVRP mais de deux VRPTW : $VRPTW_{deb}$ et $VRPTW_{fin}$.

Pour le problème $VRPTW_{deb}$, il s'agit de calculer les tournées S_{deb} portant à la fois sur les nœuds en c_1 et sur les nœuds en c_2 ($Z_{c_1} \cup Z_{c_2}$). Pour le problème $VRPTW_{fin}$, il s'agit de calculer les tournées S_{fin} portant uniquement sur les nœuds en c_2 (Z_{c_2}).

Pour calculer S_{deb} et S_{fin} les objectifs utilisés sont :

- minimiser la distance totale parcourue dans S_{deb} ,
- minimiser la distance totale parcourue dans S_{fin} ,
- minimiser les écarts de S_{fin} par rapport à S_{deb} .

Hypothèse 2 : $|Z_{c_1}| > |Z_{c_2}|$

Cette hypothèse a pris naissance en raison d'une volonté de la CAM de réduire le nombre de nœuds à collecter en c_2 pour diminuer les coûts. Nous avons donc comme combinaisons possibles pour la collecte des nœuds : $R_i = \{debut, fin, debut - fin\} \forall z_i \in Z_{c_1} \cup Z_{c_2}$ avec $R_i = \{debut, fin\}$ pour $z_i \in Z_{c_1}$ et $R_i = \{debut - fin\}$ pour $z_i \in Z_{c_2}$.

Dans cette hypothèse, nous sommes bien en présence, d'un $PVRPTW$. Il s'agit d'affecter un jour de collecte pour les nœuds à collecter en c_1 et de calculer les tournées appartenant à S_{deb} et à S_{fin} tout en minimisant les distances totales de parcours de S_{deb} et S_{fin} ainsi qu'en minimisant les écarts entre S_{deb} et S_{fin} .

3.2.2 Modélisation de la stabilité

Afin d'étudier la stabilité hebdomadaire, nous ajoutons au modèle de base les expressions des différents niveaux de stabilité présentés au chapitre 2.

Nous définissons l'objectif de stabilité par les variations observées entre les deux ensembles de tournées S_{deb} et S_{fin} . Pour toutes les méthodes, nous calculons ces variations par la différence entre des caractéristiques de S_{deb} et de S_{fin} .

1. stabilité globale : ses différentes composantes vues au chapitre 2 se calculent comme suit :

- l'écart sur le nombre de véhicules :

$$E_{nbv} = |m_{deb} - m_{fin}|$$

où m_{deb} (respectivement m_{fin}) représente le nombre de véhicules utilisés dans S_{deb} (respectivement S_{fin}).

- l'écart entre la distance totale parcourue par les tournées de S_{deb} et la distance totale parcourue par celles de S_{fin} :

$$E_{dist} = \left| \sum_{i \in Z} \sum_{j \in Z} \sum_{k \in K} d_{ij}^k \cdot x_{ij}^{kdeb} - \sum_{i \in Z} \sum_{j \in Z} \sum_{k \in K} d_{ij}^k \cdot x_{ij}^{kfin} \right|$$

où d_{ij}^k représente la distance de parcours de l'arc (z_i, z_j) .

- l'écart entre la durée totale des tournées de S_{deb} et la durée totale des tournées de S_{fin} :

$$E_{tps} = \left| \sum_{i \in Z} \sum_{j \in Z} \sum_{k \in K} t_{ij}^k \cdot x_{ij}^{kdeb} - \sum_{i \in Z} \sum_{j \in Z} \sum_{k \in K} t_{ij}^k \cdot x_{ij}^{kfin} \right|$$

où t_{ij}^k représente le temps de parcours de l'arc (z_i, z_j) .

1. stabilité du point de vue usager : elle se traduit par l'écart entre l'horaire d'arrivée dans les nœuds de S_{fin} par rapport aux nœuds de S_{deb} :

$$E_h = \sum_{i \in Z} \sum_{k \in K} \left| st_i^{kdeb} - st_i^{kfin} \right|$$

2. stabilité du point de vue employés : elle est mesurée par différents écarts entre les tournées de véhicules k communes à S_{deb} et S_{fin} :

- l'écart entre la distance parcourue par le véhicule k dans S_{deb} et celle effectuée par ce même véhicule dans S_{fin} :

$$E_{dist-emp} = \sum_{k \in K} \left| \sum_{i \in Z} \sum_{j \in Z} d_{ij}^k \cdot x_{ij}^{kdeb} - \sum_{i \in Z} \sum_{j \in Z} d_{ij}^k \cdot x_{ij}^{kfin} \right|$$

- l'écart entre le temps de collecte par le véhicule k dans S_{deb} et celui mis par ce même véhicule dans S_{fin} :

$$E_{tps-emp} = \sum_{k \in K} \left| \sum_{i \in Z} \sum_{j \in Z} t_{ij}^k \cdot x_{ij}^{kdeb} - \sum_{i \in Z} \sum_{j \in Z} t_{ij}^k \cdot x_{ij}^{kfin} \right|$$

- l'écart entre la quantité collectée par le véhicule k dans S_{deb} et celle collectée par ce même véhicule dans S_{fin} :

$$E_{capa} = \sum_{k \in K} \left| \sum_{i \in Z} q_i^{kdeb} \cdot y_i^{kdeb} - \sum_{i \in Z} q_i^{kfin} \cdot y_i^{kfin} \right|$$

- l'écart entre les nœuds visités par le véhicule k dans S_{deb} et ceux visités par ce même véhicule dans S_{fin} :

$$E_{compo} = \sum_{k \in K} \left| \sum_{i \in Z} y_i^{kdeb} - \sum_{i \in Z} y_i^{kfin} \right|$$

- l'écart sur l'ordre de passage dans les nœuds pour un véhicule k dans S_{deb} et S_{fin} :

$$E_{ordre} = \sum_{k \in K} \left| \sum_{i \in Z} \sum_{j \in Z} x_{ij}^{kdeb} - \sum_{i \in Z} \sum_{j \in Z} x_{ij}^{kfin} \right|$$

L'ensemble de ces indicateurs vont définir l'objectif de stabilité et guider le décideur dans le choix d'une solution.

Nous allons présenter différentes méthodes, les résultats qu'elles fournissent ainsi que les conclusions que l'on peut en tirer. Comme évoqué précédemment, nous allons présenter des méthodes dédiées au cas où le nombre de nœuds en c_1 est très inférieur au nombre de c_2 (hypothèse 1) et où le nombre de nœuds c_1 est très supérieur au nombre de nœuds c_2 (hypothèse 2).

3.3 Hypothèse 1 : $|Z_{c_1}| < |Z_{c_2}|$

3.3.1 Méthodes proposées

On se place ici dans l'hypothèse selon laquelle le nombre de nœuds en c_1 est très inférieur au nombre de nœuds en c_2 , en considérant qu'il y a entre 10 et 30 % de nœuds à collecter en c_1 . Selon le fonctionnement actuel de la collectivité partenaire, nous plaçons tous les nœuds à collecter en c_1 dans l'ensemble des tournées de début de semaine (S_{deb}).

La figure (3.1) illustre les deux ensembles de tournées S_{deb} et S_{fin} . Sur cette figure, on retrouve dans S_{deb} l'ensemble des nœuds en c_1 et il y a besoin de 3 véhicules pour collecter l'ensemble des nœuds. Dans S_{fin} , seuls les nœuds en c_2 sont présents, sur cet exemple seulement 2 véhicules sont nécessaires pour effectuer la collecte.

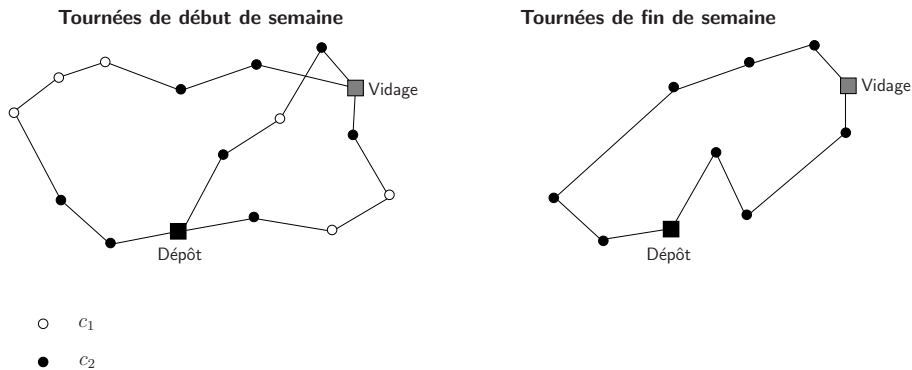


FIG. 3.1 – Schéma représentant la collecte si $|Z_{c_1}| < |Z_{c_2}|$

Sachant qu'il y a moins de nœuds à collecter en fin de semaine qu'en début de semaine et que la quantité de déchets est plus faible en fin de semaine qu'en début, nous pouvons réduire le nombre de véhicules à utiliser dans S_{fin} ($m_{S_{deb}} > m_{S_{fin}}$).

Pour construire les ensembles de tournées S_{deb} et S_{fin} , trois principes sont utilisés :

- construction de l'ensemble S_{deb} puis construction de l'ensemble S_{fin} en se basant sur S_{deb} : ce sont les méthodes 1, 2 et 3 présentées ci-dessous ;
- construction de l'ensemble S_{fin} puis construction de l'ensemble S_{deb} en se basant sur S_{fin} : ce sont les méthodes 4 et 5 présentées ci-dessous ;
- construction de S_{deb} et S_{fin} séparément : méthode 6 présentée ci-dessous.

Pour construire la première solution (S_{deb} ou S_{fin} , selon la méthode utilisée), nous utilisons l'heuristique d'insertion de Solomon décrite dans le chapitre 1.

Cette heuristique peut être utilisée soit en séquentielle (véhicule par véhicule) soit en parallèle. Le choix de construire les tournées de manière parallèle est apparu comme un meilleur compromis pour l'équilibrage des différentes tournées élaborées. La difficulté de l'utilisation de cette heuristique

3.3. Hypothèse 1 : $|Z_{c_1}| < |Z_{c_2}|$

en parallèle est de déterminer initialement le nombre de véhicules nécessaires et les points de départ de chaque tournée. Pour trouver ce nombre, nous utilisons le nombre de véhicules connus pour les instances de la littérature (ou le nombre de véhicules nécessaires au minimum pour collecter la quantité de déchets pour les instances inconnues). Ce nombre ayant déjà fait l'objet de plusieurs études, il est en général celui qui garantit de trouver une solution admissible correct en coût pour construire les premières tournées. Si cela ne permet pas de trouver une solution admissible, nous augmentons ce nombre de 1 jusqu'à trouver une solution admissible. Pour les nœuds de départ, nous les choisissons de manière à ce qu'ils soient les plus éloignés les uns des autres ainsi que par rapport au dépôt.

A la suite de cet algorithme, les méthodes de descente "String-exchange" et "Or-opt" vues au chapitre 1 sont appliquées. La méthode "String-exchange" consiste à améliorer plusieurs tournées en même temps, tandis que la méthode "Or-opt" améliore une tournée à la fois. La procédure est décrite dans l'algorithme 1.

Algorithme 1 Construction des tournées

Données: $VRPTW$

Résultat: S : ensemble de tournées

Heuristique_Solomon_Parallele($VRPTW, S$)

String_exchange(S)

Or_opt(S)

A l'aide de cette première solution, nous allons présenter les méthodes proposées pour construire la deuxième de collecte. Ces méthodes sont toutes des méthodes heuristiques.

Description des méthodes 1, 2 et 3

Ces méthodes se basent sur les tournées construites de début de semaine (S_{deb}) pour élaborer celles de fin de semaine (S_{fin}). Pour cela, les tournées de S_{deb} sont "recopiées" dans S_{fin} en retirant les nœuds en c_1 comme illustré par la figure (3.2). Puis la quantité des nœuds en c_2 est mise à jour.

Cette procédure est commune aux méthodes 1, 2 et 3 et est décrite dans l'algorithme 2.

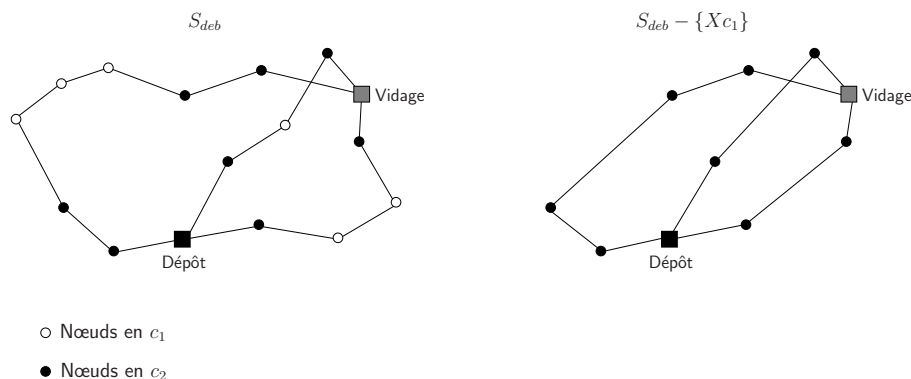


FIG. 3.2 – Suppression des nœuds collectés en c_1

Algorithme 2 Procédure d'initialisation pour les méthodes 1, 2 et 3 : *Initialisation1*

Données: $S_{deb}, VRPTW_{fin}$

Résultat: S

$S \leftarrow S_{deb}$

$Mettre_a_jour(S, VRPTW_{fin})$

La procédure $Mettre_a_jour(S, VRPTW_{fin})$ permet non seulement de retirer les nœuds en c_1 de S_{deb} , mais également met à jour la quantité des nœuds en c_2 à partir de $VRPTW_{fin}$. Cette procédure nous donne ainsi un ensemble de tournées S initialisées pour démarrer la construction des tournées de S_{fin} .

Dans S_{fin} , le nombre de véhicules est potentiellement moins important que dans S_{deb} , pour cela les méthodes 1, 2 et 3 tentent d'éliminer progressivement des tournées.

Pour réduire le nombre de véhicules, nous allons utiliser différentes technique d'élimination de tournées. Pour les méthodes 1 et 2, il va s'agir de sélectionner des tournées à éliminer, pour ce qui est de la méthode 3, il va s'agir de vider les tournées jusqu'à pouvoir en éliminer une. Ces méthodes sont décrites ci-dessous.

Pour tenter d'éliminer une (des) tournée(s), les **méthodes 1 et 2** essayent de transférer (si possible) les nœuds de tournées sélectionnées sur les autres, comme illustré sur la figure (3.3).

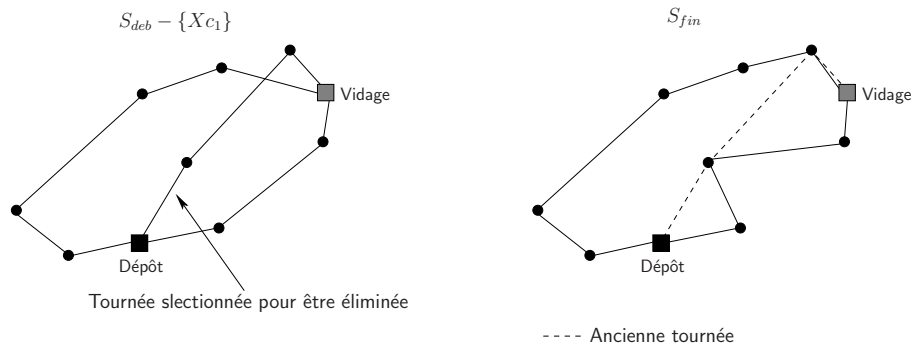


FIG. 3.3 – Transfert des noeuds sur tournées restantes

Afin d'insérer un nœud dans une nouvelle tournée, on choisit l'insertion qui minimise la distance totale de la tournée dans laquelle le nœud s'ajoute (et qui respecte les contraintes du problème). Le principe de ces méthodes est décrit dans l'algorithme 3.

Algorithme 3 Méthodes 1 et 2 pour $Z_{c_1} < Z_{c_2}$ **Données:** $S_{deb}, VRPTW_{fin}$ **Résultat:** S_{fin} $Sol \leftarrow Initialisation1(S_{deb})$ **répéter** $k \leftarrow Select_tournee_elimination(Sol)$ $OK \leftarrow Vider_Tournee(k, Sol)$ **jusqu'à** $OK = Faux$ $S_{fin} \leftarrow Sol$ $Or_opt(S_{fin})$

La procédure $Select_tournee_elimination(Sol)$ permet la sélection de la tournée à éliminer. Pour la méthode 1, c'est la tournée de plus petite quantité qui sera choisie, et pour la méthode 2 c'est une tournée au hasard qui sera choisie.

La procédure $Vider_Tournee(k, Sol)$ permet de transférer chaque nœud z d'une tournée k vers une autre tournée k' telle que la distance dans k' de Sol soit la plus petite possible. Si le transfert d'un nœud z vers une autre tournée n'est pas possible, la procédure retourne la valeur $Faux$; la solution Sol ne peut plus être améliorée.

La **méthode 3** se différencie dans le transfert des nœuds d'une tournée à une autre afin d'essayer de vider des tournées. Dans cette méthode, une tournée est sélectionnée avec une probabilité inversement proportionnelle au nombre de nœuds qu'elle contient. Ainsi une tournée ayant un faible nombre de nœuds aura une plus forte chance d'être sélectionnée pour être éliminée. Par la suite, un nœud est sélectionné au hasard dans la tournée à éliminer et il est transféré dans une nouvelle tournée choisie avec une probabilité proportionnelle à son nombre de nœuds, comme illustré sur la figure (3.4).

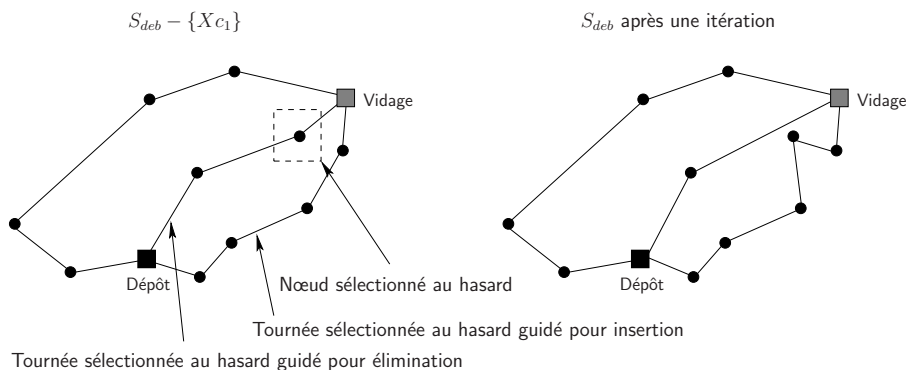


FIG. 3.4 – Déplacement d'un nœud dans la méthode 3

L'insertion du nœud dans la nouvelle tournée est effectuée de manière à minimiser la distance à parcourir. L'algorithme 4 résume la méthode 3.

La procédure $Select_tournee_elimination_hasard_guide(Sol)$ sélectionne une tournée pour en éliminer un nœud. Cette tournée est choisie de manière inversement proportionnelle au nombre de nœuds la constituant. La procédure $Select_noeud_hasard(k)$ sélectionne un nœud au hasard dans une tournée

3.3. Hypothèse 1 : $|Z_{c_1}| < |Z_{c_2}|$

Algorithme 4 Méthode 3 pour $Z_{c_1} < Z_{c_2}$

Données: $S_{deb}, VRPTW_{fin}$

Résultat: S_{fin}

$Sol \leftarrow Initialisation1(S_{deb})$

répéter

$k \leftarrow Select_tournee_elimination_hasard_guide(Sol)$

$x \leftarrow Select_noeud_hasard(k)$

$k' \leftarrow Select_tournee_insertion_hasard_guide(Sol)$

$OK \leftarrow Deplacer_Noeud(x, k, k', Sol)$

jusqu'à $OK = Faux$

$S_{fin} \leftarrow Sol$

$Or_opt(S_{fin})$

k . La procédure $Select_tournee_insertion_hasard_guide(Sol)$ sélectionne une tournée pour intégrer le nœud x . Et la procédure $Deplacer_noeud(x, k, k', Sol)$ déplace le nœud x de la tournée k à la tournée k' de Sol de manière à minimiser la distance parcourue dans k' .

Description de la méthode 4 et 5

La méthode 4 et 5 se basent sur les tournées de fin de semaine S_{fin} (obtenues par l'application de l'algorithme de construction des tournées sur le problème $VRPTW_{fin}$) pour élaborer les tournées de début de semaine S_{deb} .

Algorithme 5 Procédure d'initialisation pour les méthodes 4 et 5 : $Initialisation2$

Données: $S_{fin}, VRPTW_{deb}$

Résultat: S

$S \leftarrow S_{fin}$

$Mettre_a_jour(S, VRPTW_{deb})$

Pour cela, les tournées de S_{fin} sont recopiées dans S_{deb} et les données de collecte sont mises à jour. Il reste alors à intégrer dans S_{deb} les nœuds à collecter en c_1 (ensemble de nœuds X_{c_1} de $VRPTW_{deb}$) voir l'algorithme 5. Le nombre de véhicules de S_{deb} étant potentiellement plus important que celui de S_{fin} , les méthodes 4 et 5 ajoutent progressivement des véhicules.

Pour cela, la **méthode 4** ajoute progressivement des véhicules pour réaliser la collecte des nœuds en c_1 . Une première tournée de collecte des nœuds en c_1 est établie à l'aide de l'heuristique de Solomon sans prise en compte des contraintes de capacité comme illustré sur la figure (3.5). Puis en cas de surcharge la méthode déplace des nœuds vers les tournées en sous-charge en utilisant une méthode d'insertion visant à minimiser la distance. Si cela n'aboutit pas à une solution admissible, le processus est réitéré à partir de la solution initiale (S_{fin}) en construisant deux tournées avec les nœuds en c_1 . Ce processus est réitéré tant que la méthode n'aboutit pas à une solution admissible. La méthode 4 est décrite par l'algorithme 6.

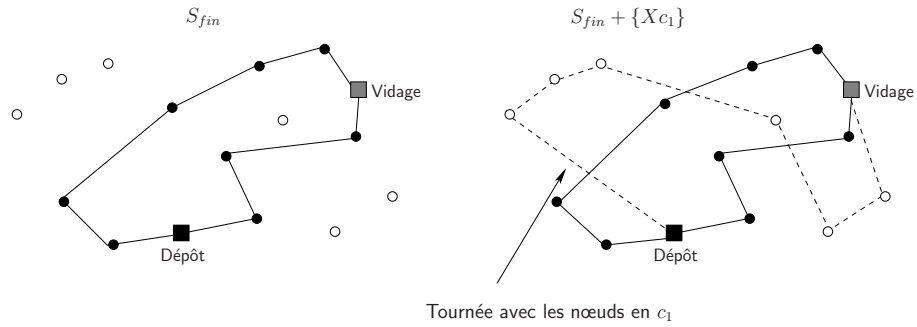


FIG. 3.5 – Construction d'une tournée avec les nœuds en c_1

Algorithme 6 Méthode 4 pour $Z_{c_1} < Z_{c_2}$

Données: $S_{fin}, VRPTW_{deb}$

Résultat: S_{deb}

$Sol \leftarrow Initialisation2(S_{fin})$

$cpt \leftarrow 1$ //nb de véhicules à ajouter

répéter

$S_{ajout} \leftarrow Solomon_sans_Capacite(Z_{c_1}, cpt)$

$Equilibrer_Tournees(S_{ajout}, Sol, OK)$

si not_OK alors

$cpt \leftarrow cpt + 1$

fin si

jusqu'à OK

$S_{deb} \leftarrow Sol$

$Or_opt(S_{deb})$

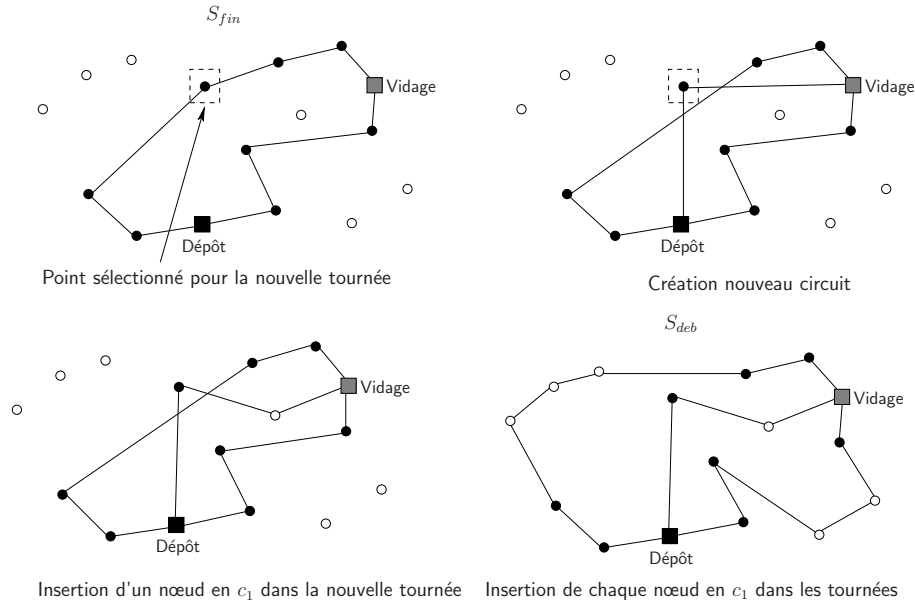


FIG. 3.6 – Méthode 5 pour $Z_{c_1} < Z_{c_2}$

La procédure *Solomon_sans_Capacite*(Z_{c_1}, cpt) construit cpt tournées distinctes avec les nœuds en c_1 sans prendre en compte les contraintes de capacité. La procédure *Equilibrer_Tournees*(S_{ajout}, Sol, OK) déplace des nœuds des tournées surchargées vers celles en sous charge. Si le déplacement ne peut s'effectuer sans respecter les contraintes, cette procédure retourne la valeur *Faux* et dans ce cas *Sol* contient la solution qu'elle avait lors de l'appel de la procédure.

La **méthode 5** cherche quant à elle des points de départ pour le(s) nouveau(x) circuit(s) parmi les nœuds à collecter en c_2 qui seraient proches des nœuds à collecter en c_1 . Une fois ces nouveaux points de départ choisis, chaque nœud en c_1 s'intègre au mieux aux différentes tournées sans prise en compte des contraintes de capacité comme illustré par la figure (3.6).

Cette méthode est itérée sur le nombre de véhicules jusqu'à trouver une solution admissible. L'algorithme 7 décrit la méthode.

Algorithme 7 Méthode 5 pour $Z_{c_1} < Z_{c_2}$

Données: $S_{fin}, VRPTW_{deb}$

Résultat: S_{deb}

$Sol \leftarrow Initialisation2(S_{fin})$

répéter

$cpt \leftarrow 1$

$Zx \leftarrow Select_noeud_depart(Sol, cpt)$

$S_{new} \leftarrow Construire_tournee(Zx)$

$Sol \leftarrow Sol \cup S_{new}$

$Integrer_noeud(Zc_1, Sol)$

jusqu'à $OK = true$

$S_{deb} \leftarrow Sol$

Or_opt(S_{deb})

3.3. Hypothèse 1 : $|Z_{c_1}| < |Z_{c_2}|$

La procédure $Select_noeud_depart(Sol, cpt)$ sélectionne cpt nœuds en c_2 proches des c_1 pour construire cpt tournées. $Construire_tournees(Zx)$ construit des tournées à partir des nœuds sélectionnés dans Zx . Et la procédure $Integrer_noeud(Zc_1, Sol)$ insère les nœuds en c_1 dans l'ensemble des tournées de Sol .

Description de la méthode 6

La **méthode 6** (voir l'algorithme 8) consiste à créer les ensembles de tournées de début et de fin de semaine indépendamment l'une de l'autre en utilisant pour chaque création l'heuristique de construction des tournées de l'algorithme 1.

Algorithme 8 Méthode 6 pour $Z_{c_1} < Z_{c_2}$

Données: $VRPTW_{deb}$ et $VRPTW_{fin}$

Résultat: S_{deb} et S_{fin}

$S_{deb} \leftarrow construction_des_tournees(VRPTW_{deb})$

$S_{fin} \leftarrow construction_des_tournees(VRPTW_{fin})$

3.3.2 Expérimentations

Les expérimentations ont été menées sur les instances à 100 clients de Solomon [1]. Ces instances sont divisées en 3 sous-groupes : C, R et RC. Comme on peut voir sur la figure (3.7), les instances C ont une distribution des clients en "cluster" (les nœuds sont regroupés géographiquement), les instances R ont une distribution aléatoire de clients et les instances RC sont un compromis entre les deux. Les instances C, R et RC sont de deux types. Les instances de type 1 ont des fenêtres temporelles plus resserrées que celles de type 2. De plus les capacités des véhicules dans les instances de type 2 sont plus grandes que celles associées des véhicules des instances de type 1. Il y a 9 instances de type C1, 12 de type R1, 8 de type RC1, 8 de type C2, 11 de type R2 et 8 de type RC2, soit 56 instances en tout.

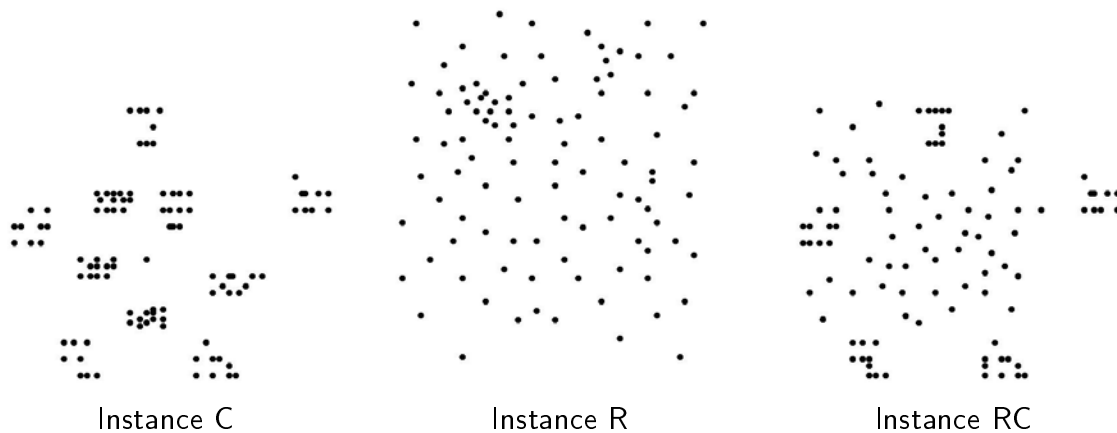


FIG. 3.7 – Instances de Solomon à 100 clients

Pour pouvoir tester nos méthodes de calcul de tournées hebdomadaires, nous avons adapté chaque instance comme expliqué ci-après.

Soit I une instance à tester, I comporte un ensemble Z de clients à collecter avec $Z = \{z_1, \dots, z_n\}$ ayant chacun une quantité q_i .

Dans un premier temps, on fixe parmi l'ensemble Z des nœuds de l'instance I ceux collectés en c_1 et ceux collectés en c_2 ainsi $Z = Z_{c_1} \cup Z_{c_2}$. Puis on duplique le fichier d'instance pour représenter les collectes de début et de fin de semaine : I_{deb} et I_{fin} .

Dans I_{deb} les quantités à collecter sur les nœuds en c_1 sont augmentées : $q_i^{deb} = q_i * \alpha \forall i \in Z_{c_1}$ avec $\alpha > 1$ et les quantités associées aux nœuds à collecter en c_2 restent inchangées : $q_i^{deb} = q_i \forall i \in Z_{c_2}$.

Dans I_{fin} les quantités à collecter sur les nœuds en c_1 sont nulles : $q_i^{fin} = 0 \forall i \in Z_{c_1}$ et les quantités associées aux nœuds à collecter en c_2 sont diminuées : $q_i^{fin} = q_i * \beta \forall i \in Z_{c_2}$ avec $\beta < 1$.

On fixe les deux coefficients de manière empirique. En moyenne, selon la collectivité, pour les nœuds qui sont collectés deux fois par semaine il y a 20% de déchets en moins en fin de semaine par rapport à la quantité de déchets de début de semaine ($\beta = 0,8$) et pour ceux qui ne sont collectés qu'une fois par semaine, la quantité de déchets est inférieure au cumul des quantités collectées deux fois par semaine d'où le coefficient ($\alpha = 1,6$).

Pour toutes les instances, on a $|Z| = 100$ et nous avons effectué 3 tests sur la proportion de nœuds en c_1 : 10%, 20% ou 30% de nœuds en c_1 .

Enfin, les nœuds appartenant à Z_{c_1} sont choisis de manière aléatoire sur les instances.

3.3.3 Résultats et conclusion

Les résultats des expérimentations sont présentés tout d'abord sur l'objectif de coût, puis sur l'objectif de stabilité. Afin de mieux appréhender la stabilité, nous la décrivons selon les différents niveaux de stabilité évoqués plus haut (de manière globale, selon le point de vue des usagers et des employés). Les tableaux détaillés des résultats de chaque méthode sur chaque objectif sont répertoriés en Annexe A.

Pour chaque type d'instance, les graphiques suivants donnent la moyenne sur les instances des coûts et des écarts en valeur absolue entre les tournées de début de semaine et celles de fin de semaine.

Résultats en termes de coût

Les graphiques (3.8) répertorient pour chaque type d'instance les résultats obtenus par les différentes méthodes pour l'objectif de coût en moyenne pour toutes les proportions de nœuds en c_1 .

Pour toutes les instances, la méthode 6 qui construit de manière indépendante les deux ensembles de tournées est celle qui donne la meilleure évaluation ou une très bonne évaluation. Sur cet objectif, la méthode 6 n'est pas la meilleure dans 4 cas (RC1 avec 10% de nœuds en c_1 , C2 avec 10 et 20% de nœuds en c_1 et R2 avec 20% de nœuds en c_1). Mais même dans ces cas là, les résultats qu'elle obtient sont très proches du meilleur résultat obtenu.

En deuxième position sur l'objectif de coût, derrière la méthode 6, on trouve la méthode 1 puis la méthode 4 pour les instances de type 1. En revanche pour les instances de type 2, c'est la méthode 2 qui arrive en deuxième position.

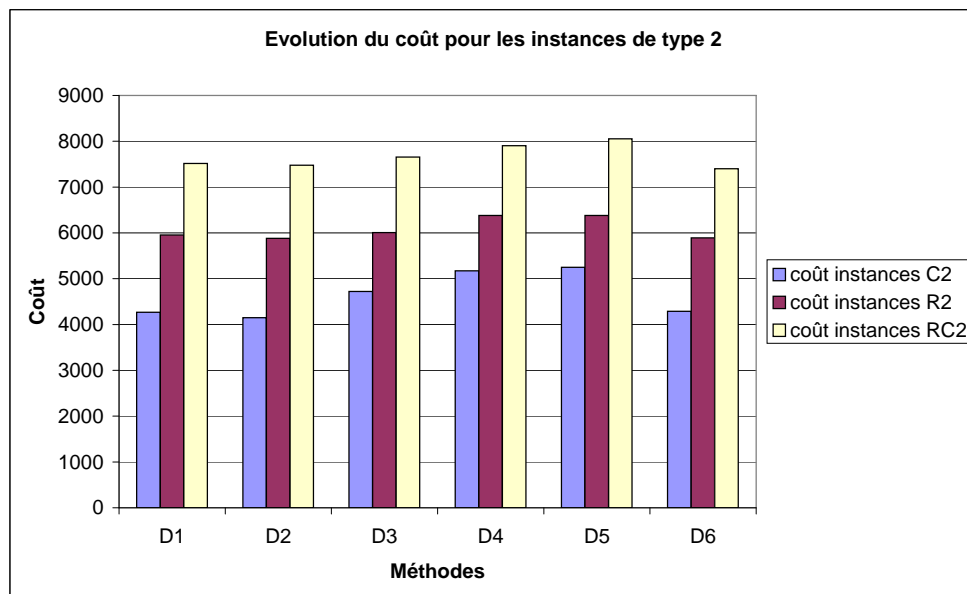
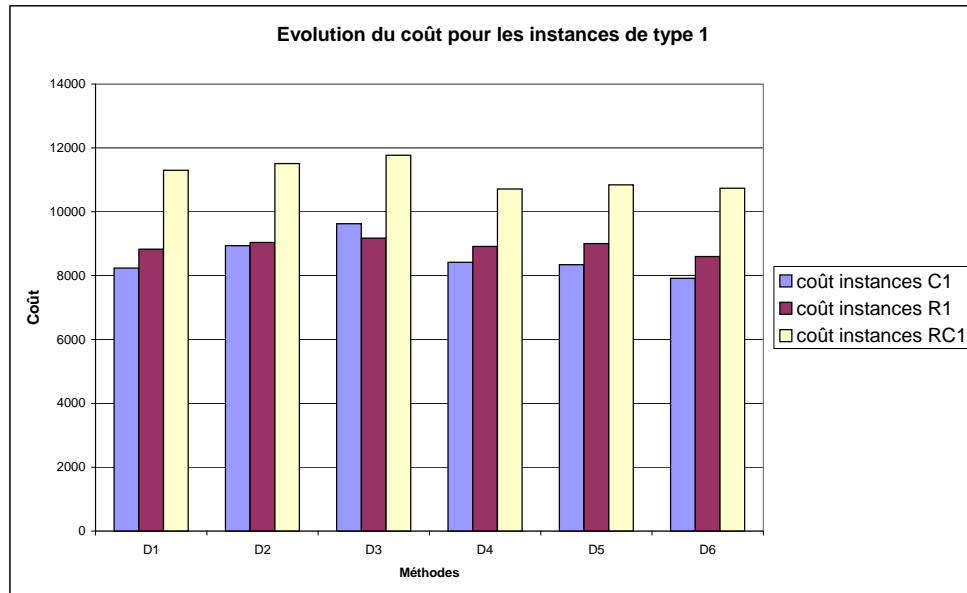


FIG. 3.8 – Comparaison des méthodes en termes de coût par type d'instances

De plus, nous pouvons observer que la méthode 5 donne en moyenne de moins bons résultats que la méthode 4, sauf sur une instance (instances C1 à 30% de nœuds en c_1). Nous pouvons dire que la méthode 4 domine la méthode 5 sur cet objectif. De même, la méthode 2 donne toujours de meilleurs ou d'équivalents résultats que la méthode 3, nous pouvons dire qu'elle la domine sur l'objectif de coût.

En résumé, sans surprise, la méthode 6 donne les meilleurs résultats en moyenne sur l'objectif de coût, puis si on doit traiter une instance avec des plages horaires plus resserrées, on trouve les méthodes 1 puis 4 et si on doit traiter des instances dont les fenêtres temporelles sont plus larges la méthode 2 donne de très bons résultats.

Résultats en termes de stabilité globale

Les graphiques (3.9, 3.10 et 3.11) présentent par type d'instance les résultats obtenus par les différentes méthodes sur l'objectif de stabilité globale, c'est-à-dire sur les écarts en termes de nombre de véhicule (E_{nbv}), de distance totale (E_{dist}) et de temps de travail total (E_{tps}).

La méthode 2 est celle qui minimise l'écart en nombre de véhicule, tandis que la méthode 5 est celle qui le maximise.

Cet écart sur le nombre de véhicule notamment sur les écarts en distance. Par exemple pour les instances de type C1, pour ce qui est de 10% de nœuds en c_1 , nous pouvons constater que les méthodes 1 et 4 donnent en moyenne des écarts en nombre de véhicules moins importants et présentent un écart de distance parmi les plus faibles mais un écart de temps des plus importants.

Pour affiner les résultats nous allons passer en revue les différents types d'instances.

- Pour les instances de type 1, la méthode 2 donne de bons résultats en terme de distance et de temps puis les méthodes 3 et 1 sont également proches des résultats de la méthode 2. La méthode 6 donne en général de mauvais résultats en termes de distance, ceci est dû au fait qu'elle utilise en moyenne plus de véhicule, et obtient de moins bons résultats en termes d'écart de temps. La méthode 1 utilise en moyenne plus de véhicules que la méthode 2 mais conserve de bons résultats sur les deux autres objectifs.
- Pour les instances de type 2, l'écart en distance est meilleur pour les méthodes 4 et 6, et l'écart en temps est meilleur pour les méthodes 2 puis 3.

Nous pouvons remarquer que la méthode donnant le plus grand écart en nombre de véhicules est celle qui donne les moins bons résultats en termes d'écarts de distance et de temps (méthode 5).

En résumé, pour les instances de type 1, les méthodes 1, 2 et 3 offrent de bons compromis. Pour ce qui est des instances de type 2, ce sont plutôt les méthodes 4 ou 6 qui restent les plus stables selon l'importance du temps ou de la distance.

Résultats en termes de stabilité du point de vue usagers

Les graphiques (3.12) présentent par type d'instance les résultats obtenus par chacune des méthodes sur l'objectif de stabilité "usagers", c'est-à-dire l'écart horaire de passage aux nœuds de collecte.

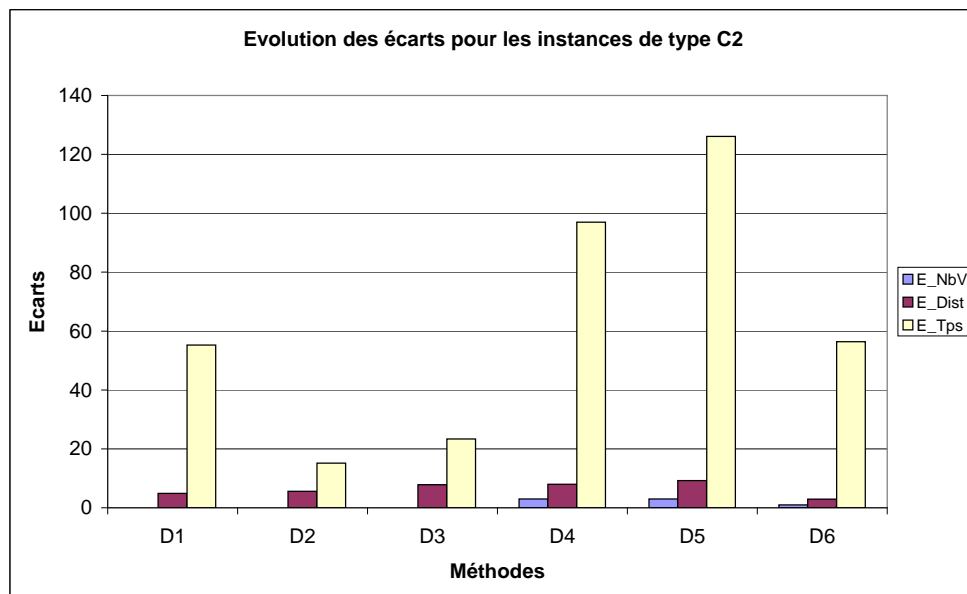
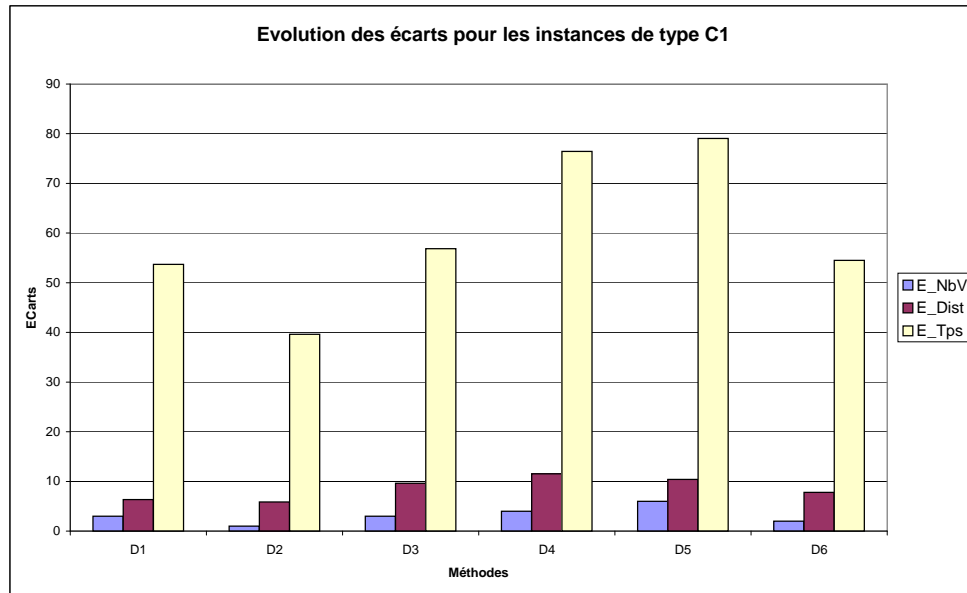


FIG. 3.9 – Comparaison des méthodes en termes de stabilité globale par type d'instances

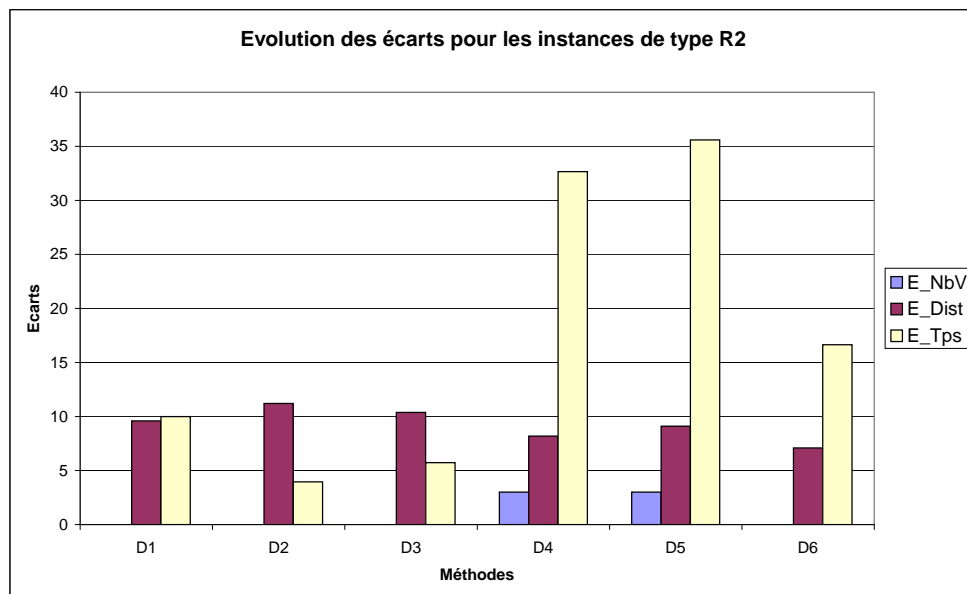
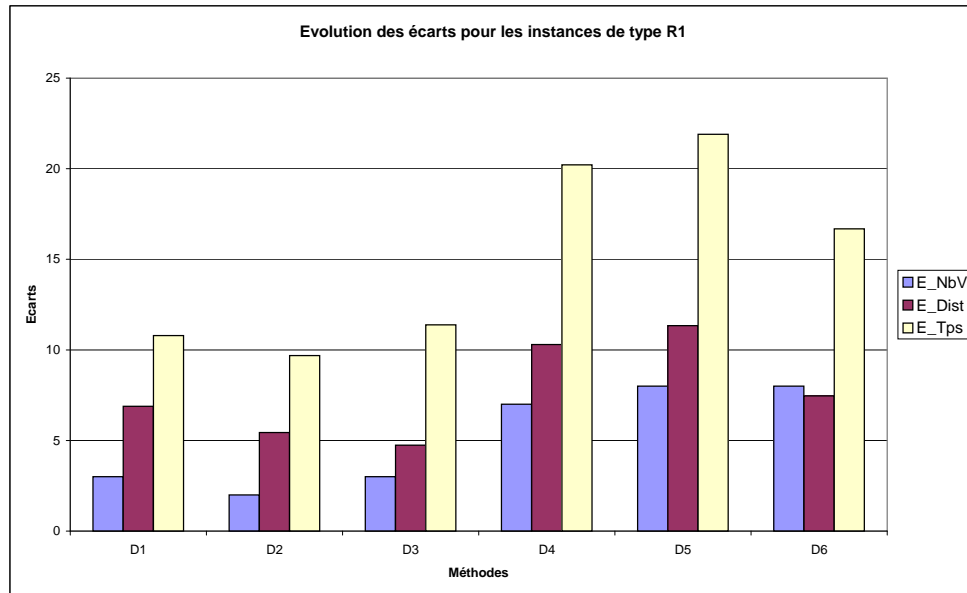


FIG. 3.10 – Comparaison des méthodes en termes de stabilité globale par type d'instances

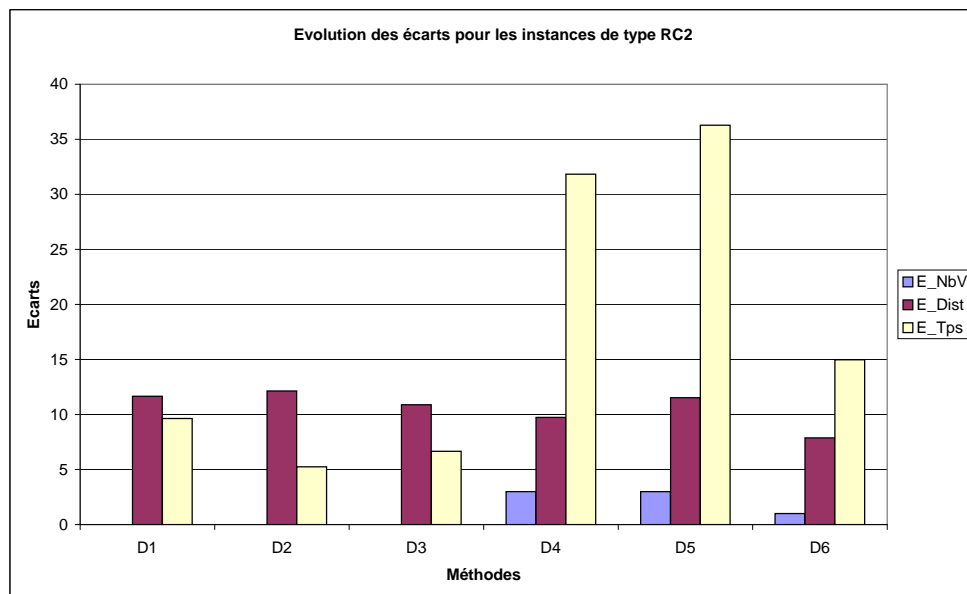
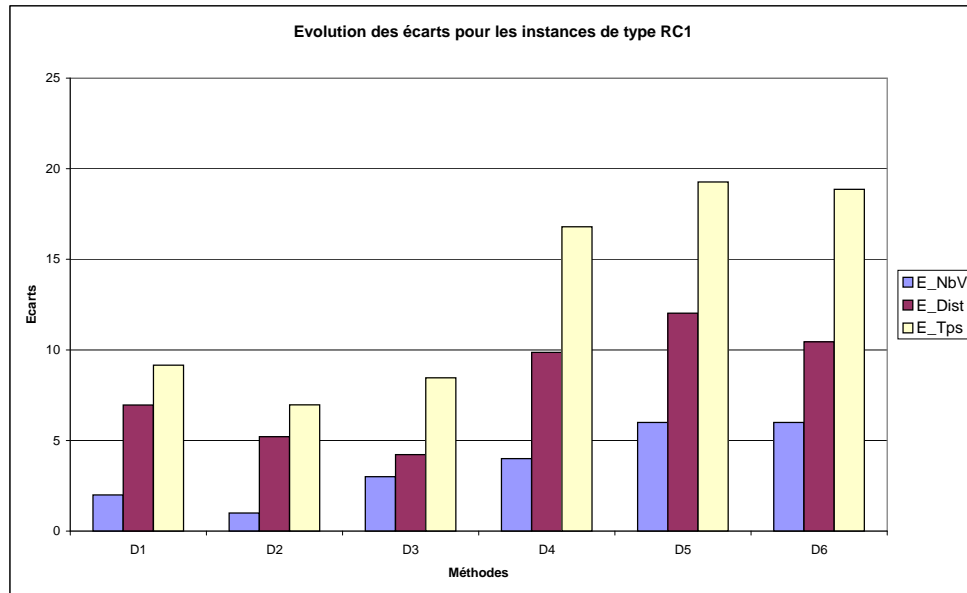


FIG. 3.11 – Comparaison des méthodes en termes de stabilité globale par type d'instances

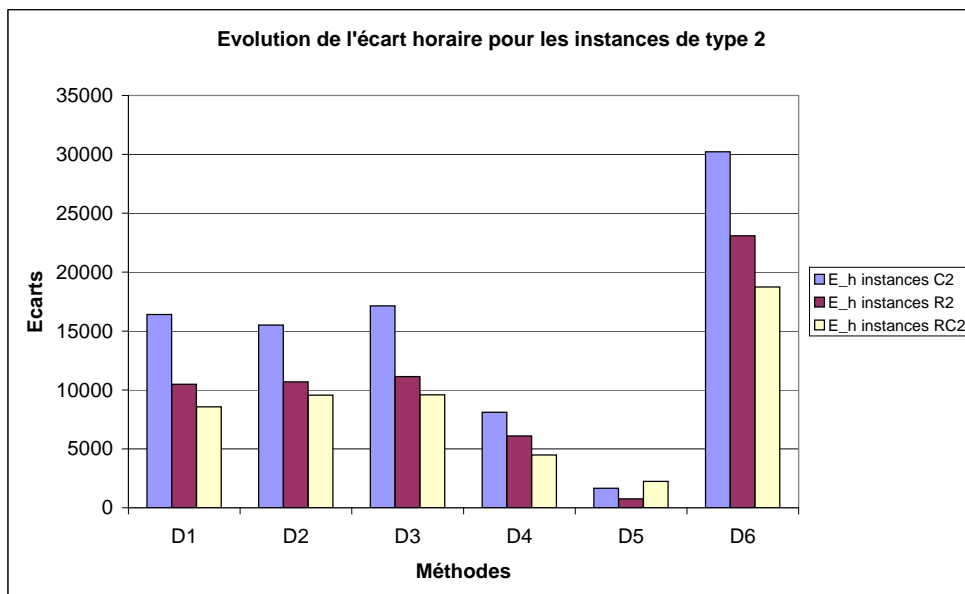
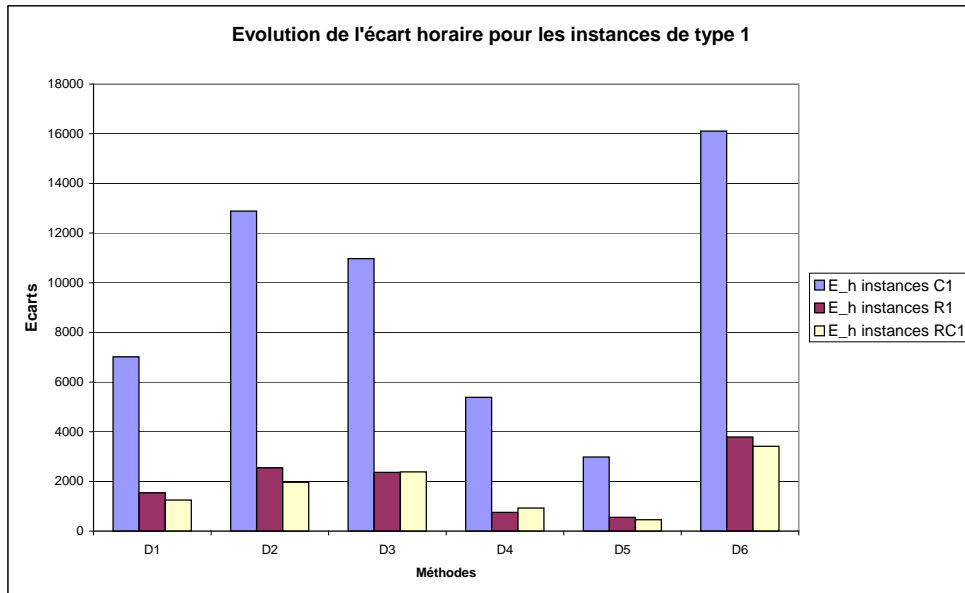


FIG. 3.12 – Comparaison des méthodes en termes de stabilité “usagers” par type d’instances

La méthode 5, donne en général les meilleurs résultats pour ce qui est de cet indicateur, elle change le moins les horaires de passage, ceci peut s'expliquer par le fait qu'elle construit en moyenne plus de tournées, et donc ne change pas du tout les autres. Elle est suivie de la méthode 4, puis de la méthode 1 pour les instances de type 1 et de la méthode 2 pour les instances de type 2. La méthode 6 donne les moins bons résultats en termes d'écart horaire, c'est elle qui bouleverse le plus les horaires de passage sur les nœuds.

En résumé, pour cet indicateur, la méthode 5 donne les meilleurs résultats, puis la méthode 4.

Résultats en termes de stabilité du point de vue employés

Les graphiques (3.13, 3.14 et 3.15) répertorient par type d'instance les résultats obtenus par chacune des méthodes sur l'objectif de stabilité "employés".

- Pour les instances de type 1, les méthodes 1, 4 et 5 donnent en général les meilleurs résultats en termes de stabilité "employés" et notamment pour ce qui de l'écart sur la composition des circuits et sur l'ordre de passage dans les nœuds pour les instances dans lesquels le nombre de nœuds à collecter en c_1 est peu important (10%) et dont les fenêtres de temps sont resserrées.
- Pour les instances de type 2, la méthode 5 donne en général les meilleurs résultats suivie par les méthodes 2 et 3, sauf sur l'indicateur d'écart de capacité qui peut être parfois assez important. La méthode 1 est souvent un compromis sur les différents indicateurs décrivant l'objectif de stabilité "employés".

En résumé, nous pouvons mettre en avant la méthode 5 pour les instances de type 1 et de type 2, suivie des méthodes 4 et 1 pour les instances de type 1 et des méthodes 2 et 3 pour les instances de type 2. La méthode 1 qui est un compromis toutes instances confondues.

La méthode 6 donne en moyenne les moins bons résultats en termes de stabilité "employés". En effet, c'est celle qui a les plus grands écarts sur la composition des circuits et sur l'ordre de passage dans les nœuds. Elle donne également les plus forts écarts en termes de distance et de temps. Seul l'indicateur de capacité est parfois bon.

En conclusion les résultats obtenus sur ces méthodes heuristiques nous permettent d'en faire ressortir leurs caractéristiques principales. En effet, les méthodes 1 et 4 semblent adaptées pour les instances de type 1, et proposent un compromis entre le coût et la stabilité. La méthode 2 semble être plus adaptée pour les instances de type 2 et permet de conserver une bonne stabilité. La méthode 3 est souvent dominée par rapport à la méthode 2 et ne présente donc pas d'intérêt particulier. La méthode 5 est mauvaise pour ce qui est de l'objectif de coût mais très bonne pour ce qui est de l'objectif de stabilité usagers et employés. En moyenne elle construit plus de tournées que les autres méthodes. Et enfin, la méthode 6, qui nous sert de comparatif pour les autres méthodes en terme de coût, donne de très bons résultats pour ce qui est du coût mais bien entendu ne conserve pas des tournées "similaires".

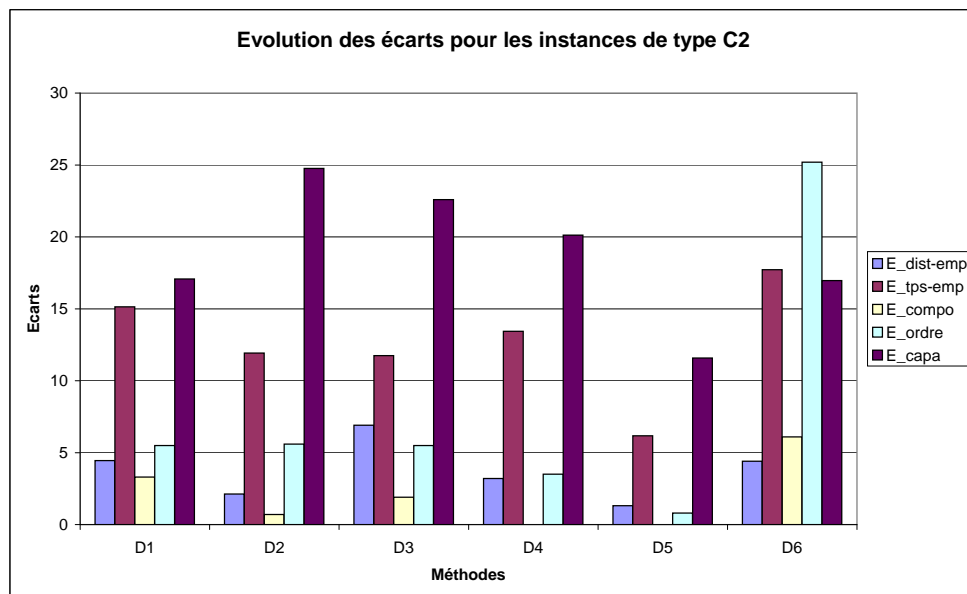
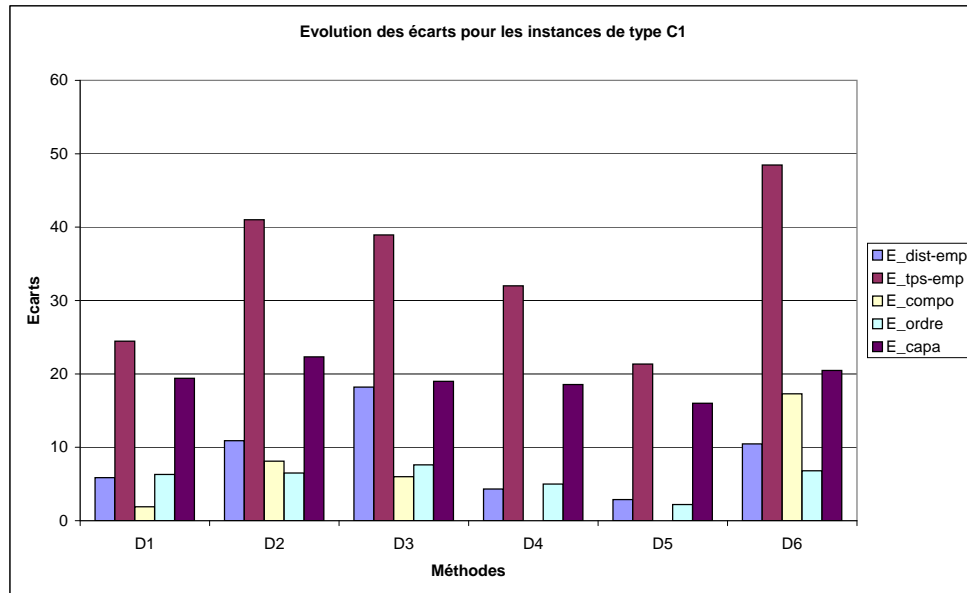


FIG. 3.13 – Comparaison des méthodes en termes de stabilité “employés” par type d’instances

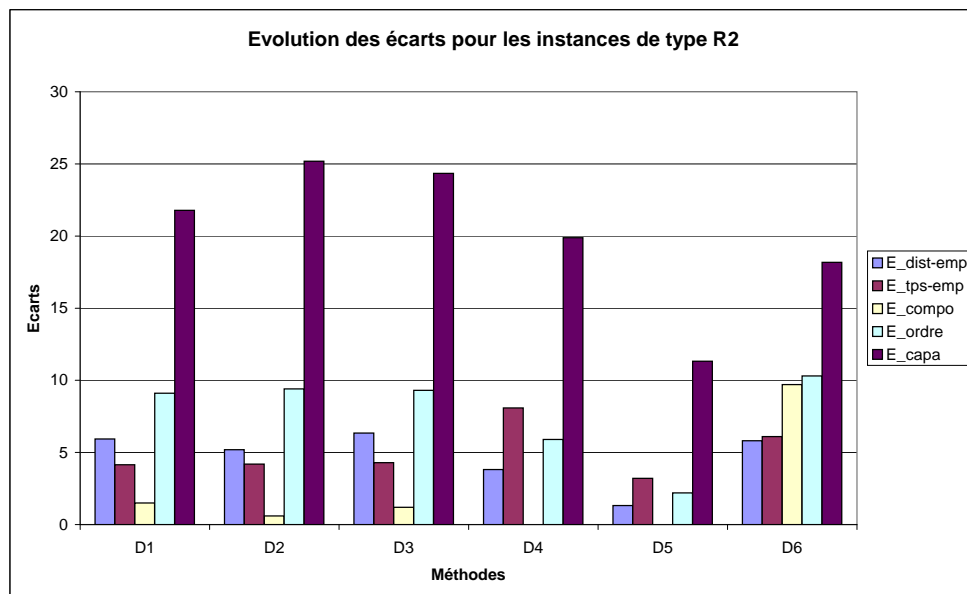
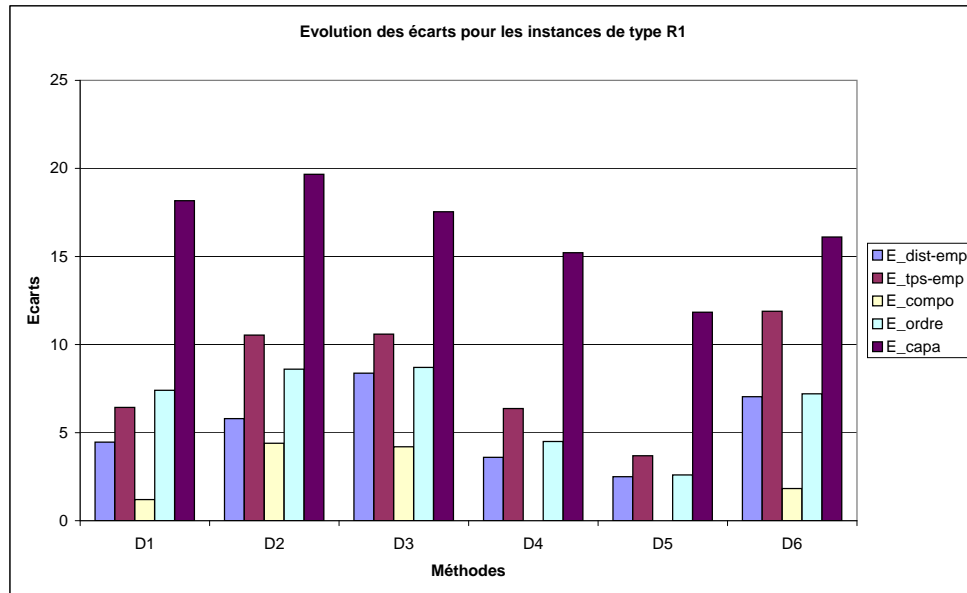


FIG. 3.14 – Comparaison des méthodes en termes de stabilité “employés” par type d’instances

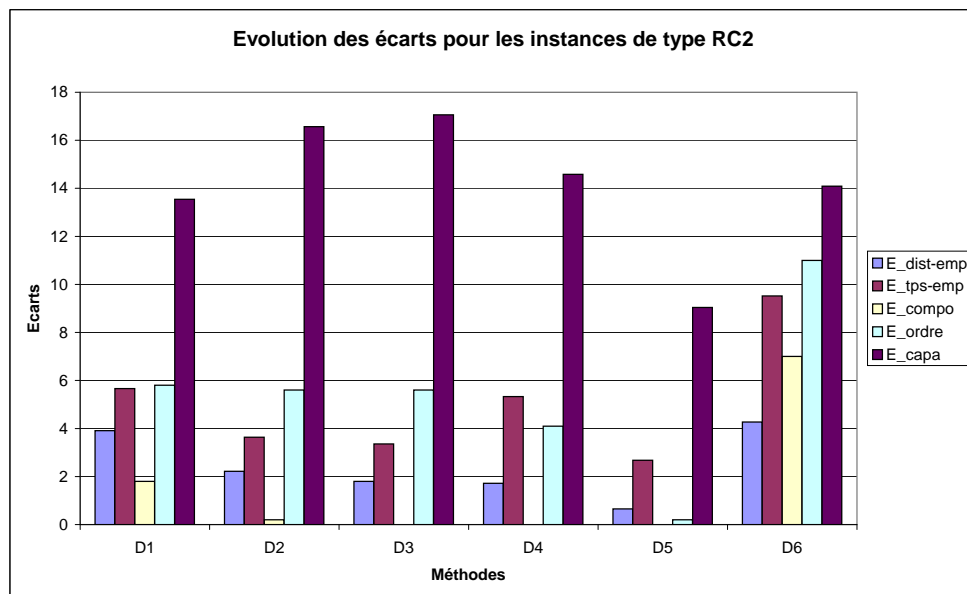
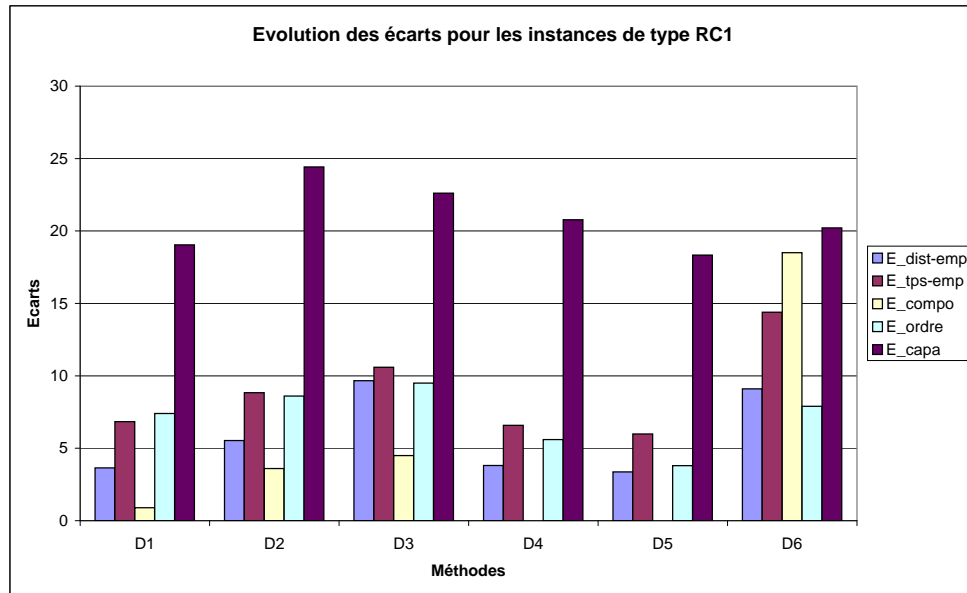


FIG. 3.15 – Comparaison des méthodes en termes de stabilité “employée” par type d’instances

3.3.4 Métaheuristiques multi-objectifs

Cette partie s'intéresse à l'exploitation de méthodes métaheuristiques multi-objectifs pour la résolution du problème de collecte hebdomadaire.

Deux métaheuristiques multi-objectifs ont été utilisées et leurs résultats comparés avec les méthodes heuristiques précédentes. Ces méthodes font partie des métaheuristiques utilisant la notion de voisinage. Nous nous limitons dans cette étude à des méthodes à voisinage. Les deux méthodes testées sont :

- une recherche tabou,
- un recuit simulé.

Dans la partie suivante, nous allons présenter ces métaheuristiques et les résultats obtenus sur les différentes instances.

Description des méthodes

Pour les méthodes suivantes basées sur les notions de voisinage, nous avons défini deux voisinages classiques en tournées de véhicules :

- déplacement d'un ou plusieurs nœuds dans un circuit (choix du nœud dû au hasard),
- échange de 0 à 1 nœud entre deux tournées de la solution courante.

Méthode basée sur une recherche tabou

La méthode tabou proposée a été adaptée pour les problèmes multi-objectifs tel que décrit dans l'algorithme 9.

Algorithme 9 Méthode de recherche tabou appliquée au multi-objectifs

Données: $S_{deb}, VRPTW_{fin}$

Résultat: S_{fin}

$S \leftarrow Initialisation1(S_{deb}, VRPTW_{fin})$

tant que critère d'arrêt non atteint **faire**

$V \leftarrow Select_{voisinage}(V1, V2)$

 Selectionner meilleure S' admissible et non tabou dans V (avec limite sur le nombre de voisin)

si ($S'Non_Dominee$) et ($S'Non_inclue_dans_Front_Pareto$) **alors**

 Intégrer S' au Front Pareto

 Ajouter S' dans la liste Tabou

fin si

$S \leftarrow$ choix au hasard d'une solution dans le Front Pareto Non Tabou

fin tant que

Le critère d'arrêt est fixé selon un nombre d'itérations. Il doit être suffisant pour permettre une recherche assez approfondie mais pas trop long pour que la méthode reste dans un temps raisonnable.

Une particularité de cette méthode est de ne pas explorer l'ensemble du voisinage, pour limiter le temps d'exécution de la méthode, mais de définir un certains nombre de voisin envisagé. Le voisinage est choisi au hasard parmi les deux voisinages possibles présentés ci-dessus. Cela implique que la recherche

tabou traditionnelle a été légèrement modifiée pour laisser un peu de marge à un aspect probabiliste ceci nous permet juste de diversifier un peu l'exploration du voisinage.

Le Front Pareto : le Front Pareto obtenu correspond aux différentes solutions non-dominées trouvées dans un des voisinages utilisés au hasard comme expliqué ci-dessus.

La solution de départ est celle obtenue à l'aide de l'algorithme de construction des tournées (l'algorithme 1). Pour ce qui est de la solution de travail courante, elle est choisie au hasard sur le Front Pareto, du moment qu'elle ne soit pas dans la liste Tabou. Cela permet de diversifier la recherche sur différentes solutions de départ.

La taille de la liste tabou : ce paramètre est le paramètre fondamental de la recherche avec tabou, il consiste à interdire un certain nombre de retour sur des solutions déjà envisagées.

Méthode multi-objectif basée sur un recuit simulé

Nous avons adapté une méthode de recuit simulé aux problèmes multi-objectifs considéré (l'algorithme 10).

Cette méthode utilise différents paramètres, comme pour la recherche avec tabou, la difficulté réside dans le réglage de ces paramètres. Pour cela, nous allons développer le paramétrage suivant :

- La température initiale T_0 : elle doit être assez importante afin de pouvoir visiter un grand nombre de transformations. Dans cet algorithme, nous avons choisi de générer un certain nombre de transformations et de calculer leur coût Δ . Un taux d'acceptation, t_a est choisi au départ et la valeur de la température de départ T_0 est calculée de la manière suivante : $T_0 = \Delta / \ln(1/t_a)$.
- La décroissance de température (fonction g) : la loi de décroissance fait baisser la température de manière continue. Cette loi est appelée g dans l'algorithme avec $t_k = g(t_{k-1})$, où t_k et t_{k-1} correspondent respectivement à la température au moment k et au moment $k-1$. Nous choisissons donc une loi de décroissance de telle sorte que $t_k = t_{k-1} * pas$, avec $0 < pas < 1$.
- niveau : ce paramètre correspond au moment où la température va changer, autrement dit un nombre d'itérations données pour une même température.
- Critère d'arrêt : dans notre algorithme nous fixons un nombre d'itérations maximales sur la valeur de la température.

Pour utiliser les méta-heuristiques sous chacun des objectifs, nous avons fixé les différents paramètres pour chaque méthode. Plusieurs tests ont été effectués pour établir ses paramètres. Ils ont donc été fixés de manière empirique.

Algorithme 10 Méthode de recuit simulé appliquée au multi-objectifs

Données: $S_{deb}, VRPTW_{fin}$ **Résultat:** S_{fin} $S \leftarrow \text{Initialisation1}(S_{deb}, VRPTW_{fin})$ $t_{10} \leftarrow$ température de départ pour le coût $t_{20} \leftarrow$ température de départ pour la stabilité $f_1(S) \leftarrow$ évaluation du coût de la solution de départ $f_2(S) \leftarrow$ évaluation de la stabilité de la solution de départ $k = \text{Nb incrmnts} //$ indice**pour** Nb incréments **faire** Choisir aléatoirement une solution S' dans le voisinage $V(S)$ de S $\Delta f_1 : \Delta f_1 = f_1(S') - f_1(S)$ $\Delta f_2 : \Delta f_2 = f_2(S') - f_2(S)$ **si** $(\Delta f_1 \leq 0)$ et $(\Delta f_2 \leq 0)$ **alors** $S = S'$ et $f_1(S) = f_1(S')$ et $f_2(S) = f_2(S')$ **sinon si** $(\Delta f_1 \leq 0)$ et $(\Delta f_2 > 0)$ **alors** $S = S'$ avec une probabilité $e^{-\Delta f_2/t_{20}}$ **sinon si** $(\Delta f_1 > 0)$ et $(\Delta f_2 \leq 0)$ **alors** $S = S'$ avec une probabilité $e^{-\Delta f_1/t_{10}}$ **sinon** $S = S'$ avec une probabilité $= \min(e^{-\Delta f_1/t_{10}}, e^{-\Delta f_2/t_{20}})$ **fin si** **si** $(S' \text{Non_Dominee})$ et $(S' \text{Non_inclue_dans_Front_Pareto})$ **alors** Intégrer S' au Front Pareto **fin si** $t_{10} = g_1(k - 1)$ $t_{20} = g_2(k - 1)$ $k = k - 1$ **fin pour**

Expérimentations

Pour tester ces méthodes nous avons gardé les mêmes instances que précédemment (instances de Solomon C, R et RC) avec les mêmes modifications pour obtenir un problème de tournées hebdomadaires.

- La difficulté pour les expérimentations est de faire ressortir les deux objectifs définis précédemment :
- un objectif de coût (distance totale parcourue par les tournées de début et de fin de semaine),
 - un objectif sur la stabilité : cet objectif est plus difficile à fixer puisqu'il met en jeu différents sous-objectifs. Nous proposons de faire une agrégation de différents sous-objectifs de cet objectif total de la manière suivante :

$$\text{stabilité} = E_{\text{dist-emp}} + E_{\text{tps-emp}} + E_h * 0,05 + E_{\text{compo}} + E_{\text{ordre}} + E_{\text{capa}}$$

L'accent a été mis sur la stabilité des usagers et des employés, pour faire ressortir ces deux aspects primordiaux de la stabilité, ce sont eux qui vont être affectés par les changements. Cette agrégation est issue de différents tests, nous avons mis un coefficient inférieur pour les écarts horaire pour ne pas appuyer de façon exagérée sur cet indicateur car il est en général très supérieur aux autres.

Résultats et comparaison

Les résultats de cette partie ne peuvent être présentés qu'instance par instance, puisque le Front de Pareto ne peut être agrégé dans une moyenne. Afin de comparer les méthodes heuristiques et multi-objectifs, nous ne pouvons pas présenter l'ensemble des résultats sur les 56 instances. Nous avons choisi de présenter un seul résultat par type d'instance utilisées (C, R et RC de type 1 et 2).

La figure (3.16) présente les solutions explorées en utilisant les méthodes métaheuristiques multi-objectifs décrites ci-dessus sur l'instance C101.

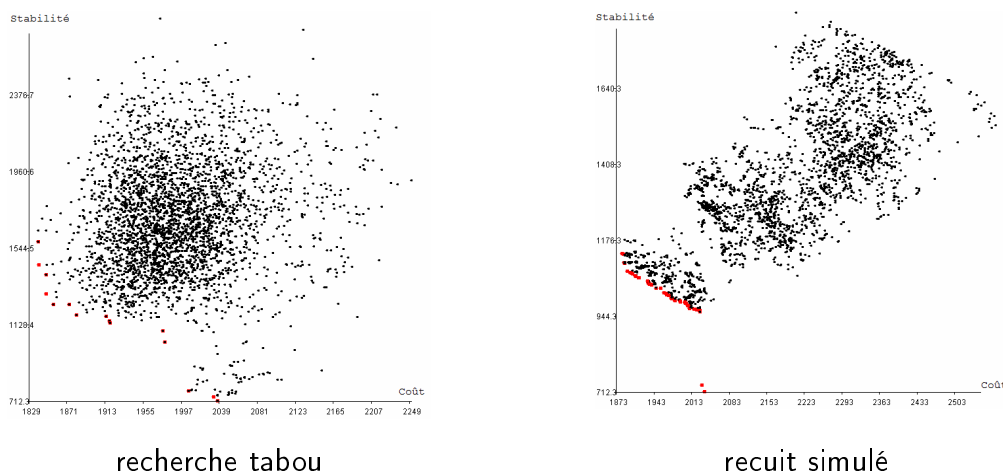


FIG. 3.16 – Fronts Pareto obtenus par une recherche tabou et un recuit simulé sur l'instance C101

Nous avons représenté dans les figures (3.17, 3.18 et 3.19) les fronts Pareto obtenus par chacune

des méthodes sur une instance de chaque type. Sur ces figures, nous avons également positionné les résultats obtenus par nos différentes méthodes heuristiques.

Nous pouvons dans un premier temps observer globalement que le nombre de solutions proposées par les métaheuristiques est plus important que celui proposé par les heuristiques. Ces solutions sont similaires en termes de distance et de stabilité. Pour ce qui est des méthodes heuristiques, les solutions sont plus dispersées et donnent des résultats très différents les uns des autres. Les méthodes heuristiques donnent moins de solutions exploitables, par exemple pour les instances C101 et RC201, une solution domine l'ensemble des autres.

Nous allons voir par la suite chaque instance plus en détails.

Pour l'instance **C101**, les méthodes heuristiques donnent une solution (méthode 1) de bonne qualité pour ce qui est des deux objectifs. Elle fait partie des solutions trouvées par les métaheuristiques sinon les autres sont dominées. Les métaheuristiques, elles proposent plusieurs compromis coût-stabilité qui peuvent être intéressants. Notamment elle domine les heuristiques en termes de coût.

Pour l'instance **R101**, les heuristiques sont bien réparties par rapport aux métaheuristiques et proposent même les meilleures solutions extrêmes (celle qui donne une meilleure solution en terme de distance et une meilleure solution en terme de stabilité). Elle propose également un compromis trouvé par les métaheuristiques.

Pour ce qui est de l'instance **RC101**, une méthode heuristique propose une très bonne solution qui domine l'ensemble des autres en termes de coût et de stabilité, de même pour ce qui est de l'instance **RC201**.

Pour les deux dernières instances, les métaheuristiques proposent de très bonnes solutions en termes de coût tandis que les méthodes heuristiques proposent une bonne solution en terme de stabilité. De plus chacune d'entre elles présente une solution heuristique faisant partie des Fronts Pareto (solution minimisant le coût, en début de Front).

Conclusion

La force des métaheuristiques est de proposer plusieurs solutions non-dominées, qui donnent donc un choix plus important au décideur. Selon les expérimentations réalisées, nous pouvons constater que ces méthodes obtiennent des solutions ayant un meilleur coût. En revanche, les méthodes heuristiques peuvent obtenir des solutions avec une meilleure stabilité. Le point négatif est que le temps d'exécution des ces méthodes est très supérieur à celui des méthodes heuristiques.

Les méthodes heuristiques trouvent une unique solution pour chaque méthode (pas de compromis à proposer a priori) et permettent, si on choisit la bonne heuristique adaptée au problème, de trouver une solution de très bonne qualité en termes de coût et de stabilité. Elles permettent en général de trouver de très bonnes solutions sur l'objectif de stabilité et leur temps d'exécution est bien inférieur à celui d'exécution des métaheuristiques.

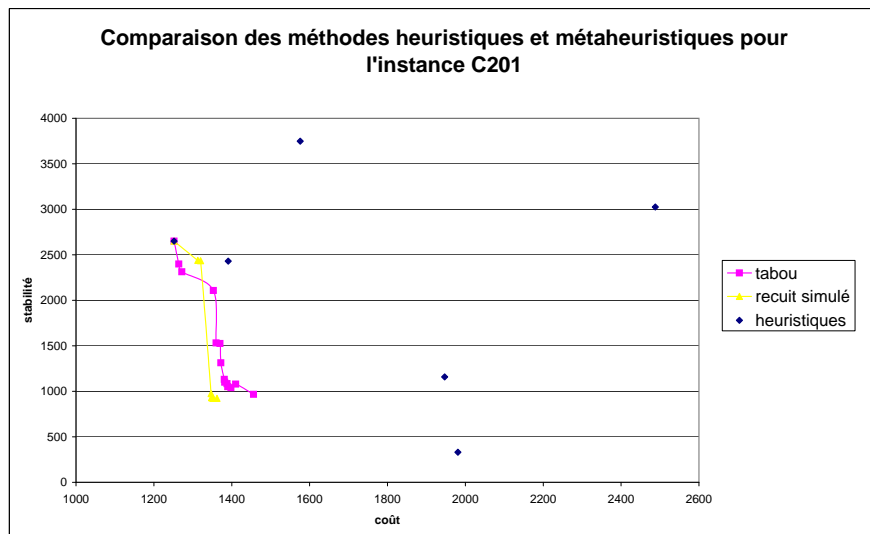
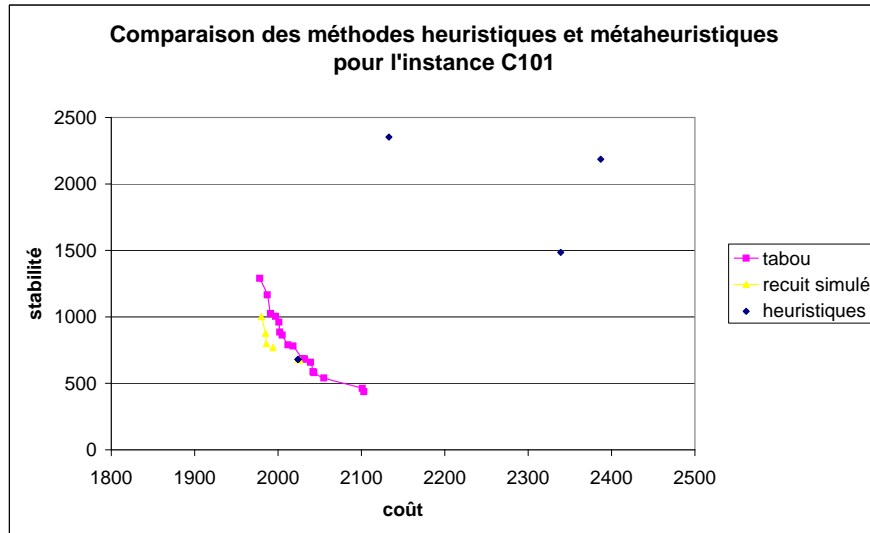


FIG. 3.17 – Résultats comparatifs heuristiques-métaheuristiques

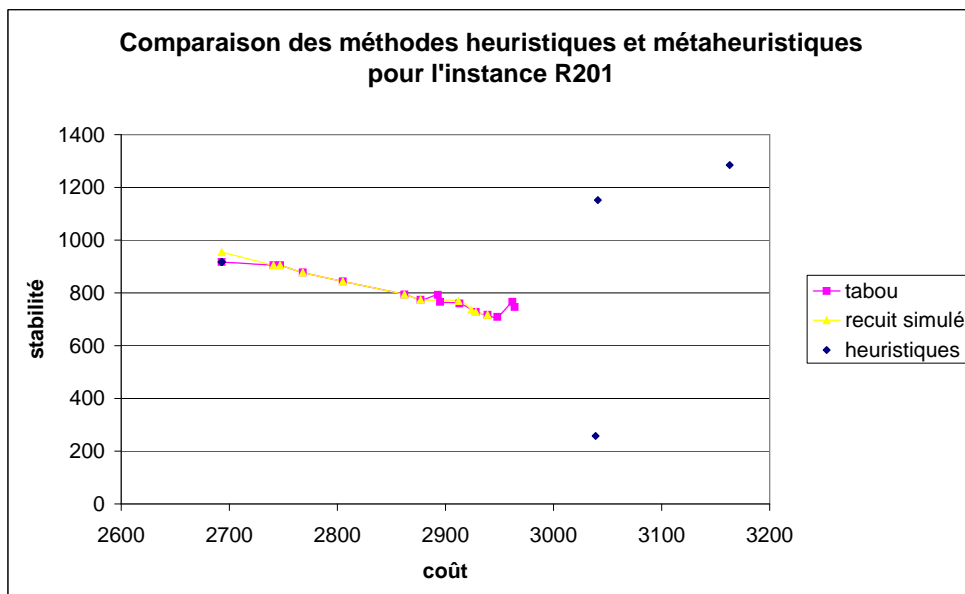
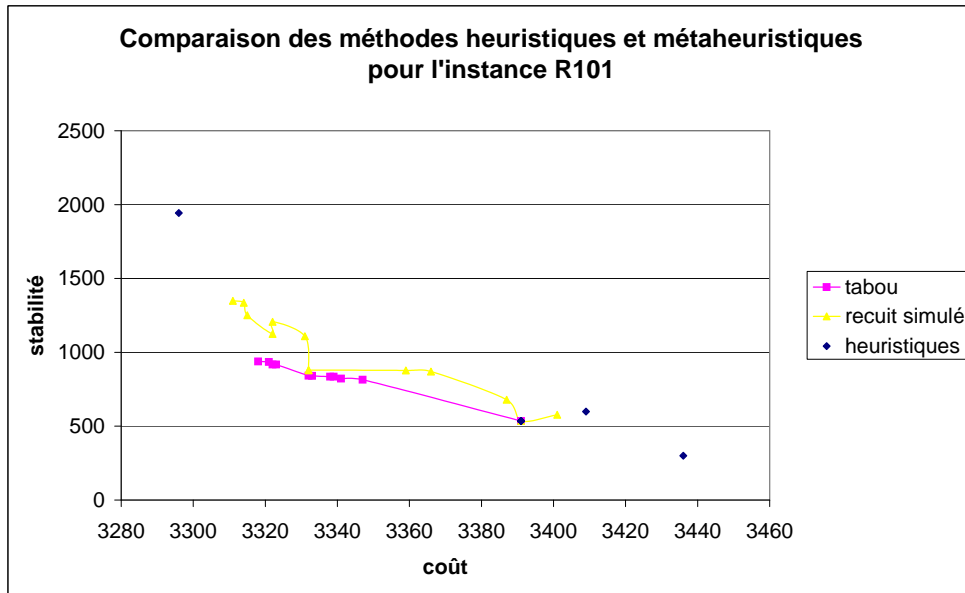


FIG. 3.18 – Résultats comparatifs heuristiques-métaheuristiques

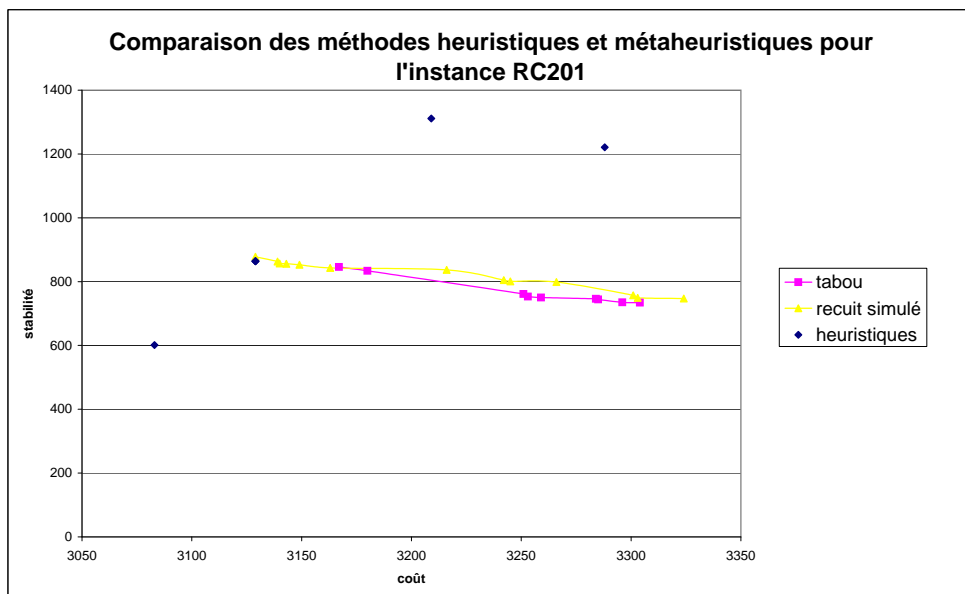
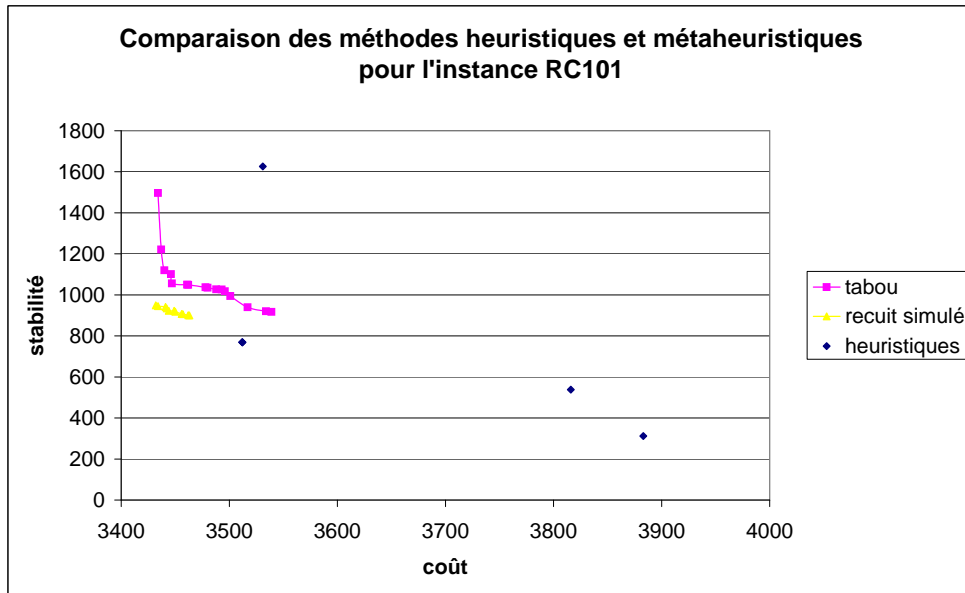


FIG. 3.19 – Résultats comparatifs heuristiques-métaheuristiques

3.4 Hypothèse 2 : $|Z_{c_1}| > |Z_{c_2}|$

Dans cette seconde partie de l'étude du problème de collecte hebdomadaire, le nombre de nœuds à collecter en c_1 est très supérieur au nombre de nœuds à collecter en c_2 . Pour cela des méthodes spécifiques et différentes de celles énoncées dans le cadre de la première hypothèse sont proposées.

En effet, avec cette nouvelle hypothèse tous les nœuds en c_1 ne peuvent plus être collectés uniquement en début de semaine, il faut donc les répartir sur la période complète de collecte, c'est-à-dire sur l'ensemble de la semaine comme décrit dans la figure (3.20). Nous sommes ici en présence d'un problème de *PVRPTW*. Les méthodes de création de tournées proposées précédemment ne sont plus adaptées, il est nécessaire d'en proposer de nouvelles. Malgré la différence notable pour ce qui est du modèle utilisé, la structure du problème est la même que celle de l'hypothèse 1, il s'agit ici d'organiser deux ensembles de tournées. Les méthodes heuristiques proposées pour résoudre ce problème utilisent donc des mécanismes similaires à ceux utilisés dans l'hypothèse 1.

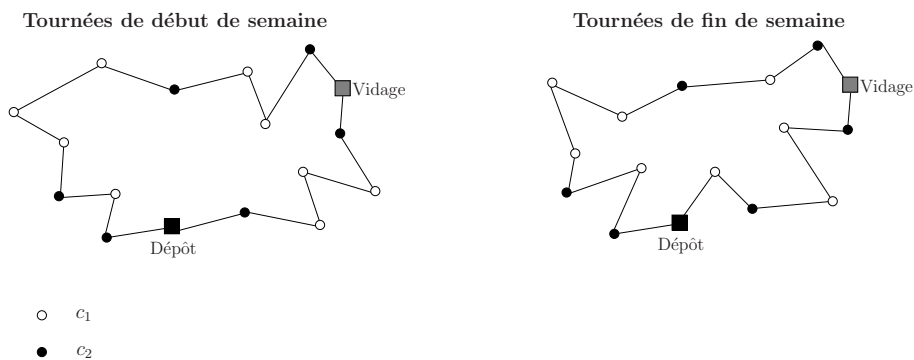


FIG. 3.20 – schéma représentant la collecte si $|Z_{c_1}| > |Z_{c_2}|$

3.4.1 Principe de résolution

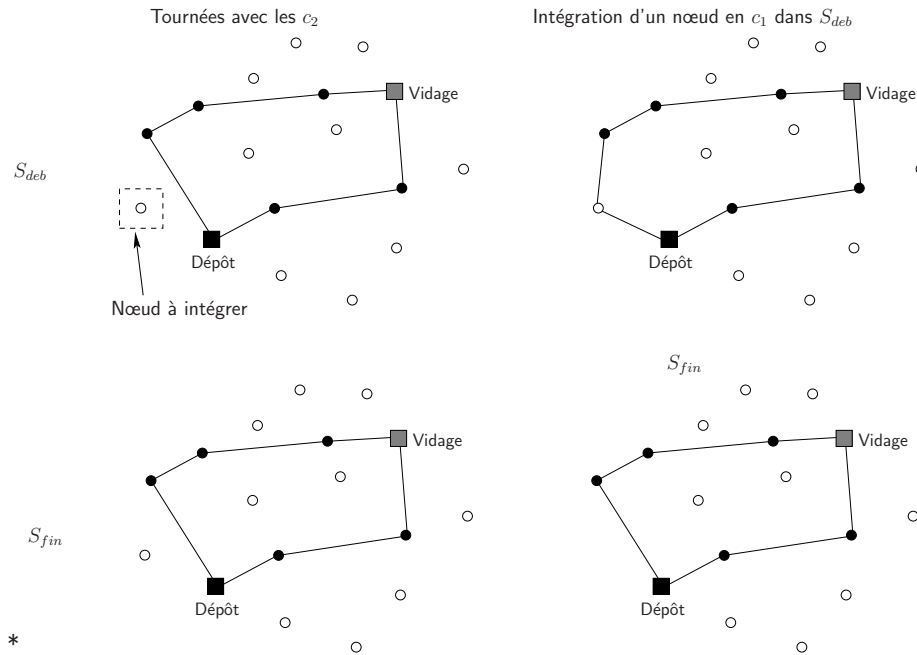
Les deux méthodes proposées visent à minimiser les écarts de stabilité entre les tournées du début et de fin semaine pour trouver des solutions les plus proches possibles. Deux manières de construire les tournées sont possibles : soit on construit les tournées à l'aide des nœuds en c_1 puis on leur rajoute les nœuds à collecter en c_2 ; soit on construit les tournées à l'aide des nœuds en c_2 puis on rajoute les nœuds à collecter en c_1 .

L'objectif de stabilité concerne principalement les nœuds à collecter en c_2 puisque ce sont ces nœuds qui sont susceptibles de changer de place d'un ensemble de tournées à l'autre. Pour limiter les écarts entre les collectes de début et de fin de semaine, nous avons choisi de constituer les tournées S_{deb} et S_{fin} avec les nœuds à collecter en c_2 . Les nœuds en c_1 sont ensuite insérés dans les tournées ainsi constituées.

Pour construire les tournées de début et de fin de semaine composées uniquement des nœuds en c_2 , nous allons utiliser l'algorithme 1 vu précédemment. Deux méthodes de résolution du problème de collecte hebdomadaire sont proposées dans la suite.

Description de la première méthode

Pour chaque nœud en c_1 restant à collecter, la méthode estime sa meilleure insertion dans les tournées construites (début ou fin de semaine), puis elle choisit le nœud qui s'insère au mieux comme décrit sur la figure (3.21).

FIG. 3.21 – Méthode 1 pour $Z_{c_1} > Z_{c_2}$

S'il n'est plus possible d'insérer des nœuds dans les tournées en place mais qu'il reste des nœuds à collecter en c_1 , la méthode construit directement des tournées avec ces nœuds restants à l'aide de l'algorithme 1. Ces tournées sont placées en début de semaine pour se calquer sur le fonctionnement actuel. La méthode est décrite par l'algorithme 11.

Algorithme 11 Méthode 1 pour $Z_{c_1} > Z_{c_2}$ **Données:** $PVRPTW$ **Résultat:** S_{deb}, S_{fin} $Z'_{c_1} \leftarrow Z_{c_1}$ $S_{deb} \leftarrow \text{construction_des_tournees}(VRPTW_{Z_{c_2}})$ $S_{fin} \leftarrow S_{deb}$ **répéter** $OK \leftarrow \text{Insérer_noeud_c1}(Z'_{c_1}, S_{deb}, S_{fin})$ **jusqu'à** $OK = \text{Faux}$ **si** $\text{Not Est_vide}(Z'_{c_1})$ **alors** $S' \leftarrow \text{construction_des_tournees}(Z'_{c_1})$ **fin si** $S_{deb} \leftarrow S_{deb} \cup S'$ $\text{Or_opt}(S_{deb})$ $\text{Or_opt}(S_{fin})$

3.4. Hypothèse 2 : $|Z_{c_1}| > |Z_{c_2}|$

La procédure $Inserer_noeud_c_1(Z'_{c_1}, S_{deb}, S_{fin})$ va chercher à insérer au mieux les nœuds en c_1 restant à être intégrés à une tournée appartenant soit à S_{deb} soit à S_{fin} . En cas d'égalité, la méthode privilégie l'insertion dans des tournées de S_{deb} . Cette procédure renvoie *Faux* si le nœud ne peut pas être inséré. La procédure $Ajouter(S', S_{deb})$ ajoute les nouvelles tournées S' aux tournées de S_{deb} .

Description de la méthode 2

La deuxième méthode se base sur le même principe de compléter les circuits déjà en place sauf qu'elle cherche à chaque itération à compléter un circuit avec les nœuds en c_1 non encore insérés. Pour chaque tournée de S_{deb} et S_{fin} on cherche pour chaque nœud en c_1 la meilleure insertion admissible dans la tournée à augmenter. Un exemple est donné sur la figure (3.22).

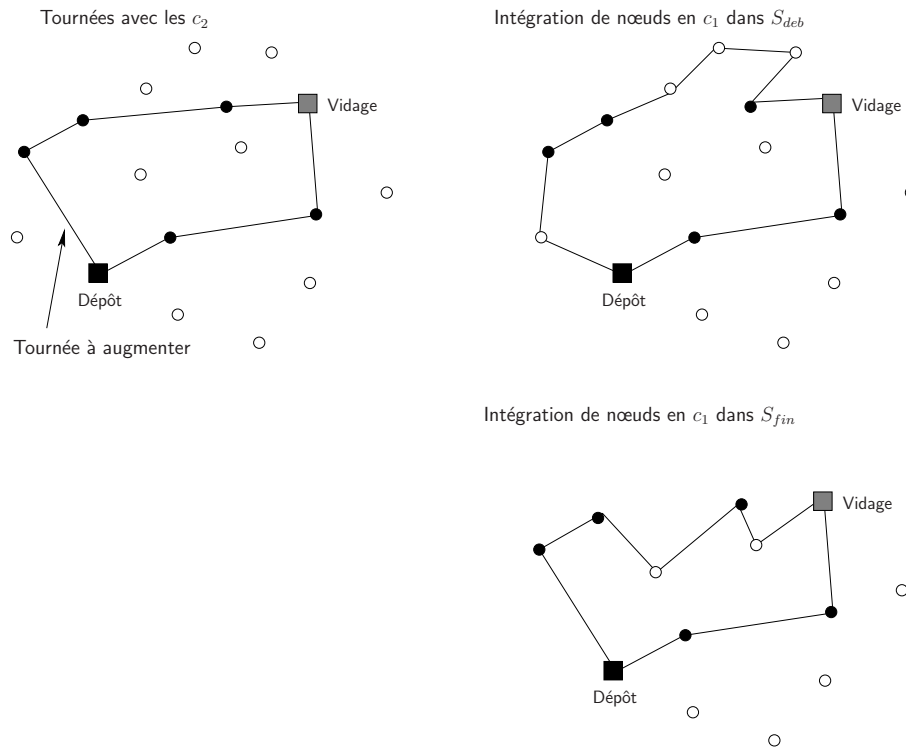


FIG. 3.22 – Méthode 2 pour $Z_{c_1} > Z_{c_2}$

De la même façon que pour la méthode 1, s'il reste des nœuds à collecter en c_1 , on construit des tournées avec ces derniers nœuds. La méthode est décrite par l'algorithme 12.

La procédure $Select_tournee_a_Completer(S_{deb}, S_{fin})$ sélectionne une tournée à compléter. Elle commence par les tournées de S_{deb} puis celles de S_{fin} dans l'ordre des circuits.

La procédure $Inserer_noeud_c_1_tournee(Z'_{c_1}, k)$ insère dans la tournée k un nœud en c_1 non encore intégré. Ce nœud est enlevé de l'ensemble Z'_{c_1} . Elle renvoie *Faux* si la tournée ne peut plus accepter de nouveaux nœuds.

Algorithme 12 Méthode 2 pour $Z_{c_1} > Z_{c_2}$ **Données:** $PVRPTW$ **Résultat:** S_{deb}, S_{fin} $Z'_{c_1} \leftarrow Z_{c_1}$ $S_{deb} \leftarrow \text{construction_des_tournees}(VRPTW_{Z_{c_2}})$ $S_{fin} \leftarrow S_{deb}$ **répéter** $k \leftarrow \text{Select_tournee_a_completer}(S_{deb}, S_{fin})$ **répéter** $OK \leftarrow \text{Inserer_noeud_c1_tournee}(Z'_{c_1}, k)$ **jusqu'à** $OK = \text{Faux}$ **jusqu'à** (plus de tournées à compléter) ou ($\text{Est_vide}(Z'_{c_1})$)**si** $\text{Not Est_vide}(Z'_{c_1})$ **alors** $S' \leftarrow \text{Construction_des_tournees}(Z'_{c_1})$ **fin si** $S_{deb} \leftarrow S_{deb} \cup S'$ $\text{Or_opt}(S_{deb})$ $\text{Or_opt}(S_{fin})$

3.4.2 Expérimentations

Les expérimentations ont également été réalisées sur les instances de Solomon que nous avons modifiées comme expliqué ci-après.

Soit I une instance à tester, I comporte un ensemble Z de clients à collecter avec $Z = \{z_1, \dots, z_n\}$ ayant chacun une quantité q_i .

Dans un premier temps, on fixe parmi l'ensemble Z des nœuds de l'instance I ceux collectés en c_1 et ceux collectés en c_2 ainsi $Z = Z_{c_1} \cup Z_{c_2}$. Puis on réplique le fichier d'instance en 3 exemplaires pour représenter les nœuds à collecter en c_1 , les nœuds à collecter en c_2 en début de semaine et en fin de semaine : I_{c_1} , I_{debc_2} et I_{finc_2} .

Dans I_{c_1} , les quantités à collecter sur les nœuds en c_1 sont augmentées : $q_i^{c_1} = q_i * \alpha \forall i \in Z_{c_1}$ avec $\alpha > 1$.

Dans I_{debc_2} , les quantités à collecter sur les nœuds en c_2 restent inchangées : $q_i^{debc_2} = q_i \forall i \in Z_{c_2}$.

Dans I_{finc_2} les quantités associées aux nœuds à collecter en c_2 sont diminuées : $q_i^{finc_2} = q_i * \beta \forall i \in Z_{c_2}$ avec $\beta < 1$.

De la même façon que dans la première hypothèse, les deux coefficients sont fixés de manière empirique à : $\alpha = 1,6$ et $\beta = 0,8$.

On a $|Z| = 100$ et nous allons tester trois valeurs pour la proportion de nœuds en c_1 : 60%, 70% et 80%.

Nous avons choisi les nœuds appartenant à Z_{c_1} de manière aléatoire sur chacune des instances.

3.4.3 Résultats et conclusion

Les résultats vont être présentés sur les différents objectifs de coût et de stabilité. Les deux méthodes explorées vont être comparées sur les différentes instances de Solomon et les différents nombres de nœuds en c_1 (60, 70, 80). Les résultats sont moyennés sur les différentes instances de chaque type (C1, R1, RC1, C2, R2 et RC2) et sur les différentes proportions de nœuds en c_1 .

Les tableaux détaillés des résultats obtenus par chaque instance sont présentés en Annexe A2.

Résultats en termes de coût

Les graphiques (3.23) répertorient les résultats obtenus sur l'objectif de coût correspond à la distance totale parcourue sur toute la semaine pour chaque méthode. Sur cet objectif, le meilleur résultat est fourni par la méthode 1, sauf pour les instances R1 et RC1 ayant un fort pourcentage de nœuds en c_1 (80% et 70% pour ce qui de RC1). Cependant, pour RC1, les valeurs en termes de coût pour les méthodes 1 et 2 sont très proches. Nous pouvons donc en conclure que sur cet objectif, la méthode 1 est préférable à la méthode 2.

Résultats en termes de stabilité globale

Les graphiques (3.24, 3.25 et 3.26) présentent les résultats concernant la stabilité globale. En moyenne, la méthode 1 utilise le même nombre ou plus de véhicules que la méthode 2, puisque l'écart sur le nombre de véhicule représente ici les véhicules ajoutés à ceux nécessaires à la collecte des nœuds en c_2 . Nous pouvons par ailleurs remarquer qu'en général, sur les instances proposant un nombre de véhicules moins important pour la méthode 2, nous avons également un coût moins important pour la méthode 2, ou une différence de coût faible entre la méthode 1 et la méthode 2.

Lorsque le nombre de véhicules est le même, nous pouvons constater que la méthode 1 donne des écarts de distance et de temps inférieurs, mais lorsque le nombre de véhicules est plus important, la différence sur la distance et le temps est en moyenne plus intéressante sur la méthode 2 (exemple, instances RC1 avec 80% de nœuds en c_1). Le fait d'avoir une tournée de plus provoque donc une différence de distance et de temps plus importante, ce qui semble logique.

Il semble donc qu'en ce qui concerne la stabilité globale, la méthode 2 domine la méthode 1.

Résultats en termes de stabilité du point de vue usagers

Les graphiques (3.27) représentent pour chaque méthode les écarts horaires sur les nœuds dans les différentes tournées. Nous pouvons remarquer que cet écart est relativement faible. Lorsque les écarts sont plus importants, c'est la méthode 1 qui donne les meilleurs résultats (instances C1, C2). Sur les autres instances les écarts sont faibles et révèlent en moyenne de meilleurs résultats sur la méthode 2.

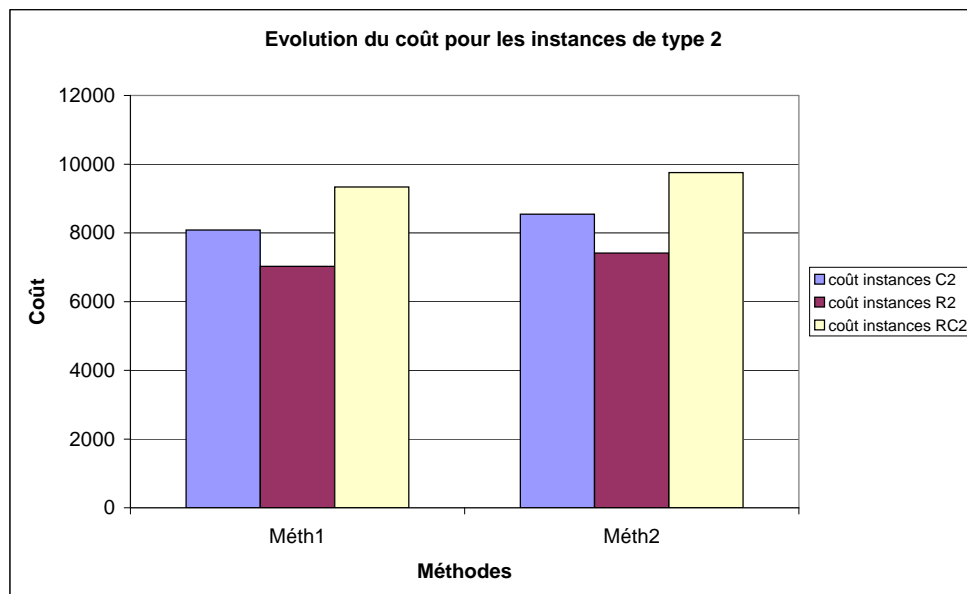
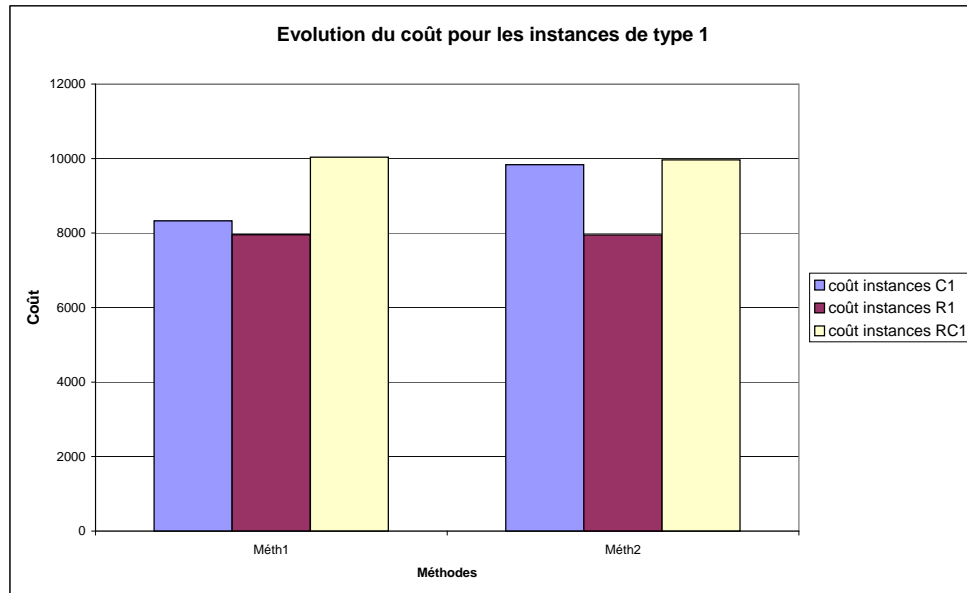


FIG. 3.23 – Comparaison des méthodes en termes de coût par type d’instances

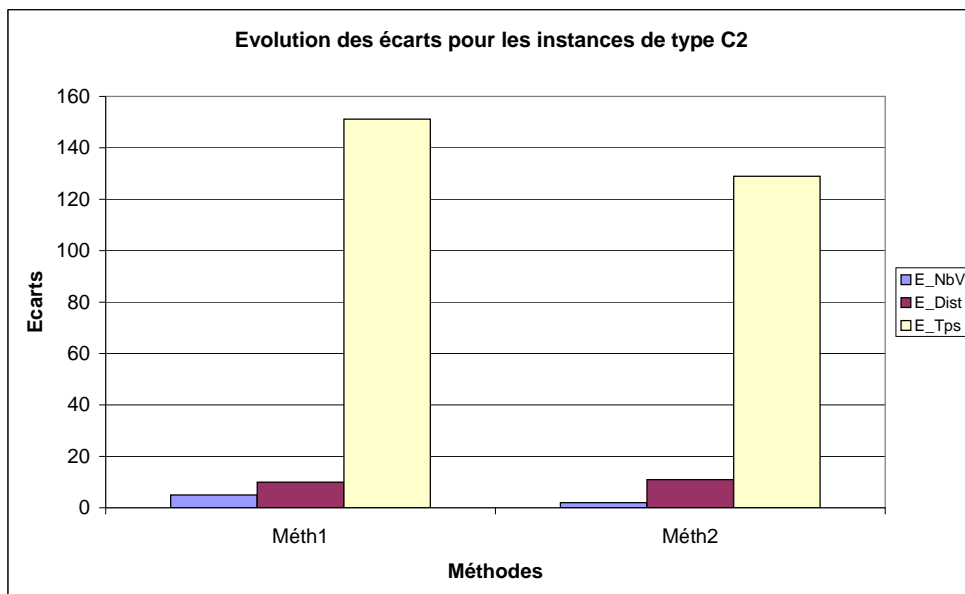
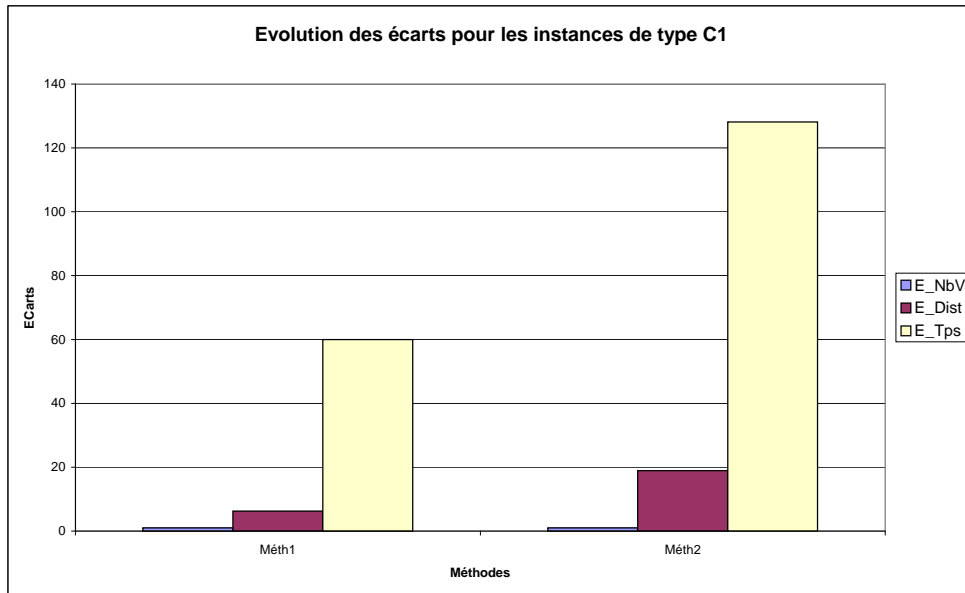


FIG. 3.24 – Comparaison des méthodes en termes de stabilité globale par type d'instances

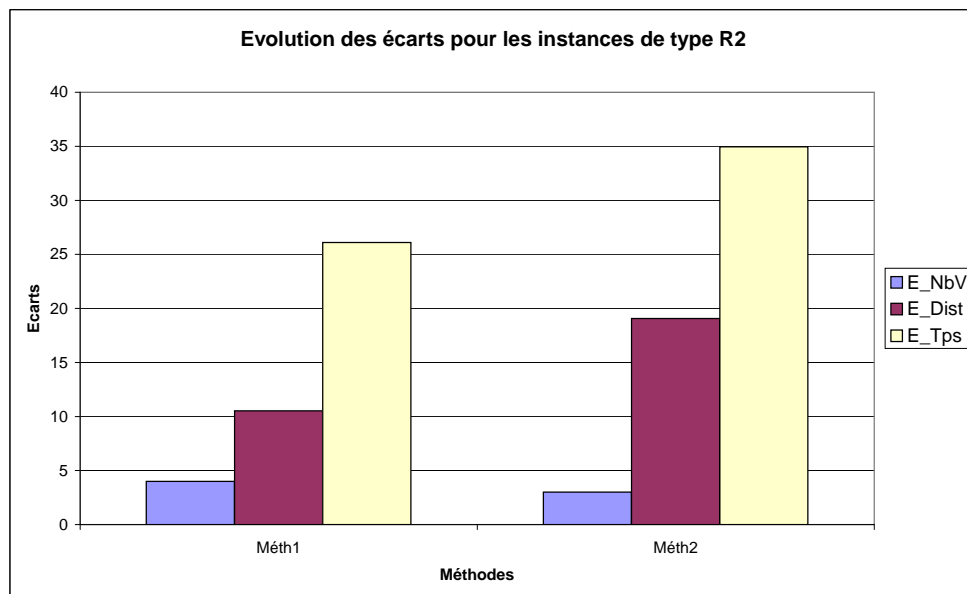
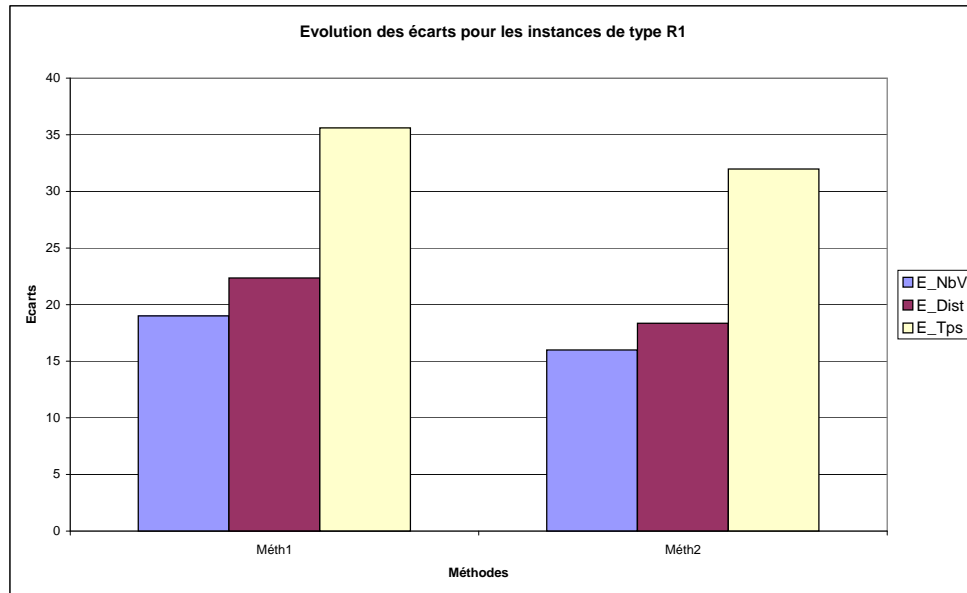


FIG. 3.25 – Comparaison des méthodes en termes de stabilité globale par type d'instances

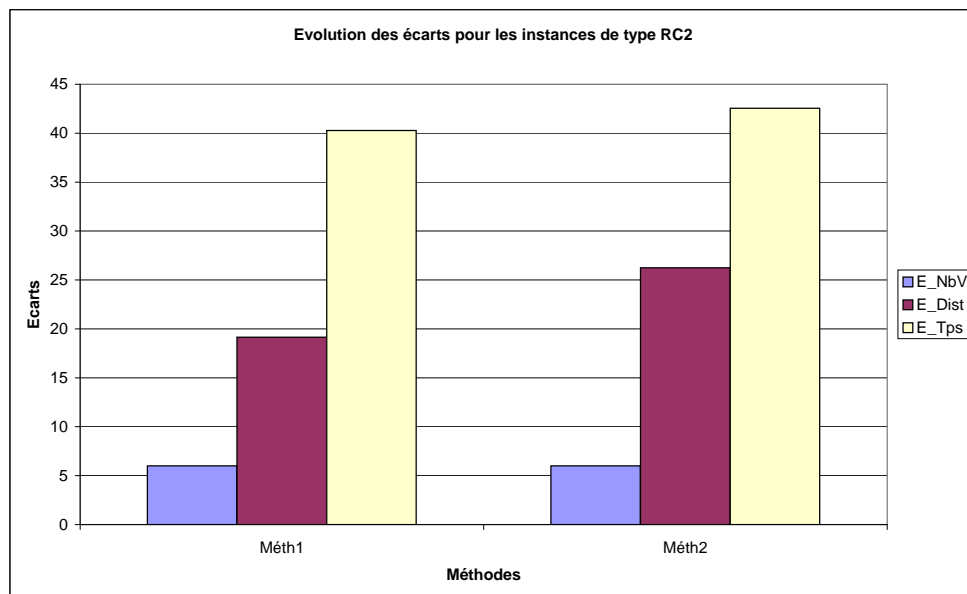
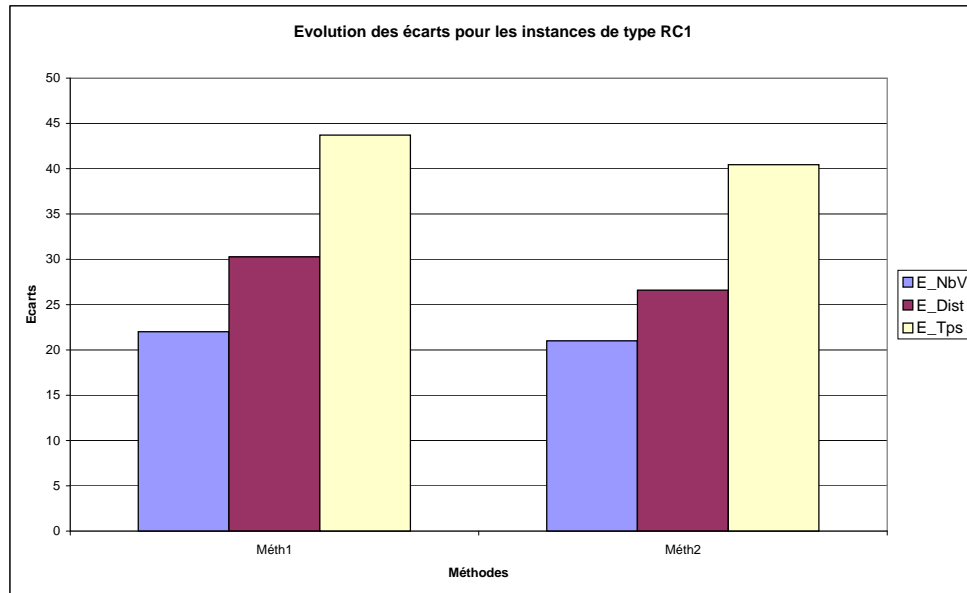


FIG. 3.26 – Comparaison des méthodes en termes de stabilité globale par type d'instances

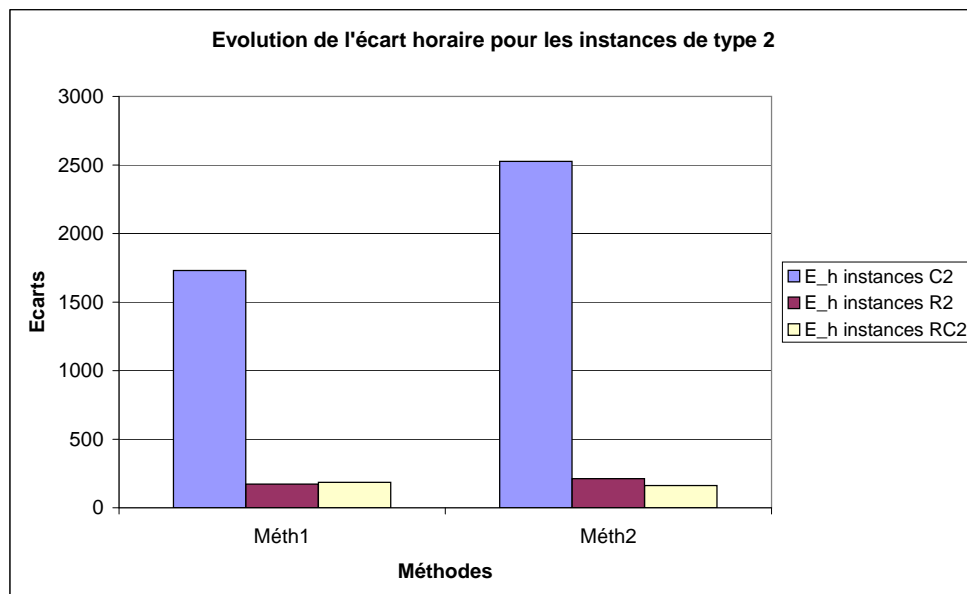
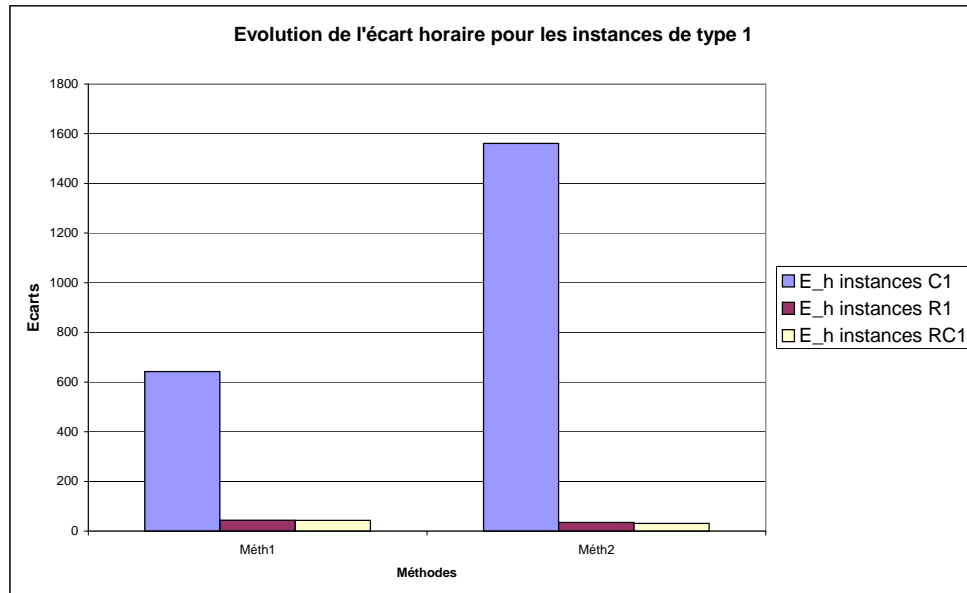


FIG. 3.27 – Comparaison des méthodes en termes de stabilité “usagers” par type d’instances

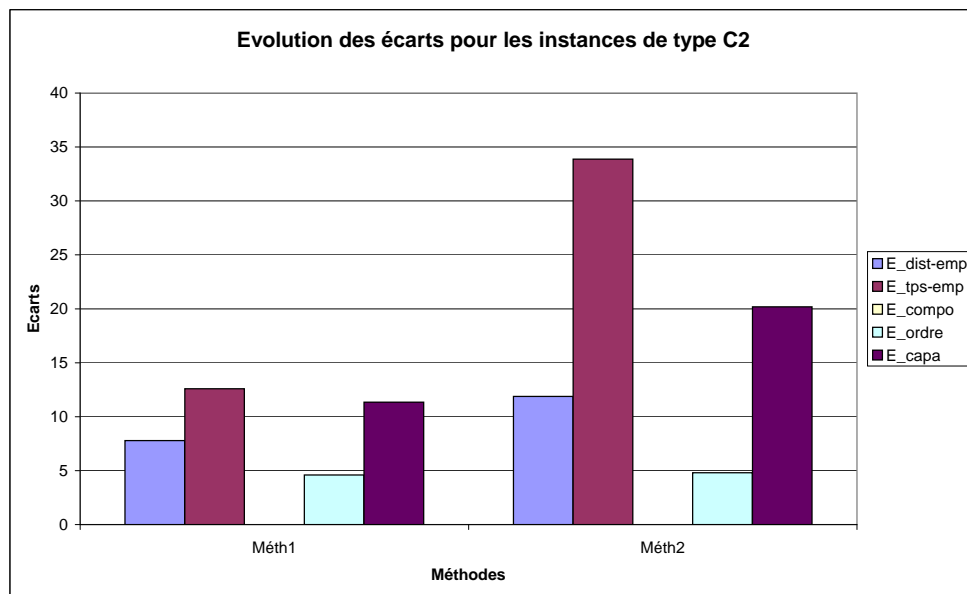
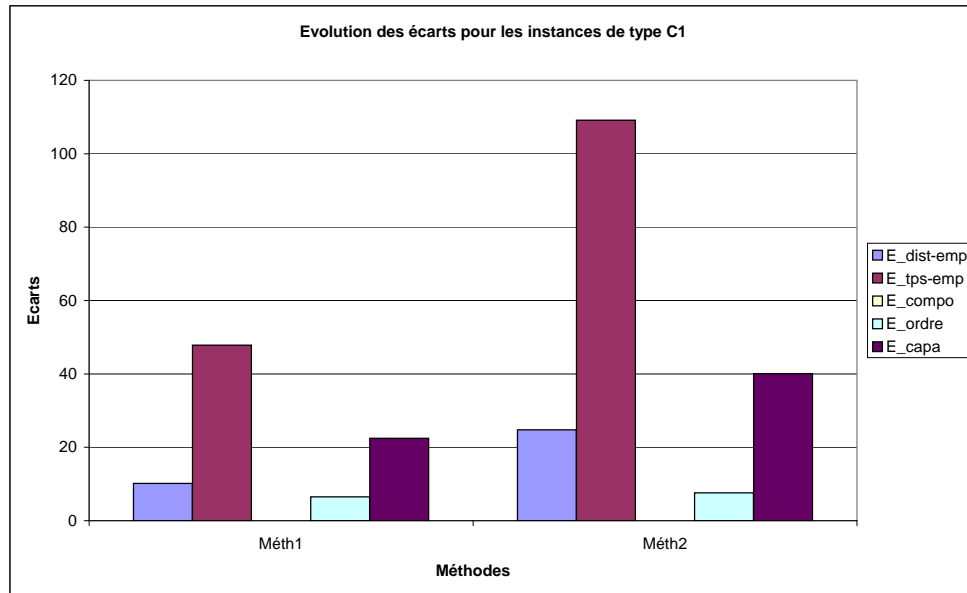


FIG. 3.28 – Comparaison des méthodes en termes de stabilité “employés” par type d’instances

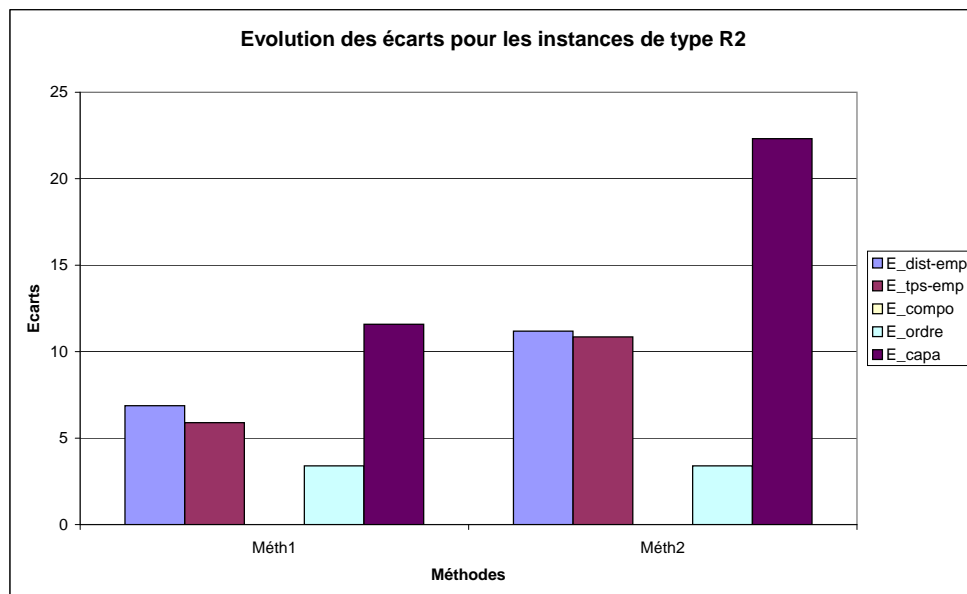
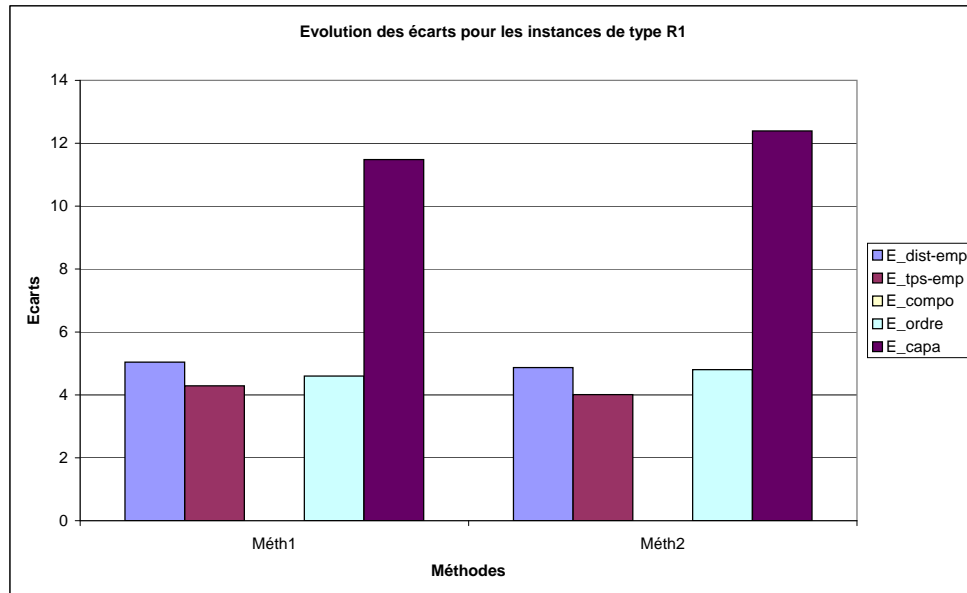


FIG. 3.29 – Comparaison des méthodes en termes de stabilité “employés” par type d’instances

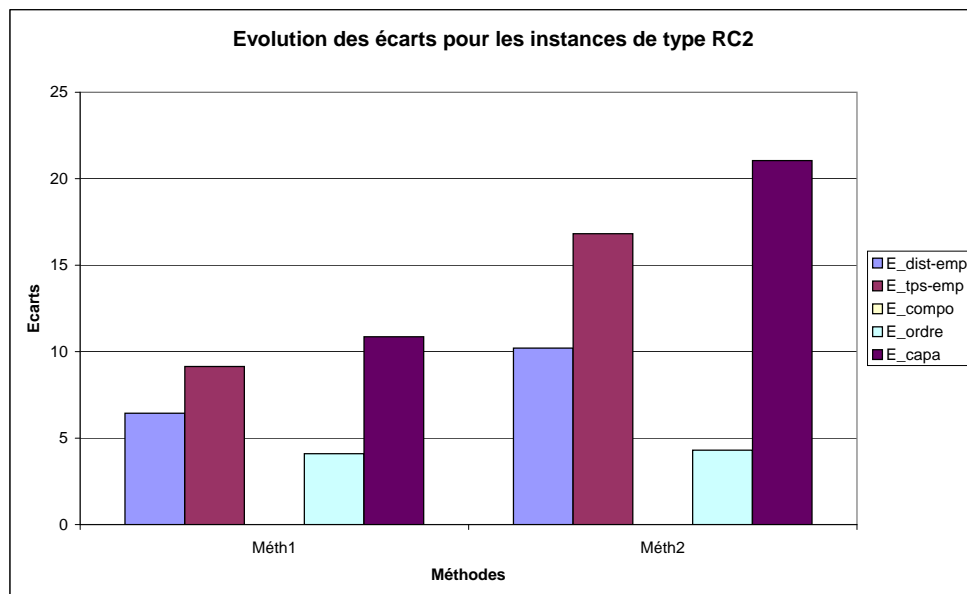
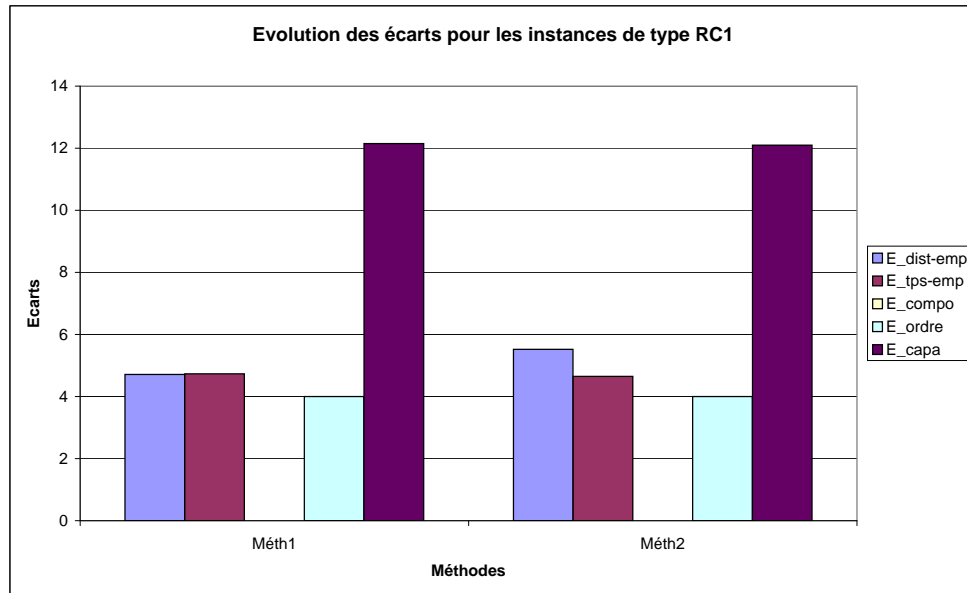


FIG. 3.30 – Comparaison des méthodes en termes de stabilité “employés” par type d’instances

Résultats en termes de stabilité du point de vue employés

Les graphiques (3.28, 3.29 et 3.30) donnent les résultats obtenus pour la stabilité du point de vue employés.

L'écart en distance est plus important pour la méthode 2. Nous retrouvons ce résultat pour l'écart en temps, en ordre et en capacité. L'écart concernant la composition est nul, ceci s'explique par le fait que les deux ensembles de tournées de début et de fin de semaine sont basés sur les mêmes tournées créées pour les nœuds collectés en c_2 . Il est donc logique que l'on obtienne les mêmes séries de nœuds sur chacun des ensembles de tournées.

Ces résultats nous font ressortir des caractéristiques sur les méthodes 1 et 2. Il semble que la méthode 2 intègre en moyenne plus de nœuds au sein des tournées déjà mises en place par rapport à la méthode 1. Cela signifie que la méthode 2 permet de réduire le nombre de véhicule par rapport à la méthode 1, mais en contre partie elle propose en moyenne des tournées plus chargées, même lorsque le nombre de véhicule supplémentaire est le même. La méthode 2 continue d'intégrer plus de nœuds au sein des tournées déjà construites.

Les deux méthodes présentées ont été créées pour minimiser le coût et les écarts entre les tournées élaborées pour le début et la fin de semaine. La différence notable avec l'étude effectuée dans l'hypothèse 1 est que les mesures de stabilité donnent des différences plus faibles.

En conclusion, on peut dire que la méthode 1 est mieux adaptée pour trouver des solutions qui minimisent les coûts tandis que la méthode 2 est mieux adaptée pour l'objectif de stabilité. Nous ne pouvons donc pas donner de meilleures méthodes qu'une autre, le choix diffère selon l'objectif prédominant du décideur.

3.5 Conclusion

En conclusion, nous avons mené dans ce chapitre une étude bi-objective de collecte hebdomadaire. Dans le problème qui nous préoccupe, il faut à construire deux ensembles de tournées pour la collecte des déchets, un premier ensemble concernant les tournées de collecte de début de semaine et un deuxième ensemble, celles effectuées en fin de semaine. Les objectifs associés à la construction de ces deux ensembles de collecte sont d'une part le coût et d'autre part la stabilité. Deux hypothèses ont été émises portant sur le nombre de nœuds à collecter une fois par semaine. En effet, selon les nœuds, les fréquences de collecte peuvent être différentes selon les nœuds, ce qui va avoir des conséquences pour la construction des ensembles de tournées de début et de fin de semaine.

Selon l'hypothèse formulée, nous avons proposé différentes méthodes de construction heuristiques de ces ensembles de tournées et proposé une étude de méthodes métaheuristiques multi-objectifs. Nous les comparons sur les instances de Solomon. Les résultats montrent que selon les instances, différentes méthodes heuristiques sont mises en avant. Elles permettent d'obtenir une solution sans avoir à fixer des préférences sur les objectifs. Les métaheuristiques proposent plusieurs solutions réparties sur le Front

Pareto, nous avons pu constater que globalement elles sont adaptées pour ce qui est de minimiser les coûts, tandis que les heuristiques semblent mieux adaptées pour ce qui est de minimiser la stabilité. Il est possible de choisir une méthode heuristique adaptée aux spécificités du problème, qui garantit non seulement un coût correct, mais également une stabilité intéressante. Comme vu dans le chapitre 1, de nombreuses études présentes des algorithmes évolutionnaires, nous n'avons pas développé ce type de métaheuristique mais il serait intéressant d'en développer une pour comparer nos résultats obtenus à l'aide des métaheuristiques utilisant la notion de voisinage.

Le compromis utilisé pour la stabilité afin de comparer ces méthodes est important, il serait donc intéressant de développer cette étude et de proposer différents compromis sur cet objectif. En effet, l'agrégation proposée est posée à priori et donc doit être fixée à l'avance pour utiliser les métaheuristiques. Ce compromis n'est pas à faire en ce qui concerne les méthodes heuristiques puisque celles-ci ne sont pas guidées par un objectif. Elles sont donc plus facilement exploitables si l'importance à priori des objectifs n'est pas fixée.

L'ensemble de ces résultats nous apprend que les deux objectifs ne sont pas forcément liés. En général les méthodes qui donnent de bons résultats sur l'objectif de coût donnent de mauvais résultats sur l'objectif de stabilité comme l'utilisation de méthode de reconstruction des tournées. Et à contrario, les méthodes donnant de bons résultats sur l'objectif de stabilité donnent de moins bons résultats sur l'objectif de coût comme la méthode 5 de l'hypothèse 1. C'est le cas également dans l'hypothèse 2 dans laquelle chacune des méthodes proposées est plus efficace sur un objectif. Il est difficile de hiérarchiser les objectifs et donc de définir lequel doit être prédominant sur l'autre et d'en faire ressortir les méthodes prédominantes.

Chapitre 4

Étude bi-objective de la stabilité dynamique

Dans ce chapitre, nous nous intéresserons au deuxième problème d'organisation de collecte mis en évidence pour la collectivité locale partenaire, c'est-à-dire le recalcul des tournées lorsque les informations de collecte ont évolué au cours du temps. Comme exposé au deuxième chapitre, la collectivité partenaire a un nombre d'habitations qui augmente chaque année en raison de sa position géographique de plus, la quantité de déchets augmente également chaque année. Ces évolutions dans les informations conduisent à la définition de nouvelles tournées. Ces tournées doivent être construites de telle sorte qu'elles ne s'éloignent pas trop de la solution opérationnelle déjà en place en exploitant un objectif de stabilité. Dans ce chapitre, nous allons proposer des méthodes de construction de tournées dédiées à ce problème dynamique. Ces méthodes vont être testées et comparées sur des instances de la littérature.

4.1 Description du problème

Comme vu au chapitre 1 concernant l'état de l'art, il existe plusieurs degrés de dynamisme dans un problème de tournées de véhicules [93]. Dans cette étude, nous rencontrons les deux volets des problèmes dynamiques :

- L'ensemble des variables du modèle ne sont pas toutes connues par avance, c'est le cas dans cette étude de nouvelles variables, représentant des nouveaux points de collecte vont apparaître. Nous pouvons par exemple faire un parallèle de ce problème avec le domaine des Constraint Solving Problem (CSP) dynamiques, dans lequel certaines variables sont inactives au début de la résolution, et sont activables ou inactivables selon le contexte [141].
- Les données du problème ne sont pas figées au début de l'étude, c'est le cas ici car les valeurs sur les quantités de certains points à collecter peuvent évoluer.

Nous sommes en présence de ces deux volets de dynamique : une évolution des variables et une évolution des valeurs des quantités à collecter.

Dans cette étude, le degré de dynamisme est faible. En effet, les évolutions évoquées peuvent être estimées par avance et ainsi être anticipées sur un certain horizon de temps. Nous ne sommes pas dans le cas particulier de l'évolution dynamique temps réel, où les données ne sont pas connues par avance mais doivent être prises en compte durant l'exécution opérationnelle des tournées. Ce cas particulier, dans lequel le degré de dynamisme est fort, n'est pas étudié dans ce mémoire.

Le but est donc double dans cette partie : proposer des méthodes d'absorption de nouveaux points de collecte ainsi que des méthodes d'absorption d'augmentation de charge à certains points de collecte. Ces méthodes sont conçues pour prendre en compte des évolutions à un instant t correspondant à l'instant définie par la collectivité pour absorber des nouvelles informations dans leur planification de tournées. C'est-à-dire que les réorganisations vont se faire régulièrement, par exemple tous les six mois, et être maintenue sur cette période (les évolutions peuvent être anticipées sur une longue période). Les méthodes proposées vont permettre de passer d'une solution courante à une nouvelle solution en prenant en compte deux objectifs. Les deux objectifs sont les mêmes que ceux qui ont été définis au chapitre précédent : l'objectif de coût, équivalent à minimiser la distance parcourue et l'objectif de stabilité, qui va être détaillé de la même façon que dans le chapitre précédent (de manière globale, selon le point de vue des usagers et celui des employés).

Pour étudier ce problème dynamique particulier nous allons indiquer la modélisation utilisée ; puis proposer une étude comparative pour les méthodes envisagées.

4.2 Modèle mathématique

La modélisation pour ce problème est basée sur le modèle classique de *VRPTW* vu au chapitre 1.

Comme vu au chapitre précédent, il existe deux ensembles de tournées sur la semaine, celles de début et celles de fin de semaine. Dans cette étude, nous allons nous intéresser séparément à l'évolution de chacun de ces ensembles de tournées. Ainsi nous disposons d'un ensemble de tournées dit de référence que nous allons faire évoluer pour prendre en compte de nouvelles informations.

Par exemple si un nouveau nœud apparaît et qu'il correspond à une collecte en c_1 (une fois par semaine), les tournées qui évoluent sont celles de début de semaine. Si ce nœud est collecté en c_2 (deux fois par semaine), nous considérons que l'intégration se fait à la fois sur le début de semaine et sur la fin de semaine indépendamment l'un de l'autre. Il est clair qu'ainsi nous pouvons remettre en cause la stabilité obtenue entre les tournées de début et de fin de semaine. Il est alors nécessaire d'utiliser des méthodes dédiées à la stabilité hebdomadaire pour rééquilibrer si besoin les tournées de début et de fin de semaine.

Pour caractériser notre problème dynamique, nous allons définir deux ensembles de tournées et adopter la notation suivante :

- S_{ref} : l'ensemble des tournées représentant la solution courante ;
- S_{evol} : l'ensemble des tournées représentant la nouvelle solution.

Le modèle de *VRPTW* est décrit au chapitre 1, nous allons juste rappeler les variables de décision du

modèle qui permettent de caractériser une solution :

- $x_{ij}^k = 1$ si l'arc (z_i, z_j) est dans la tournée du véhicule k et 0 sinon.
- $y_j^k = 1$ si le nœud z_j est visité par le véhicule k et 0 sinon.

Une solution S à un problème de VRPTW est caractérisée par l'ensemble des variables (x_{ij}^k, y_j^k) .

Dans ce chapitre, nous sommes en présence de deux VRPTW : $VRPTW_{ref}$ dans lequel figurent les variables et les données de départ et $VRPTW_{evol}$ correspondant au nouveau problème. Pour calculer les deux ensembles S_{ref} et S_{evol} nous allons, comme au chapitre précédent, utiliser les objectifs suivants :

- minimiser la distance totale parcourue pour S_{ref} ,
- minimiser la distance totale parcourue pour S_{evol} ,
- minimiser les écarts de S_{ref} par rapport à S_{evol} .

Les objectifs utilisés vont être communs avec ceux utilisés dans le chapitre 3, sauf que le critère de stabilité va être calculé entre les ensembles de tournées S_{ref} et S_{evol} .

L'objectif de coût représente la distance totale des nouveaux circuits (S_{evol}). En effet, la distance totale des tournées de S_{ref} est la même pour toutes les méthodes puisque ce sont les mêmes tournées de référence. Nous pouvons donc conserver uniquement la distance des circuits de S_{evol} .

Nous allons diviser l'étude en deux parties : une première partie traitant l'insertion de nouveaux points de collecte et une seconde partie concernant l'évolution des quantités à collecter.

4.3 Étude de l'insertion de nouveaux points de collecte

Les méthodes qui vont nous permettre d'intégrer de nouveaux nœuds sont décrites ci-dessous. Le but est de comparer les résultats selon les deux objectifs définis : celui de coût et celui de stabilité. La première méthode heuristique proposée a été élaborée afin de minimiser les écarts entre les deux ensembles. Les autres méthodes utilisent des heuristiques ou métaheuristiques de post-optimisation pour en améliorer le ou les objectifs. Les heuristiques sont de type méthodes de descente et les métaheuristiques utilisées sont de deux types : une première mono-objectif qui va chercher à minimiser uniquement l'objectif de coût et une seconde bi-objectif qui va chercher à réaliser un compromis entre l'objectif de coût et celui de stabilité.

Nous noterons également par la suite :

- Z_{ref} : ensemble des nœuds de la solution initiale ;
- Z_{evol} : ensemble des nouveaux nœuds à intégrer.

4.3.1 Méthodes d'insertion

Toutes les méthodes proposées se basent sur les circuits déjà élaborés pour la solution courante S_{ref} . Pour construire la première solution de référence S_{ref} , nous utilisons la même méthode que celle utilisée au chapitre 3 : l'algorithme de Solomon suivi d'heuristiques de descente "String-exchange" et "Or-opt" (l'algorithme 1).

Description de la méthode 1

La première méthode, nommée ici I1, intègre les nouveaux points de collecte aux circuits les plus proches de manière à respecter les contraintes. Pour cela, nous utilisons la même procédure d'insertion des nœuds que celle utilisée dans le chapitre 3. C'est-à-dire que pour chaque nœud on choisit l'insertion qui minimise la distance totale de la tournée dans laquelle le nœud s'ajoute (et qui respecte les contraintes du problème) comme illustré sur la figure (4.1). Si cela s'avère non faisable, nous ajoutons une tournée vide à l'ensemble S_{ref} pour permettre l'insertion de l'ensemble des nœuds à S_{ref} .

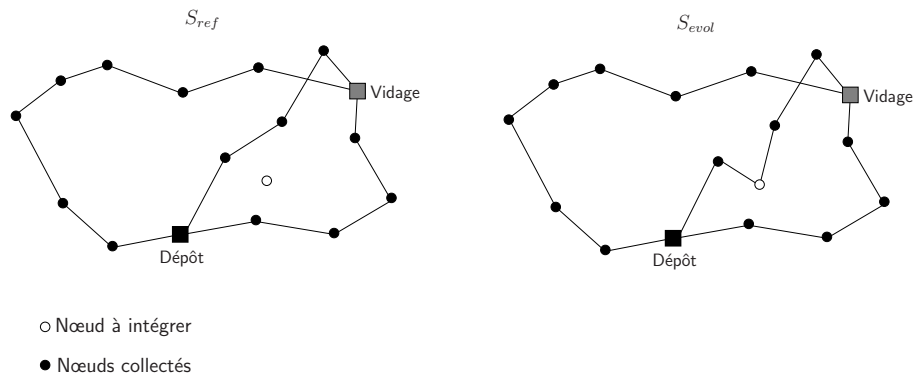


FIG. 4.1 – Méthode I1

L'algorithme 13 résume cette méthode.

Algorithme 13 Méthode I1 d'insertion de nouveaux points de collecte

Données: S_{ref} , $VRPTW_{evol}$

Résultat: S_{evol}

$S \leftarrow S_{ref}$

pour tous $y \in Z_{evol}$ **faire**

$OK \leftarrow \text{Insérer_noeud}(y, S)$

si not_OK **alors**

$\text{Ajouter_vehicule}(S)$

fin si

fin pour

$S_{evol} \leftarrow S$

La procédure *Insérer_Noeuds* (y, S) insère un nœud y dans la solution S . Cette fonction retourne *Faux* si une insertion n'a pas pu se faire. Un véhicule est alors ajouté.

Description de la méthode 2

La deuxième méthode, nommée I2, utilise I1 et améliore les circuits obtenus à l'aide de méthodes de descente (vu au chapitre 1 et 3) "String-exchange" et "Or-opt". La méthode ne les applique que sur les tournées qui ont été affectées par un changement (apparition d'un nouveau nœud) et sur les tournées "proches" de celles affectées. Nous définissons que deux tournées sont proches si il existe un lien direct

4.3. Étude de l'insertion de nouveaux points de collecte

(autrement dit sur un réseau routier une route reliant au moins un nœud de chaque tournée). Dans les instances que nous utilisons, le graphe est complet il existe donc une route qui relie tous les nœuds du graphe, nous représentons alors cette notion de “proche” par une distance minimale entre deux nœuds. L'algorithme 14 décrit cette méthode.

Algorithme 14 Méthode I2 d'insertion de nouveaux points de collecte

Données: S_{evol} par I1

Résultat: S_{evol} mise à jour

```
pour  $k \leftarrow Tournee\_affectee(k, S_{evol}) = Vrai$  faire
  pour  $k' \in Tournees\_proches(k)$  et  $k' \neq k$  faire
     $String\_exchange(k, k')$ 
     $Or\_opt(k')$ 
  fin pour
   $Or\_opt(k)$ 
fin pour
```

La procédure $Tournees_affectee(k, S_{evol})$ renvoie $Vrai$ si la tournée k a été affectée par l'insertion de un ou plusieurs nœuds lors de l'application de I1. La procédure $Tournees_proches(k)$ retourne la liste des tournées proches de k .

Description de la méthode 3

La troisième méthode, nommée I3, utilise I1 et améliore les circuits obtenus à l'aide de méthodes de recherche locale “String-exchange” et “Or-opt” entre chaque paire de tournées comme décrit dans l'algorithme 15.

Algorithme 15 Méthode I3 d'insertion de nouveaux points de collecte

Données: S_{evol} par I1

Résultat: S_{evol} mise à jour

```
pour tournées  $k \in S_{evol}$  faire
  pour tournées  $k' \in S_{evol}$  et  $k \neq k'$  faire
     $String\_exchange(k, k')$ 
     $Or\_opt(k')$ 
     $Or\_opt(k)$ 
  fin pour
fin pour
```

Description de la méthode 4

La méthode 4, nommée I4, utilise I1 et améliore les circuits par la suite à l'aide d'une recherche tabou suivant le critère de distance. Nous allons présenter la méthode de recherche tabou (vu au chapitre 3) que nous rappelons dans l'algorithme 16. Le voisinage utilisé est le même que celui définit au chapitre 3 pour la métaheuristique multi-objectifs.

Algorithme 16 Méthode I4 d'insertion de nouveaux points de collecte : Méthode tabou mono-objectif**Données:** S_{evol} par I1**Résultat:** S_{fin} $S \leftarrow S_{evol}$ $Best_sol \leftarrow S$ **tant que** critère d'arrêt non atteint **faire** $V \leftarrow Selectionner_voisinage(string - exchange, or - opt)$ **pour tout** $S' \in V(S)$ **faire**Selectionner meilleure solution S' admissible et non tabou dans $V(S)$ **fin pour****si** $cout(S') < cout(S)$ **alors** $Best_sol \leftarrow S'$ **fin si** $S \leftarrow S'$ Ajouter S' dans la liste Tabou**fin tant que** $S_{fin} \leftarrow Best_sol$ **Description de la méthode 5**

La cinquième méthode, nommée I5 utilise I1 et améliore les circuits à l'aide d'une métaheuristique d'amélioration bi-objective, qui prend en compte l'objectif de la distance et celui de stabilité comme l'algorithme 9 décrit au chapitre 3 .

Cette méthode est basée sur la méthode bi-objective de recherche tabou présentée dans le chapitre 3. Le résultat de la recherche multi-objective peut être représenté par un Front Pareto. Ce front correspond aux solutions non-dominées que nous pouvons trouver.

L'objectif de coût correspond à la distance totale parcourue par les véhicules sur S_{evol} . L'objectif de stabilité est agrégé de la manière suivante :

$$stabilité = E_{dist-emp} + E_{tps-emp} + E_h * 0,05 + E_{compo} + E_{ordre} + E_{capa}$$

L'accent a été mis sur la stabilité des usagers et des employés, pour faire ressortir ces deux aspects primordiaux de la stabilité, ce sont eux qui vont être affectés par les changements. Cette agrégation (la même que celle utilisée au chapitre 3) correspond à différents tests effectués sur la pondération des écarts. L'impact de l'écart horaire est minimisé par rapport aux autres car c'est celui qui donne en général la plus forte valeur numérique.

Une fois le Front Pareto obtenu sur les deux objectifs évoqués ci-dessus, pour ne conserver qu'une unique solution, et ainsi agréger l'ensemble des résultats pour chaque type d'instance, nous appliquons un compromis sur l'objectif de coût et de stabilité de la manière suivante :

$$compromis = 0,90 * cout + 0,10 * stabilité$$

La valeur numérique de la stabilité étant beaucoup plus forte que celle de coût, les coefficients ont été élaborés pour équilibrer les deux objectifs. Avec ce compromis, la méthode I5 fournit une seule solution et pourra être comparée simplement aux autres méthodes.

Description de la méthode 6

La sixième méthode, nommée I6, recalcule les tournées à partir des nouvelles données à l'aide de l'algorithme de construction des tournées du chapitre 3 (l'algorithme 1) comme décrit dans l'algorithme 17.

Algorithme 17 Méthode I6 d'insertion de nouveaux points de collecte

Données: $VRPTW_{ref}$ et $VRPTW_{evol}$

Résultat: S_{ref} et S_{evol}

$S_{ref} \leftarrow construction_des_tournees(VRPTW_{ref})$

$S_{evol} \leftarrow construction_des_tournees(VRPTW_{evol})$

4.3.2 Expérimentations

Les expérimentations ont été réalisées sur les instances de Solomon (décrites au chapitre 3) adaptées au problème considéré.

Soit I une instance à tester, I comporte un ensemble Z de clients à collecter avec $Z = \{z_1, \dots, z_n\}$ ayant chacun une quantité.

Dans un premier temps, on sélectionne parmi l'ensemble Z les nœuds qui sont collectés dans S_{ref} et ceux qui sont ajoutés pour former S_{evol} . On duplique chaque fichier d'instance I de la manière suivante : le premier fichier I_{ref} comporte les nœuds à collecter pour $VRPTW_{ref}$ et le second I_{evol} contient les nœuds à rajouter pour former $VRPTW_{evol}$.

Pour effectuer les expérimentations, un pourcentage de nœuds est choisi dans I pour former l'ensemble I_{evol} .

On a $|Z| = 100$ et on effectue deux tests :

- $|Z_{ref}| = 95$ et $|Z_{evol}| = 5$, ce qui correspond à une croissance de 5,2% du nombre de nœuds à collecter.
- $|Z_{ref}| = 85$ et $|Z_{evol}| = 15$, ce qui correspond à une croissance de 17,6% du nombre de nœuds à collecter.

Cette évolution du nombre de nœuds à collecter est supérieure à celle observée annuellement par la collectivité (2%) mais permet de mieux mettre en valeur les caractéristiques des méthodes élaborées et notamment le fait de devoir ajouter un véhicule.

Nous avons fait les expérimentations en choisissant aléatoirement les nœuds à insérer.

4.3.3 Résultats et conclusion

Dans cette partie, nous allons présenter les résultats sous forme de graphiques en répertoriant pour chaque type d'instances les résultats obtenus sur les deux objectifs en moyenne sur les deux variations du nombre de nouveaux nœuds à collecter. Les résultats pour ces expérimentations vont être présentés tout d'abord sur l'objectif de coût, puis sur l'objectif de stabilité. Afin de mieux appréhender la stabilité, nous allons la décrire selon les différents niveaux de stabilité. L'ensemble des résultats obtenus pour les différentes méthodes sont fournis en Annexe B.

Résultats en termes de coût

Le critère de coût correspond à la distance totale des tournées de S_{evol} .

Les graphiques (4.2) répertorient les résultats obtenus sur l'objectif de coût pour chaque type d'instances.

Pour toutes les instances, les méthodes I4 et I6 donnent les meilleurs résultats.

La méthode 5 donne des résultats sensiblement similaires aux méthodes I2 et I3. La méthode I1 donne les plus mauvais résultats sur cet objectif.

Pour ce qui est des autres méthodes, elles sont intermédiaires et nous pouvons noter que les différences entre elles sont faibles pour ce qui est de l'objectif de coût.

Sans surprise pour ce qui est de l'objectif de coût, la méthode de reconstruction ou l'utilisation d'une métaheuristique optimisant cet objectif sont les meilleures méthodes à envisager.

Résultats en termes de stabilité globale

Les graphiques (4.3, 4.4 et 4.5) présentent les résultats obtenus sur l'objectif de stabilité globale. Dans ces graphiques certaines méthodes nécessitent plus de véhicules pour l'insertion de 15 nœuds.

Pour les indicateurs E_{dist} et E_{tps} , nous pouvons séparer les résultats entre les instances qui ajoutent un véhicule et les autres.

- Pour les instances où un véhicule est ajouté (C1 et R2), nous pouvons constater que la méthode I6 donne de très bons résultats pour l'écart en distance, mais pas de bons résultats pour ce qui est de l'écart de temps. La méthode I1 donne les moins bons résultats. Pour ce qui est des autres méthodes, elles donnent des compromis entre les deux méthodes extrêmes.
- Pour les autres instances, nous pouvons séparer selon leur type.
 - Pour les instances de type 1, les méthodes I2 et I3 donnent les meilleurs résultats, tandis que la méthode I6 donne les plus mauvais.
 - Pour les instances de type 2, les méthodes I2 et I3 donnent les meilleurs résultats, tandis que les méthodes I4 et I5 donnent les plus mauvais, la méthode I6 donne de bons résultats pour E_{dist} mais mauvais pour I6

En résumé sur ces objectifs il n'y a pas de règle, cela dépend des instances et de leur type. Malgré tout les méthodes I2 et I5 semblent donner de bons résultats en général sur l'ensemble des instances.

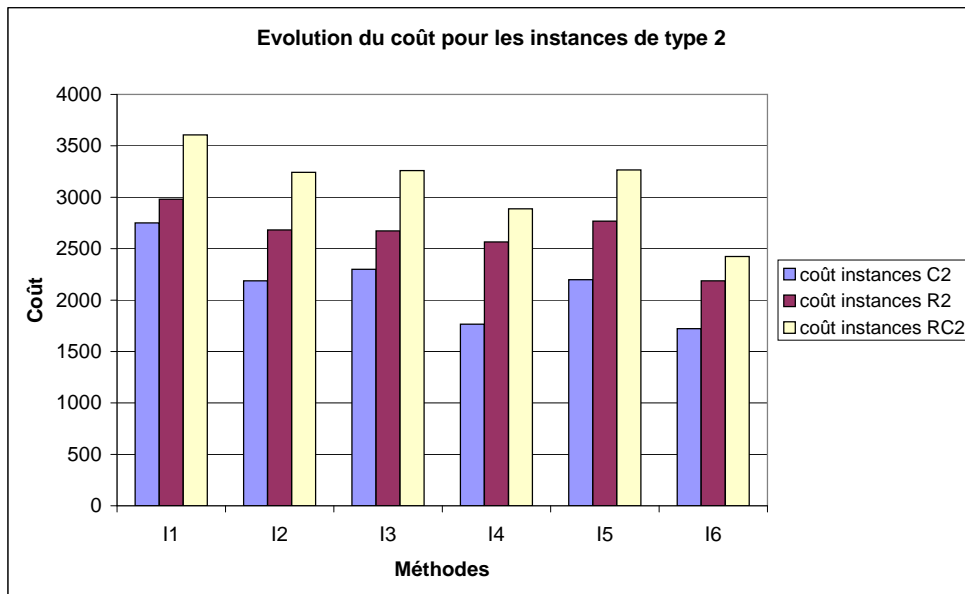
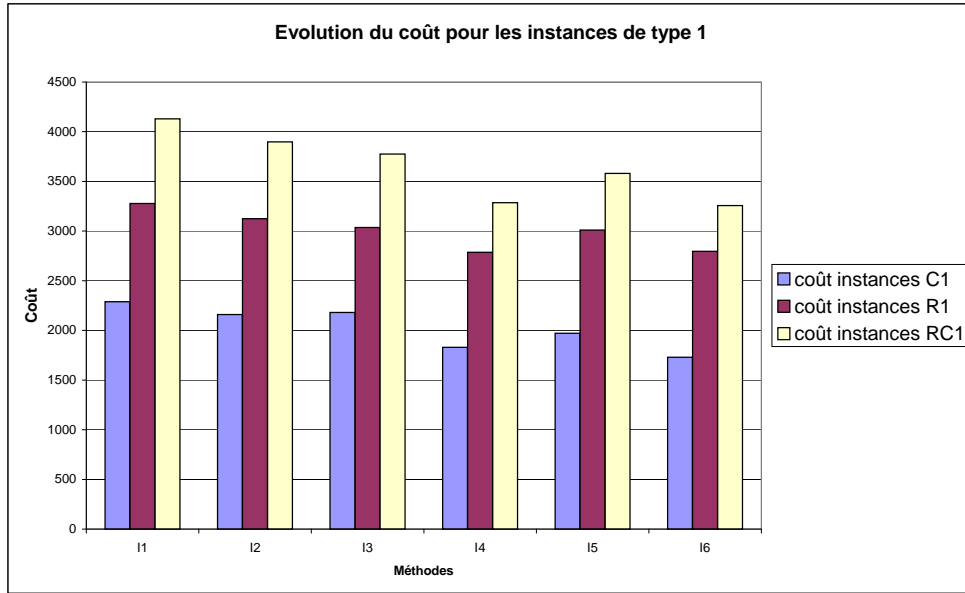


FIG. 4.2 – Comparaison des méthodes en termes de coût par type d'instances

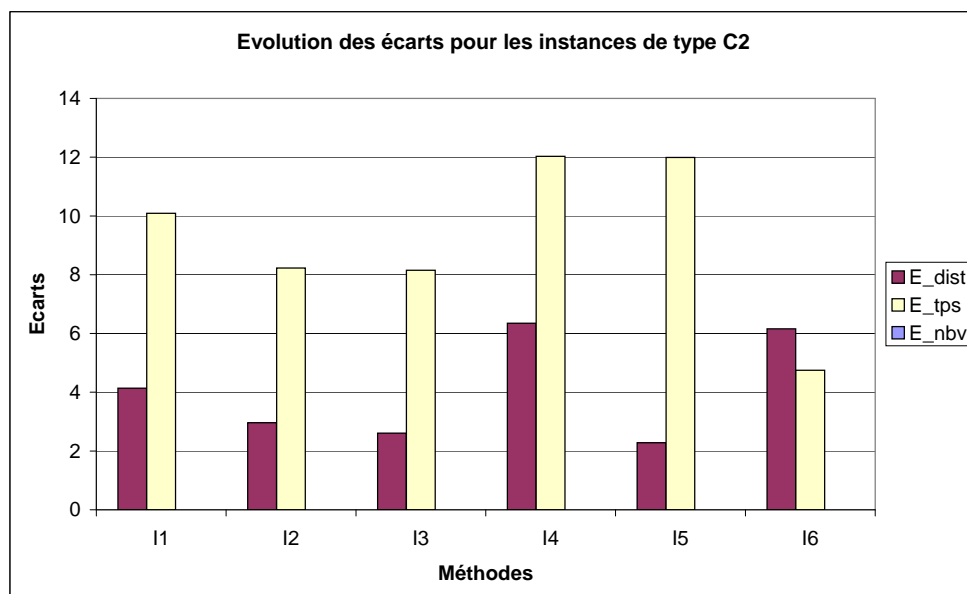
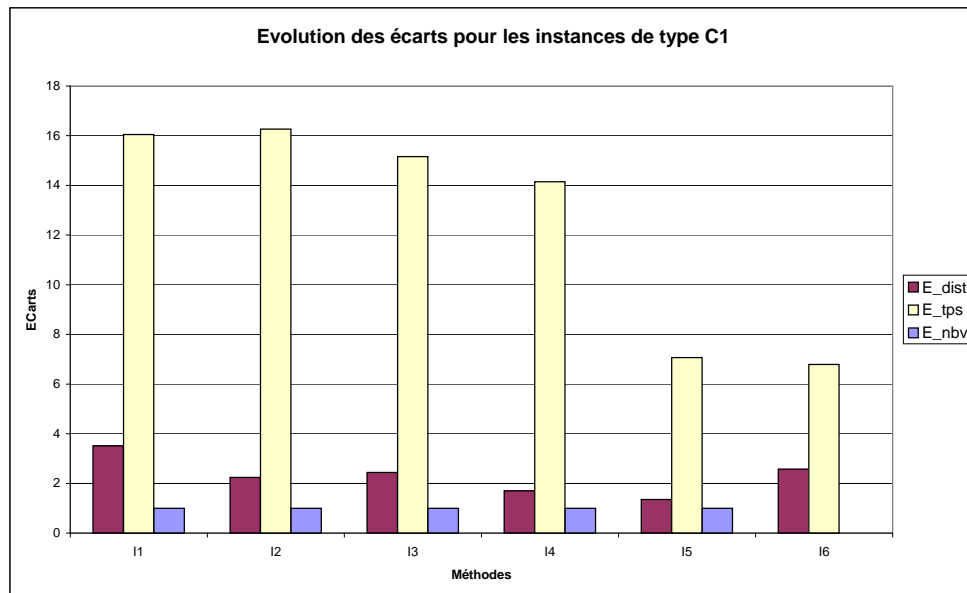


FIG. 4.3 – Comparaison des méthodes en termes de stabilité globale par type d'instances

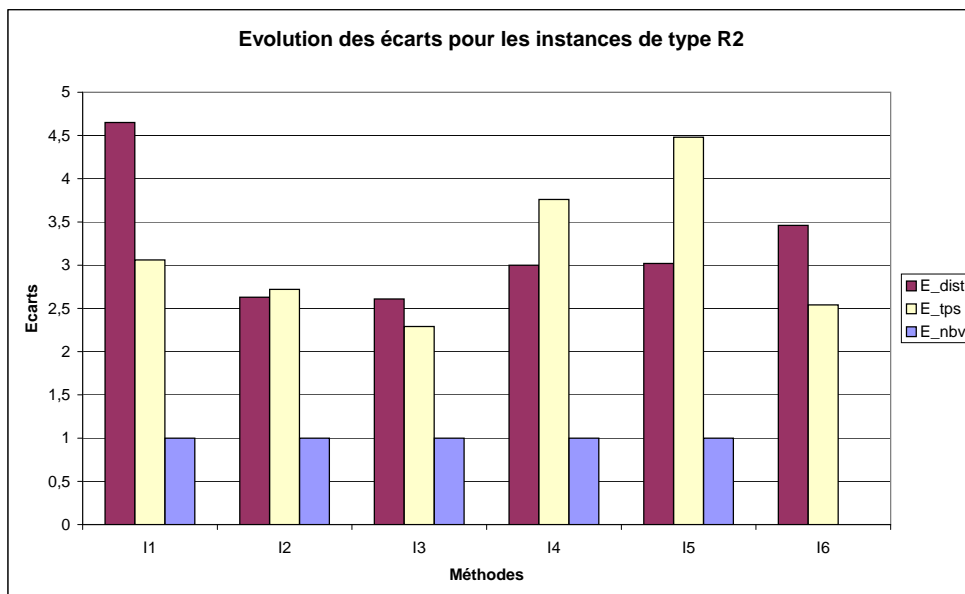
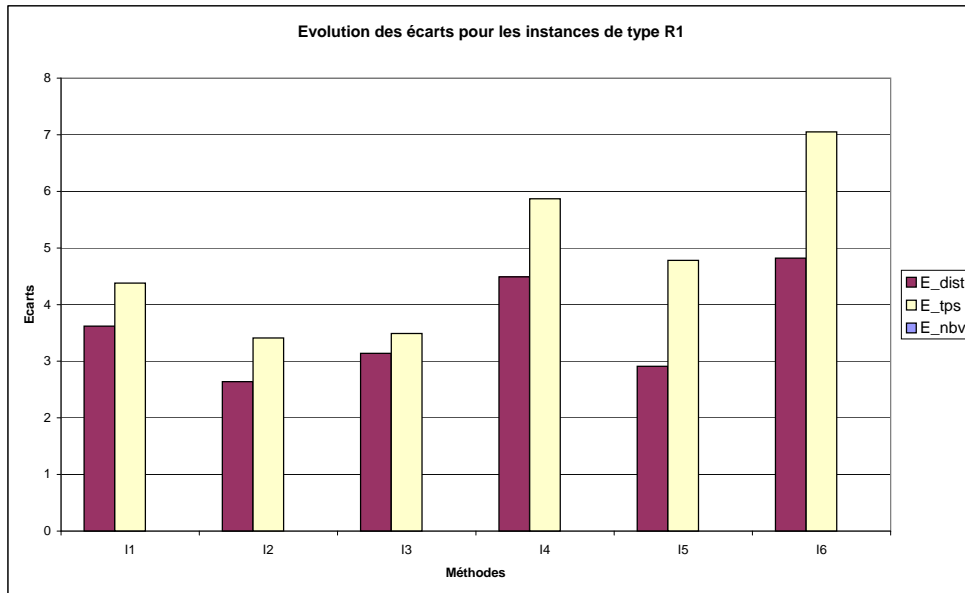


FIG. 4.4 – Comparaison des méthodes en termes de stabilité globale par type d'instances

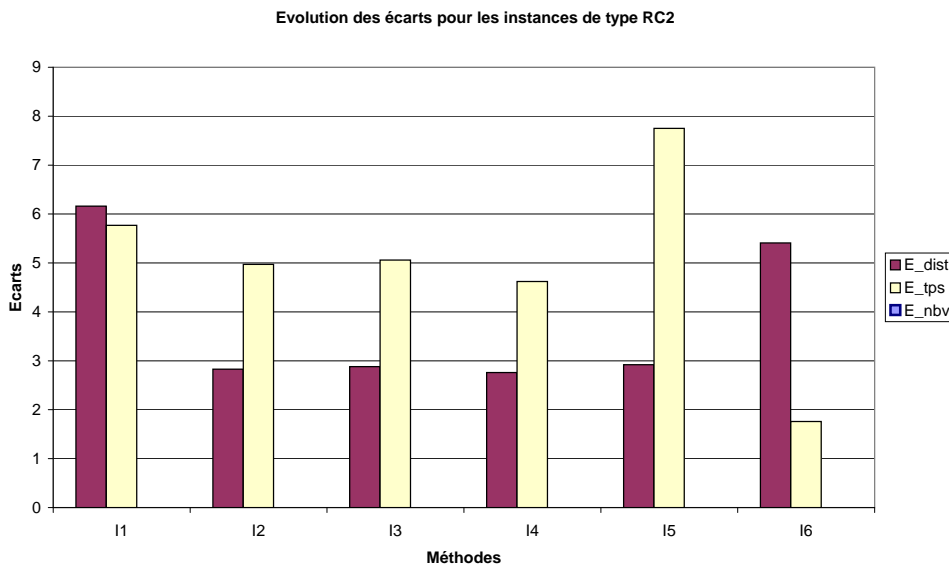
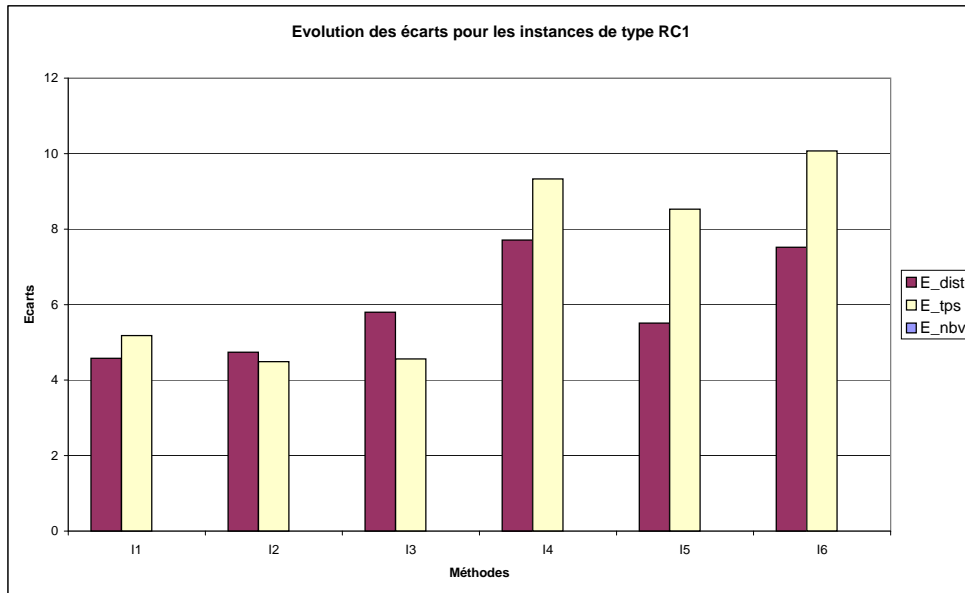


FIG. 4.5 – Comparaison des méthodes en termes de stabilité globale par type d'instances

Résultats en termes de stabilité du point de vue usagers

Les graphiques (4.6) présentent les résultats obtenus sur l'objectif de stabilité du point de vue des usagers, c'est-à-dire l'écart horaire. Le meilleur écart est celui donné par la méthode I1 et le moins bon est celui donné par la méthode I6. Les méthodes I2, puis I3 sont très proches l'une de l'autre. Les méthodes I4 et I5 se trouvent à la suite, la méthode I5 donnant en général de meilleurs résultats que la méthode I4.

Résultats en termes de stabilité du point de vue employés

Les graphiques (4.7, 4.8 et 4.9) répertorient les résultats obtenus sur l'objectif de stabilité du point de vue employés.

Pour toutes les courbes, les meilleurs résultats sont donnés par la méthode I1, puis les méthodes I2 et I3 qui donnent sensiblement les mêmes résultats. Ce sont celles qui garantissent la meilleure stabilité "employé".

Puis nous pouvons constater que la méthode I5 qui utilise une métaheuristique bi-objective donne en général de meilleurs résultats que les méthodes restantes. Elle semble donc bien adaptée pour être un compromis entre les différents indicateurs de stabilité. Enfin les méthodes I4 et I6 ne donnent pas de bons résultats sur cet indicateur de stabilité. Ce sont les deux méthodes qui ne sont pas "guidées" pour prendre en compte cet objectif, il est donc logique de le retrouver sur les résultats.

Conclusion

En conclusion, nous constatons que certaines méthodes sont plus adaptées pour minimiser l'objectif de coût (les méthodes I4 et I6) et d'autres pour minimiser l'objectif de stabilité (les méthodes I1, I2 et I3). Les méthodes I2 et I3 sont très proches sur l'ensemble des résultats. Il n'est donc pas nécessaire de conserver ces deux différentes méthodes.

La méthode I5 se trouve être un compromis entre le coût et la stabilité. Le compromis ici a été proposé à priori en pondérant les différents indicateurs de stabilité puis en pondérant cette stabilité avec l'objectif de coût. Il semblerait donc intéressant de prendre en compte différentes pondérations sur les objectifs pour proposer un ensemble de solutions.

Dans la partie suivante, nous nous intéressons à l'augmentation de charge sur certains nœuds et proposons de la même manière des comparaisons de méthodes.

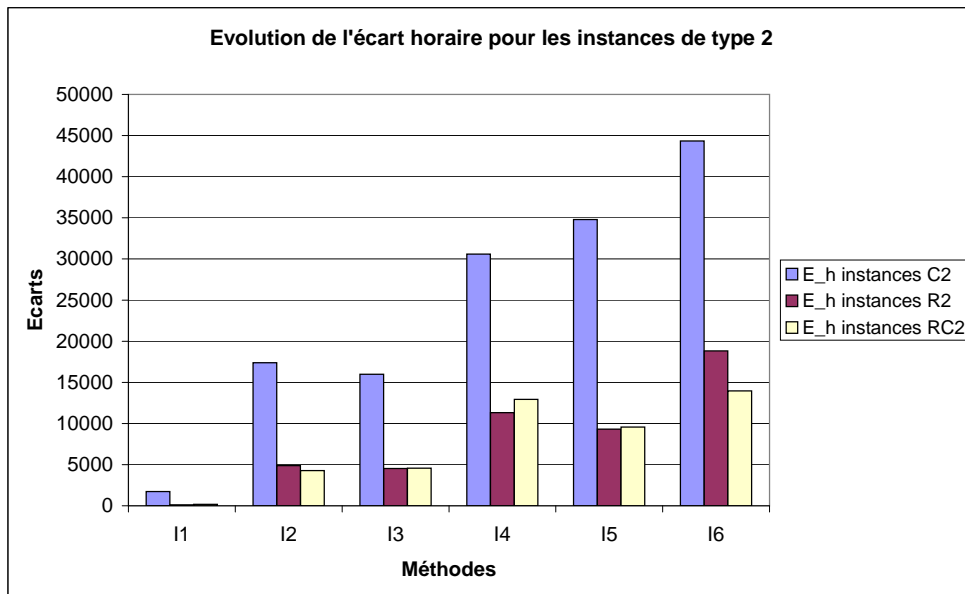
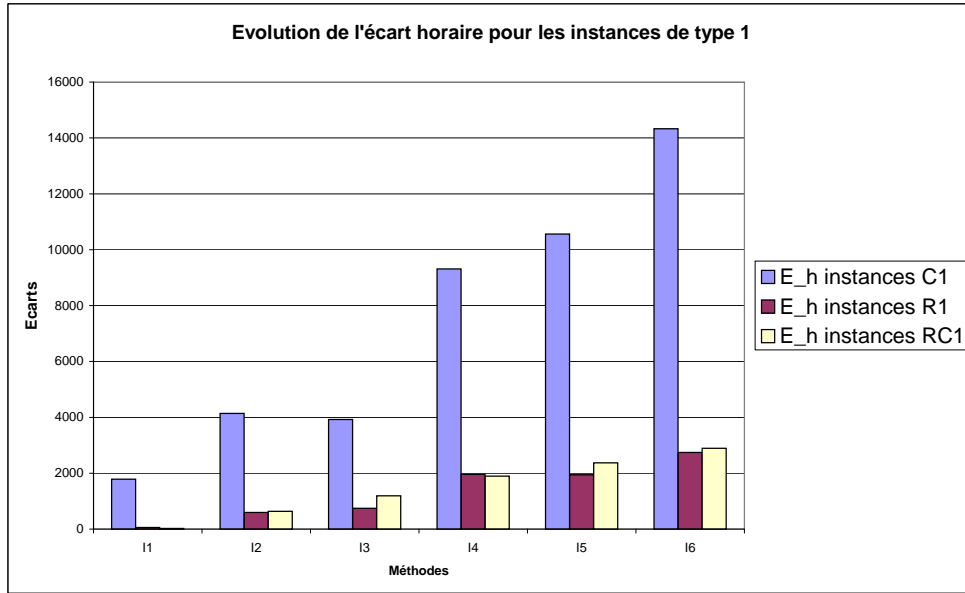


FIG. 4.6 – Comparaison des méthodes en termes de stabilité “usagers” par type d’instances

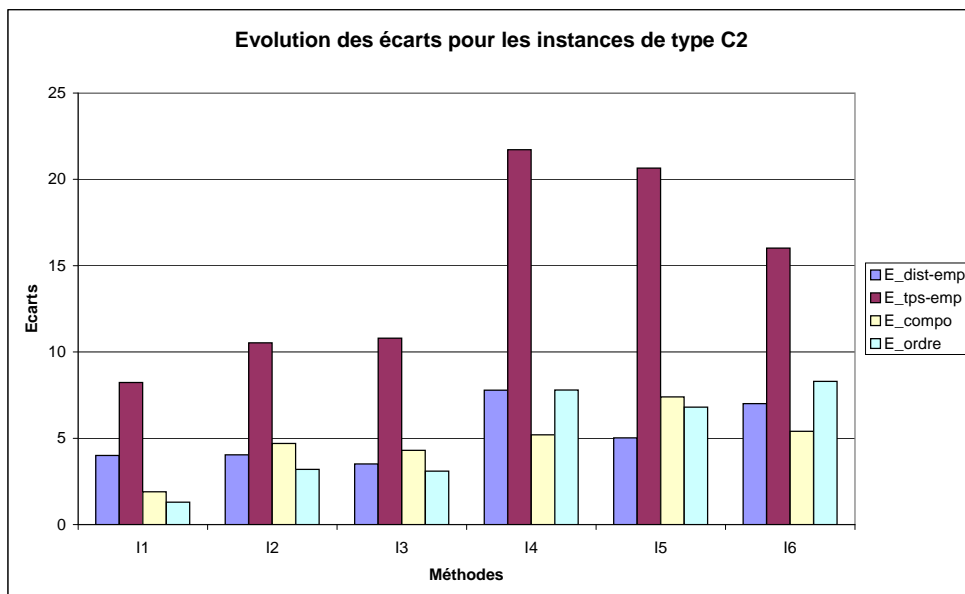
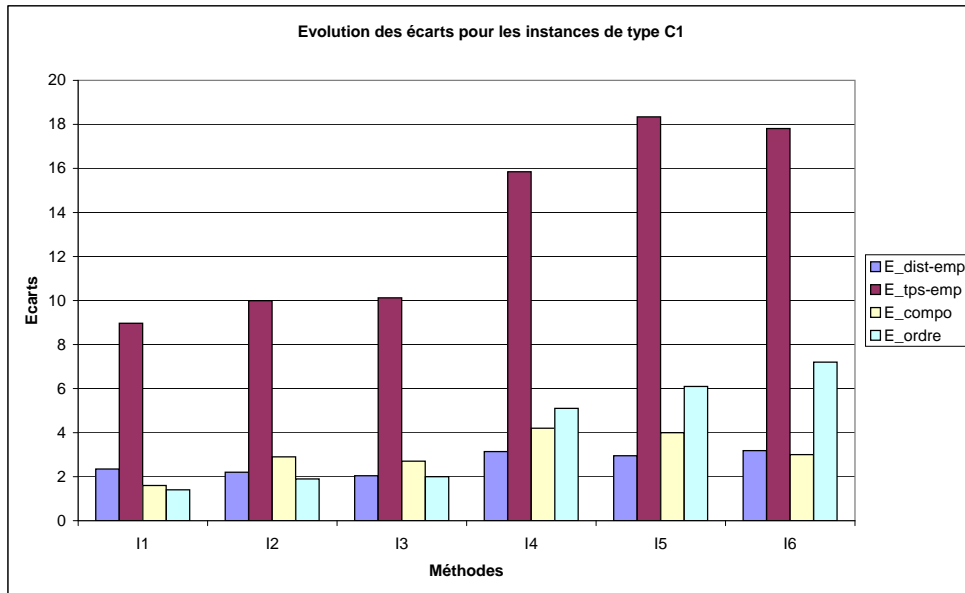


FIG. 4.7 – Comparaison des méthodes en termes de stabilité “employés” par type d’instances

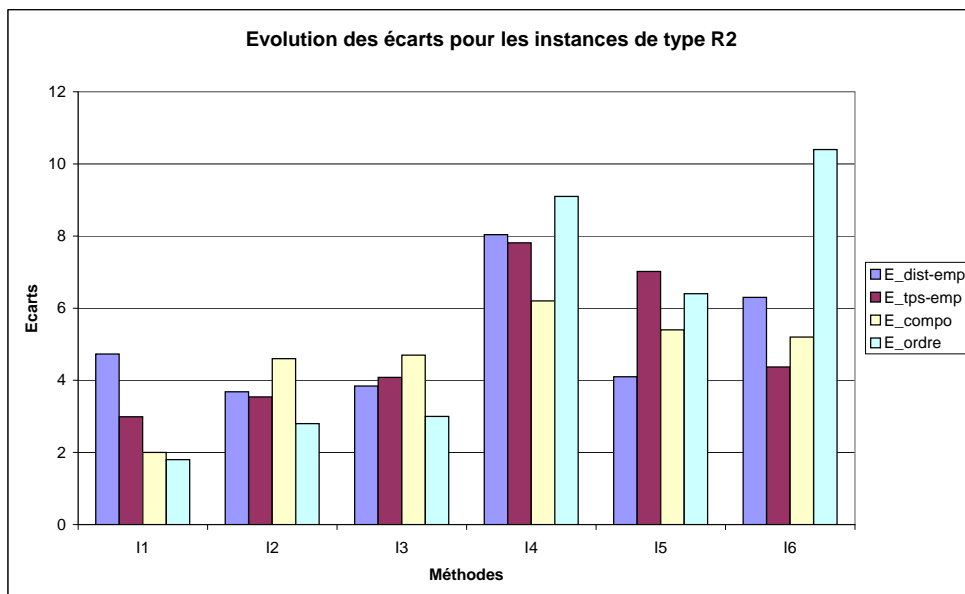
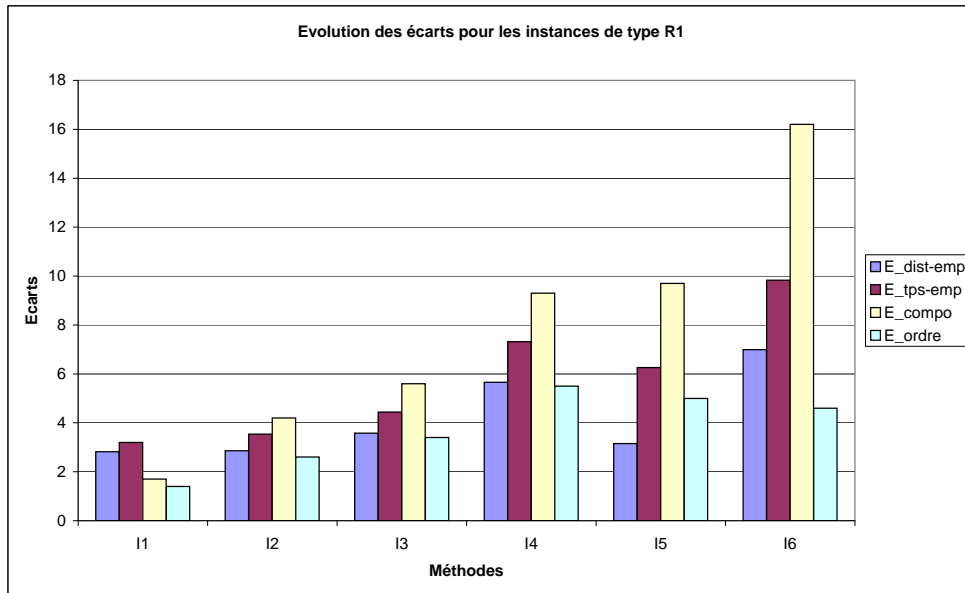


FIG. 4.8 – Comparaison des méthodes en termes de stabilité “employés” par type d’instances

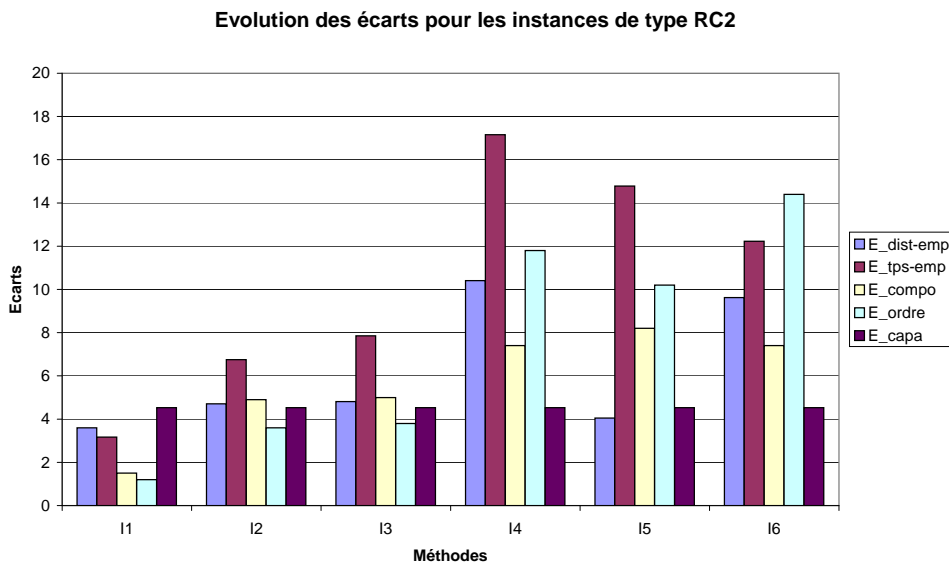
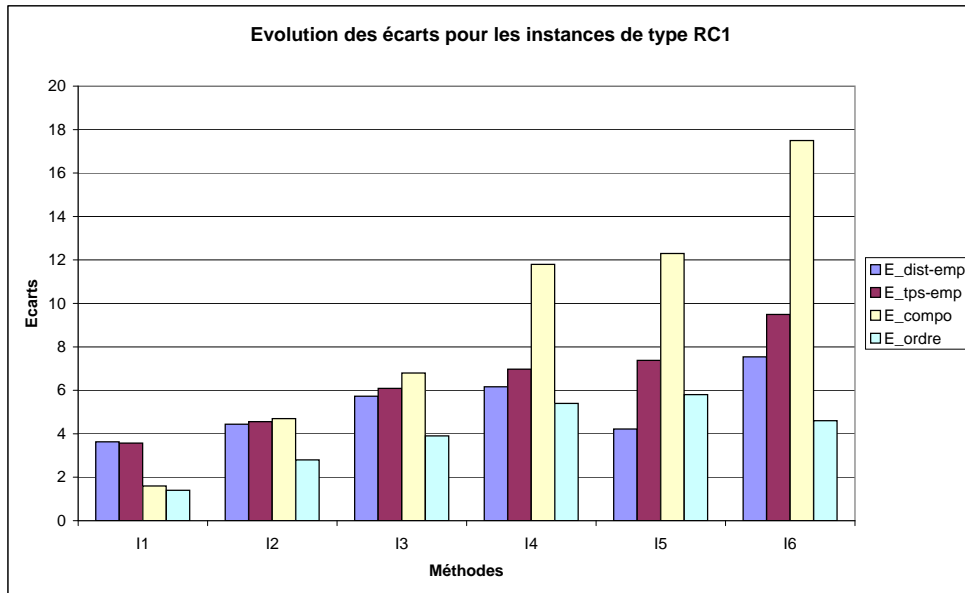


FIG. 4.9 – Comparaison des méthodes en termes de stabilité “employés” par type d’instances

4.4 Augmentation de la charge à collecter

L'étude menée dans cette partie concerne l'augmentation de charge sur certains points de collecte. Le but est de comparer différentes méthodes d'absorption de cette nouvelle charge. Lorsque la charge à collecter augmente sur certains points de collecte, cela peut provoquer une surcharge de certaines tournées. Ces tournées doivent donc se décharger d'un ou de plusieurs nœuds sur les autres tournées. Lorsqu'une solution admissible ne peut être trouvée, il faut rajouter un nouveau véhicule de collecte.

Toutes les méthodes dédiées décrites ci-dessous se basent sur l'élaboration préalable des circuits de collecte initiaux à l'aide de l'algorithme de Solomon suivi des méthodes de descente "String-exchange" et "Or-opt" (l'algorithme 1).

4.4.1 Méthode d'absorption

Description de la méthode 1

La première méthode, nommée A1 construit les tournées de S_{evol} en recopiant tout d'abord les tournées de S_{ref} en modifiant la charge des nœuds affectés par une augmentation. Puis lorsque certaines tournées se retrouvent en surcharge, des nœuds sont déplacés dans des circuits voisins de manière à respecter les contraintes. Si ce n'est pas possible la méthode sépare une tournée en deux comme illustré sur la figure (4.10). L'algorithme 18 résume cette méthode.

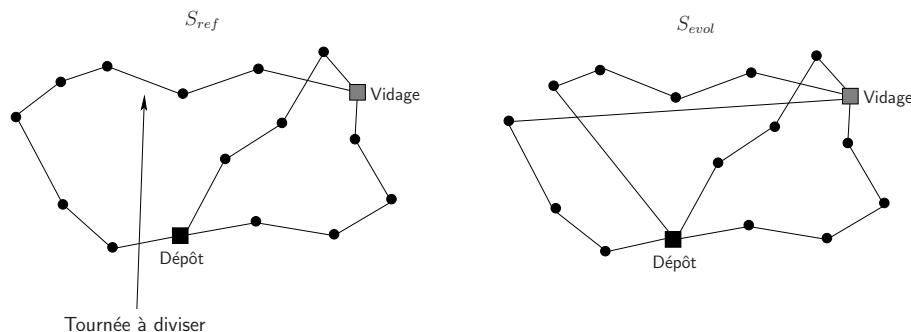


FIG. 4.10 – Méthode A1

La procédure $Actualiser_Capacité(S_{evol}, VRPTW_{evol})$ va mettre à jour les quantités des nœuds. $Tournee_surchargee(S_{evol})$ retourne la tournée k en surcharge de S_{evol} si elle existe. La procédure $Transférer_noeuds(k, S_{evol})$ va transférer des nœuds de la tournée k vers les tournées les moins chargées de S_{evol} si c'est possible. Si cela s'avère impossible la procédure $Solution_Admissible(S_{evol})$ va retourner $Faux$. Dans ce cas, la méthode va chercher la tournée k la plus chargée de S_{evol} à l'aide de $Chercher_Tournee_la_plus_chargee(S_{evol})$ et la procédure $Diviser_Tournee(k', S_{evol})$ sépare en deux la tournée k' de S_{evol} .

Algorithme 18 Méthode 1 d'augmentation de quantité sur certains points de collecte

Données: $S_{deb}, VRPTW_{evol}$

Résultat: S_{evol}

$S_{evol} \leftarrow S_{deb}$

$Actualiser_Capacite(S_{evol}, VRPTW_{evol})$

tant que il existe des tournées surchargées dans S_{evol} **faire**

$k \leftarrow Tournee_surchargee(S_{evol})$

$OK \leftarrow Transferer_noeuds(k, S_{evol})$

si not OK alors

$k' \leftarrow Tournee_la_plus_chargee(S_{evol})$

$Diviser_Tournee(k', S_{evol})$

fin si

fin tant que

Description de la méthode 2

La deuxième méthode, nommée A2, utilise la méthode A1 et améliore les circuits fournis à l'aide de méthodes de descente ("String-exchange" et "Or-opt") entre chaque paire de tournées. Comme pour la méthode I3, "String-exchange" est appliqué entre chaque paire de tournée. Elle est suivie de "Or-opt" appliquée à chacune des tournées.

Description de la méthode 3

La troisième méthode, nommée A3 utilise la méthode A1 et améliore les circuits obtenus à l'aide d'une recherche tabou cherchant à minimiser le coût (même principe que la méthode I4).

Description de la méthode 4

La quatrième méthode, nommée A4 utilise la méthode A1 et améliore les circuits à l'aide d'une recherche tabou prenant en compte à la fois le coût et la stabilité (même principe que la méthode I5).

Description de la méthode 5

La cinquième méthode, nommée A5, recalcule les tournées à partir des données brutes et propose donc une nouvelle solution à partir de l'algorithme de Solomon suivi des méthodes d'amélioration (l'algorithme 1).

4.4.2 Expérimentations

Les expérimentations ont été menées sur les instances à 100 clients de Solomon, décrites dans le chapitre 3. Ces instances ont dû être modifiées pour intégrer des variations de quantité de collecte sur certains nœuds.

Soit I une instance à tester, I comporte un ensemble Z de clients à collecter avec $Z = \{z_1, \dots, z_n\}$ ayant chacun une quantité q_i .

Dans un premier temps, l'instance I est dupliquée pour former I_{ref} et I_{evol} . I_{ref} représente la collecte de référence à effectuer et I_{evol} représente l'ensemble des nœuds dont la quantité est changée.

Dans I_{ref} les quantités à collecter sont diminuées par rapport à celles de I : $q_i^{ref} = q_i * \alpha \forall i \in Z$ avec $\alpha < 1$. Dans I_{evol} les quantités à collecter sur les nœuds restent inchangées : $q_i^{evol} = q_i \forall i \in Z$.

Nous avons choisi aléatoirement 20 nœuds dont la capacité varie et nous avons effectué 2 tests sur cette variation : +20% et +50%.

Nous avons fait les expérimentations en choisissant aléatoirement les nœuds dont la capacité varie.

4.4.3 Résultats et conclusion

Les résultats des expérimentations sont présentés tout d'abord par rapport à l'objectif de coût puis par rapport aux différents indicateurs de stabilité (globale, du point de vue des usagers et du point de vue des employés).

Les graphiques suivants donnent les résultats en moyenne pour chaque type d'instance et pour les deux tests sur les variations de capacité. Les tableaux contenant les résultats sont présentés en Annexe B.

Résultats en termes de coût

Les graphiques (4.11) répertorient les résultats obtenus sur l'objectif de coût pour chaque instance. Rappelons que le coût correspond à la distance totale des tournées de S_{evol}

Sur l'ensemble des graphiques, nous remarquons que la méthode A5 donne en général la meilleure solution ou une solution proche de la meilleure, et que la méthode A1 donne la moins bonne solution.

- Pour les instances de type 1, sur les instances de type C1, les méthodes A2 et A5 donnent les meilleures solutions et sur les autres instances, les méthodes A3 et A5 donnent les meilleures solutions.
- Pour les instances de type 2, les méthodes A2 et A5 donnent les meilleurs résultats avec la méthode A3. Nous pouvons remarquer que la méthode A2 donne souvent les mêmes résultats que la méthode A5, et que pour les instances C2 la méthode A3 donne les meilleurs résultats.

Pour cet objectif, la méthode 5 est plus recommandée, puis pour certains types d'instances la méthode 2 donne de bons résultats ainsi que la méthode 3.

Résultats en termes de stabilité globale

Les graphiques (4.12, 4.13 et 4.14) présentent les résultats obtenus sur l'objectif de stabilité globale. Pour ce qui est de cet objectif nous pouvons remarquer que la méthode A1 donne les meilleurs résultats et que les méthodes A2 et A5 donnent les moins bons résultats.

- Pour les instances de type 1 : pour les instances de type C1, il n'y a pas de véhicule ajouté par A1, ce qui implique que les différents écarts sont nuls. L'écart en temps est nul car nous n'avons pas modifié les temps de service en fonction de la quantité à collecter. Puis nous pouvons observer

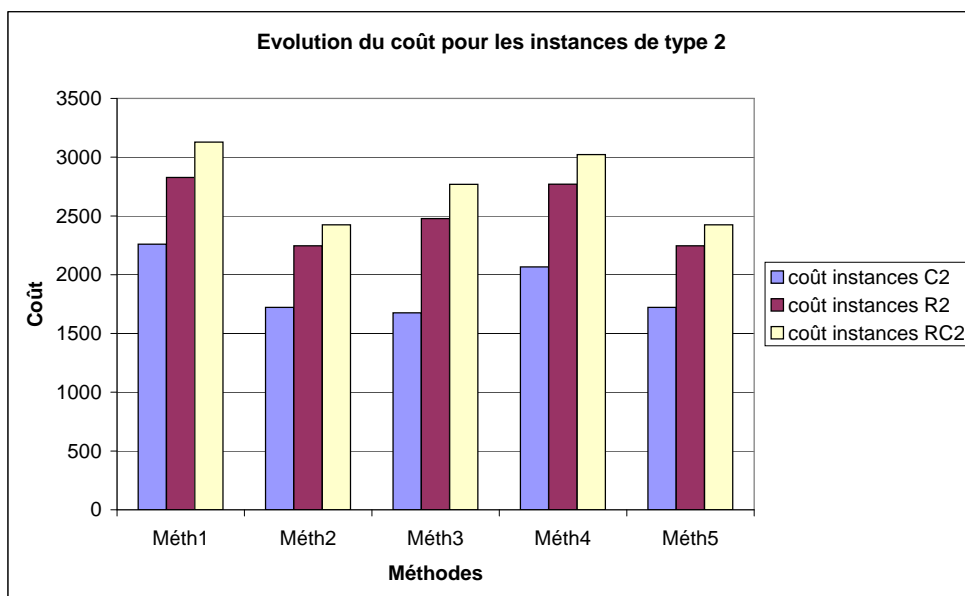
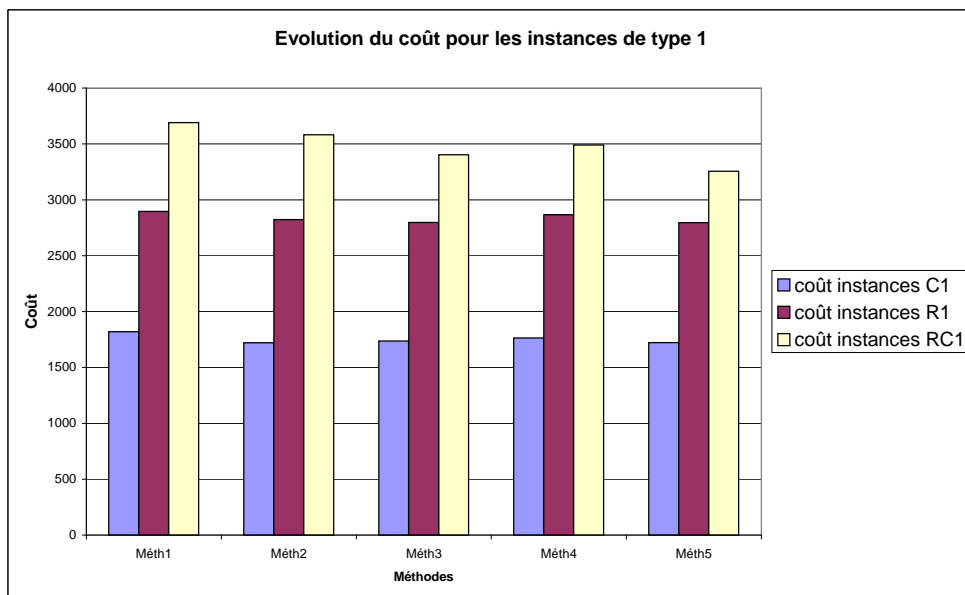


FIG. 4.11 – Comparaison des méthodes en termes de coût par type d'instances

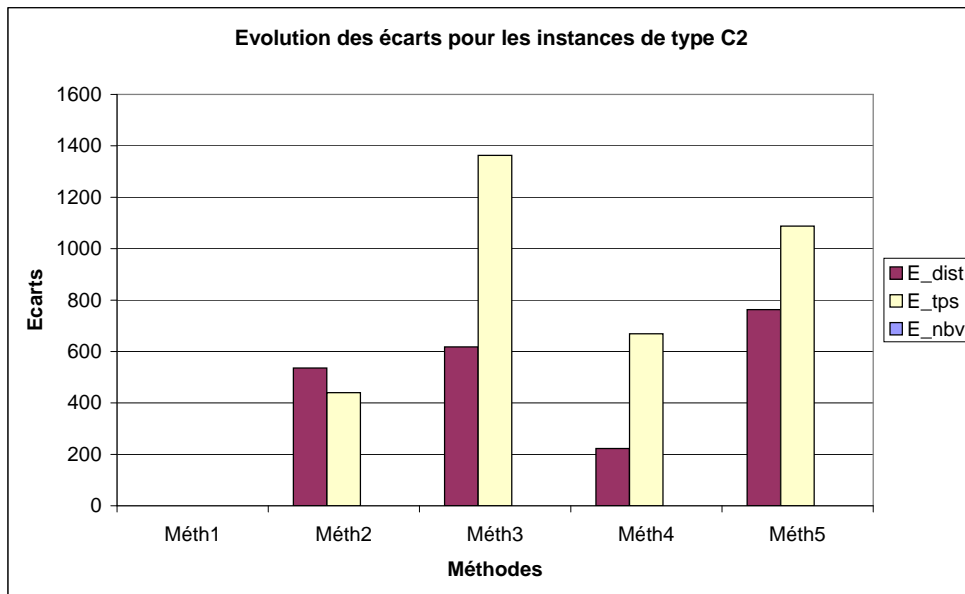
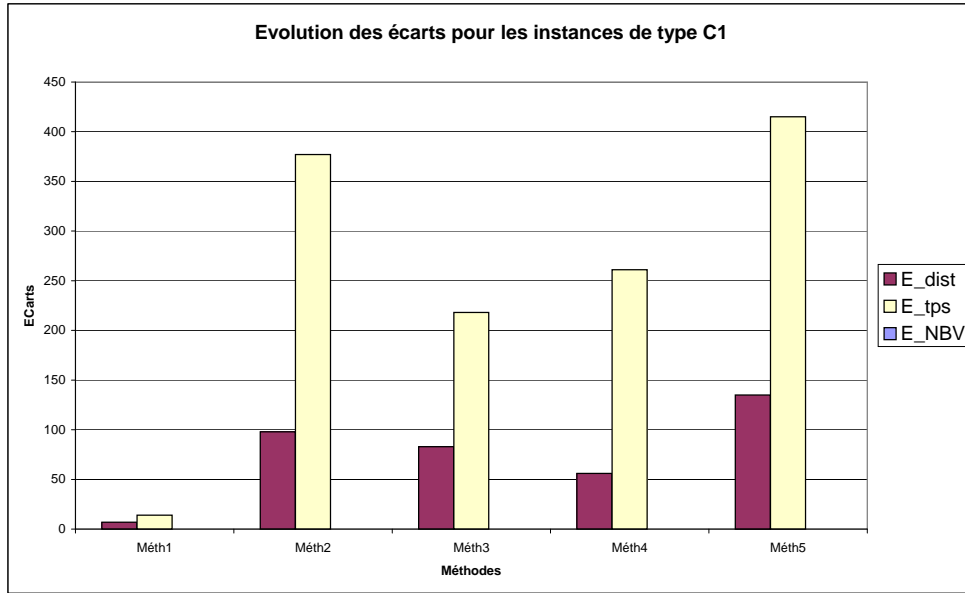


FIG. 4.12 – Comparaison des méthodes en termes de stabilité globale par type d'instances

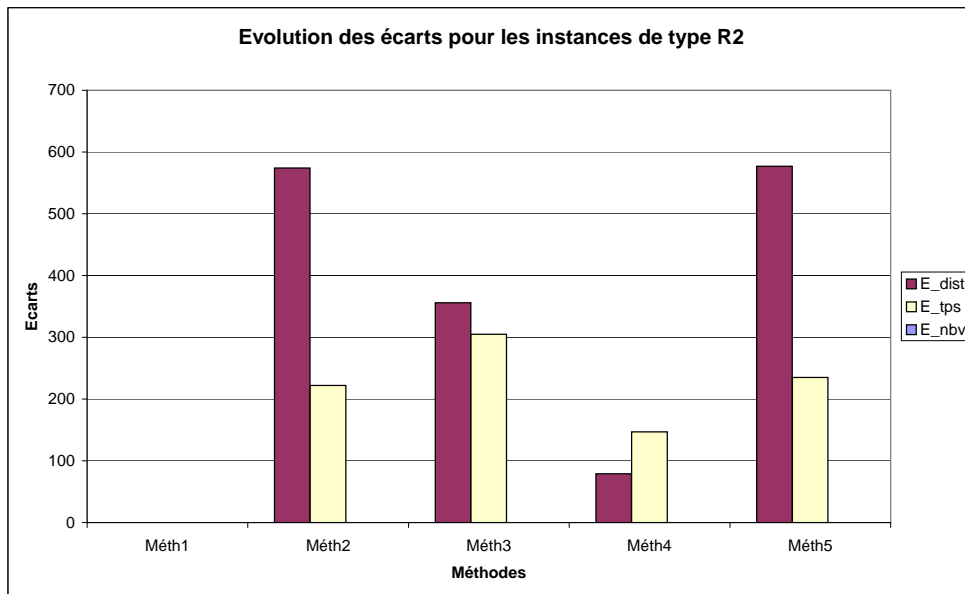
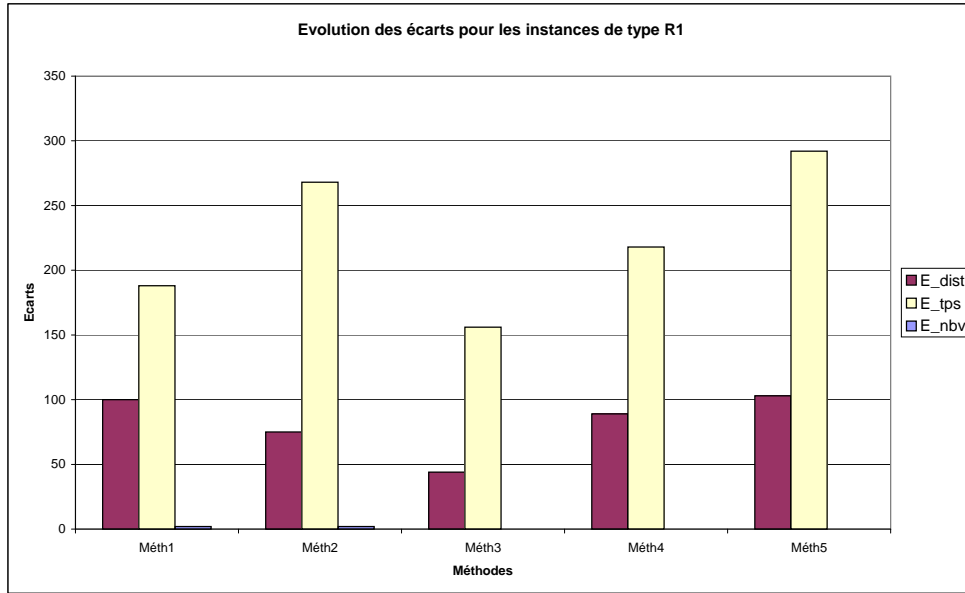


FIG. 4.13 – Comparaison des méthodes en termes de stabilité globale par type d'instances

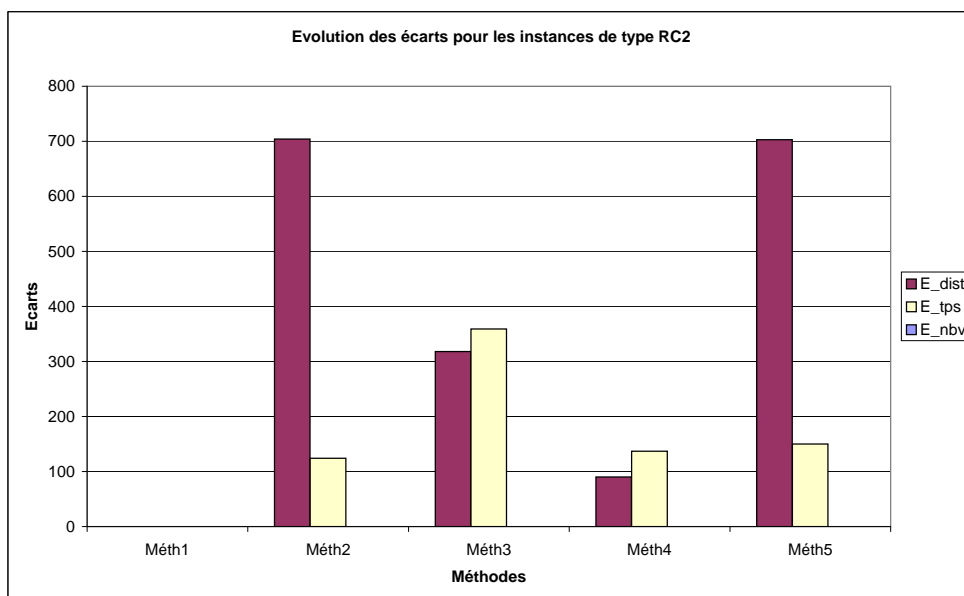
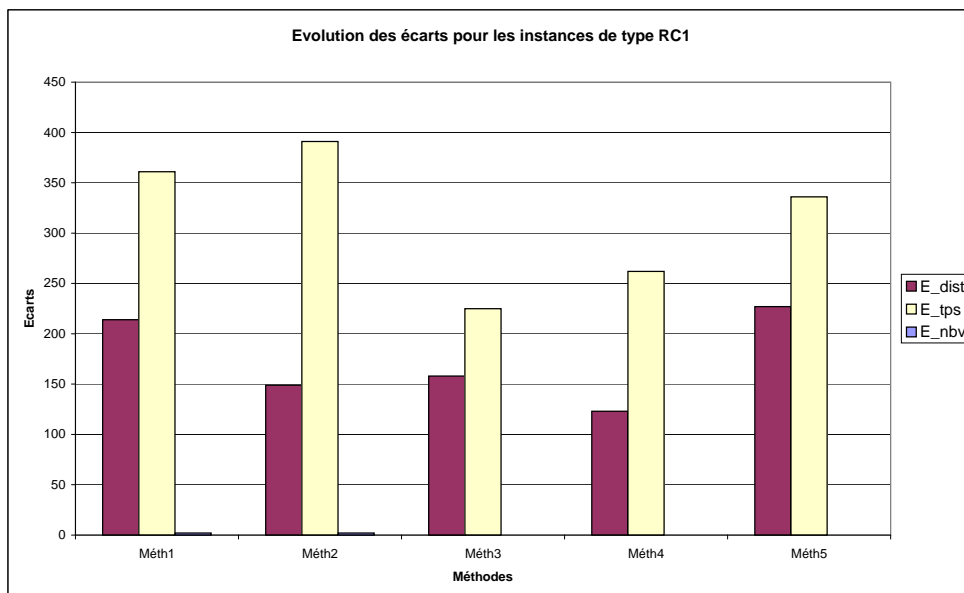


FIG. 4.14 – Comparaison des méthodes en termes de stabilité globale par type d'instances

que les méthodes A3 et A4 donnent chacune un meilleur résultat sur les deux autres indicateurs. Pour les instances R1 et RC1, un véhicule est ajouté par la méthode 1, ce qui implique que les résultats pour les méthodes A1 et A2 soient mauvais sur cet objectif.

- Pour les instances de type 2, les capacités des véhicules étant très importantes par rapport aux demandes des clients, il n'est pas nécessaire de transférer des nœud ni d'ajouter des véhicules, on se retrouve donc dans la même situation que pour l'instance C1, les écarts pour la méthode A1 sont égaux à zéro. En effet, les quantités acceptables par circuits sont plus importantes que pour les instances de type 1, il est donc difficile d'avoir un circuit en surcharge. Nous pouvons remarquer l'efficacité de la méthode A4 sur toutes les instances de ce type. Les méthodes A2 et A5 sont elles les moins adaptées.

En résumé le fait d'ajouter ou non un véhicule influence les résultats. S'il n'y a pas d'ajout de véhicule la méthode A1 donne évidemment les meilleurs résultats, puis la méthode A4. Et s'il y a un ajout de véhicule, les méthodes A3 ou A4 sont performantes sur ces indicateurs.

Résultats en termes de stabilité du point de vue usagers

Les graphiques (4.15) présentent les résultats sur l'objectif du point de vue usagers, c'est-à-dire l'écart horaire.

Sur cet indicateur, les résultats sont homogènes à ceux sur l'objectif de stabilité globale en fonction de l'ajout ou non d'un véhicule.

- Pour les instances R1 et RC1, les méthodes A3 ou A4 donnent les meilleurs résultats puis la méthode A1, la méthode A2 et enfin la méthode A5. Pour les instances C1, la méthode A1 donne est meilleure.
- Pour les instances de type 2, la méthode A1 donne les meilleurs résultats, puis la méthode A4. Ensuite arrive les méthodes A3 et A5.

Nous pouvons faire la même distinction que pour l'objectif de stabilité globale. Selon l'ajout ou non de véhicule les performances des méthodes varient. Nous pouvons tout de même observer que la méthode A4 donne sur l'ensemble des instances de bons résultats, même lorsqu'un véhicule est ajouté.

Résultats en termes de stabilité du point de vue employés

Les graphiques (4.16, 4.17 et 4.18) répertorient les résultats obtenus sur l'objectif de stabilité point de vue employés.

Sur l'ensemble des instances les meilleurs résultats sont donnés par la méthode A1. Puis nous pouvons constater les bonnes performances des méthodes A2 et A4 sur l'ensemble des instances sauf pour les instances C1 (pour lesquels la méthode A3 donne de bons résultats tandis qu'elle donne en général de mauvais résultats pour les autres instances) et pour les instances R2, où A2 donne de mauvais résultats.

Nous pouvons remarquer que pour cet objectif la méthode A1 est performante sur l'ensemble des instances et que la méthode A4 donne également de bons résultats sur l'ensemble des instances.

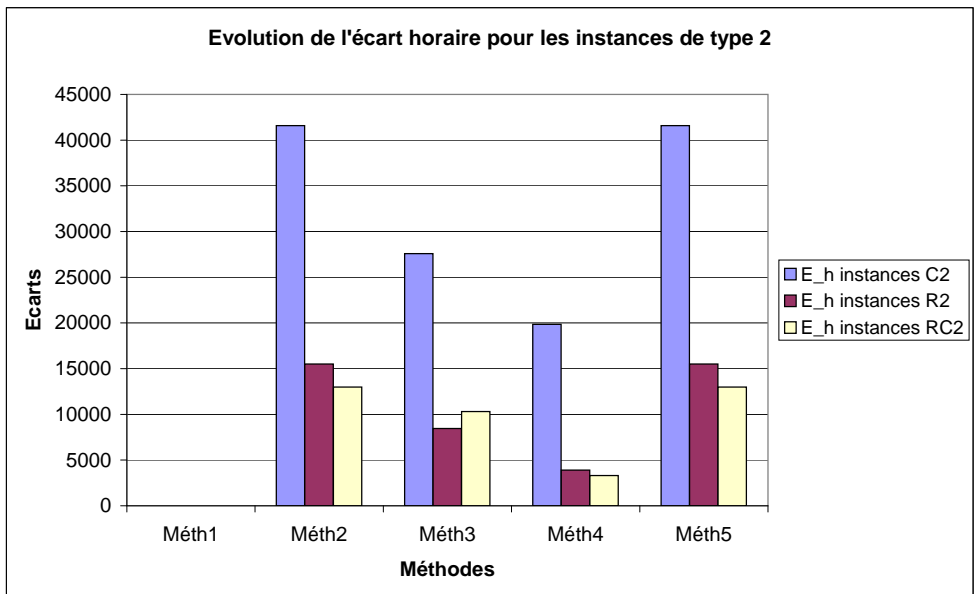
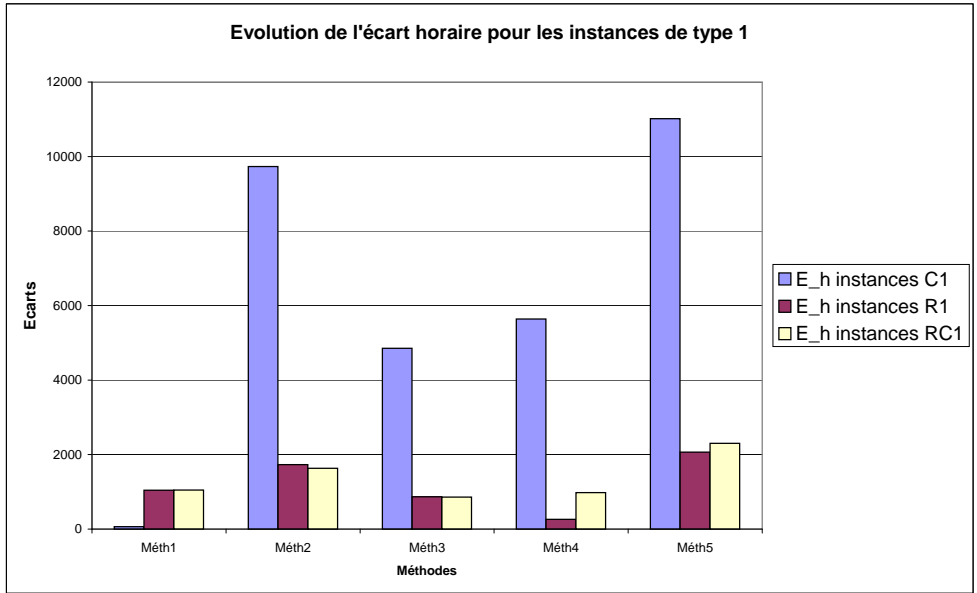


FIG. 4.15 – Comparaison des méthodes en termes de stabilité “usagers” par type d’instances

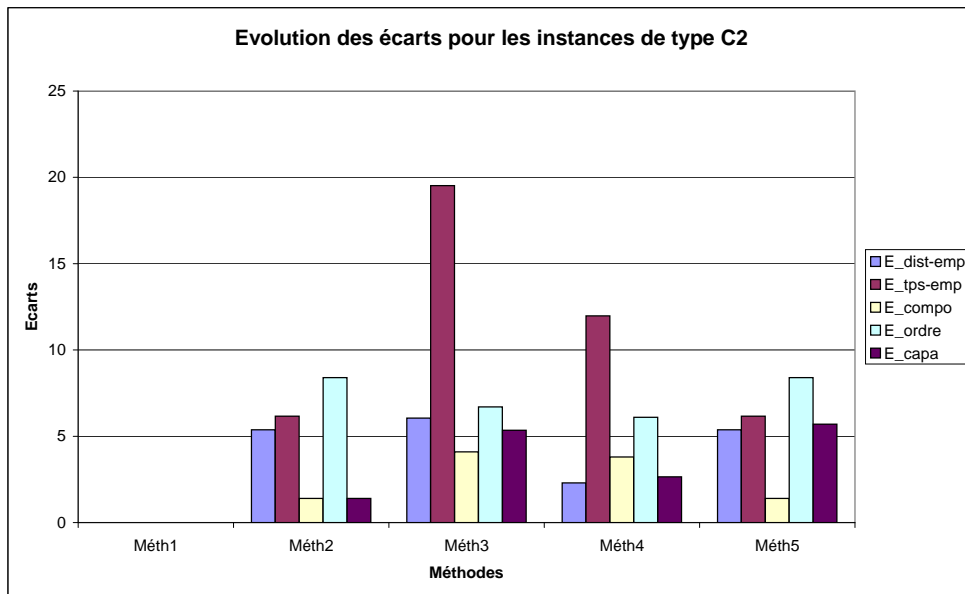
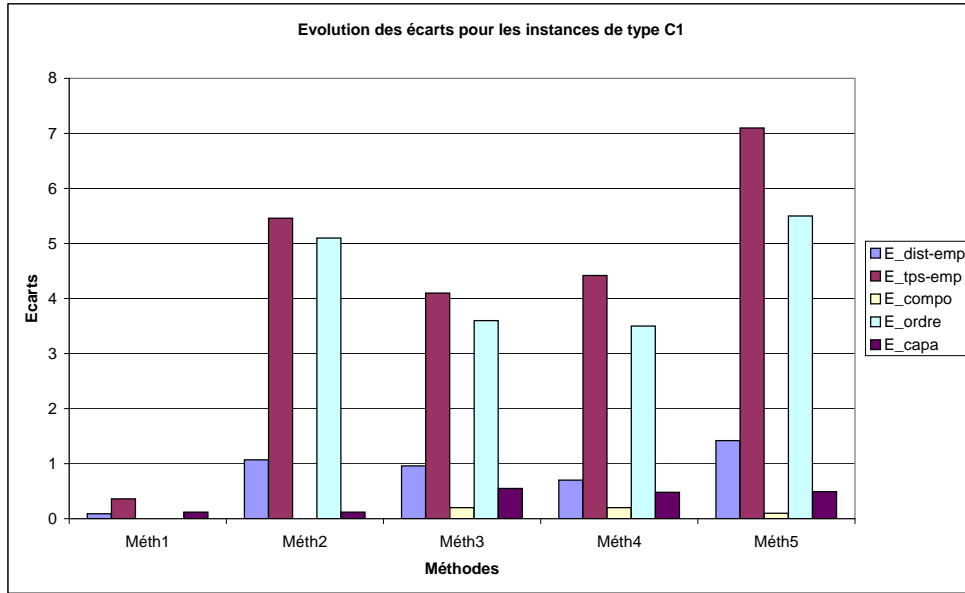


FIG. 4.16 – Comparaison des méthodes en termes de stabilité “employés” par type d’instances

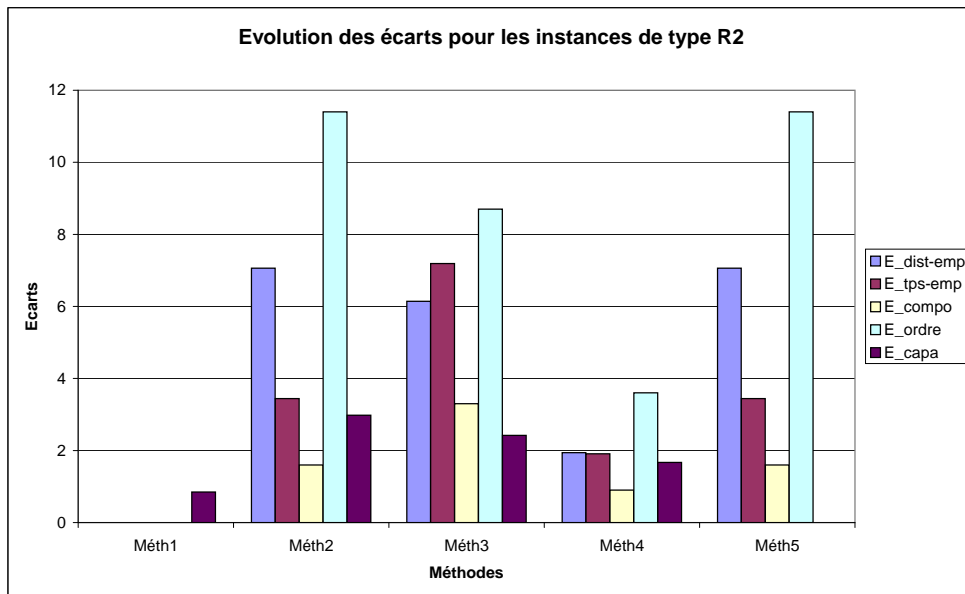
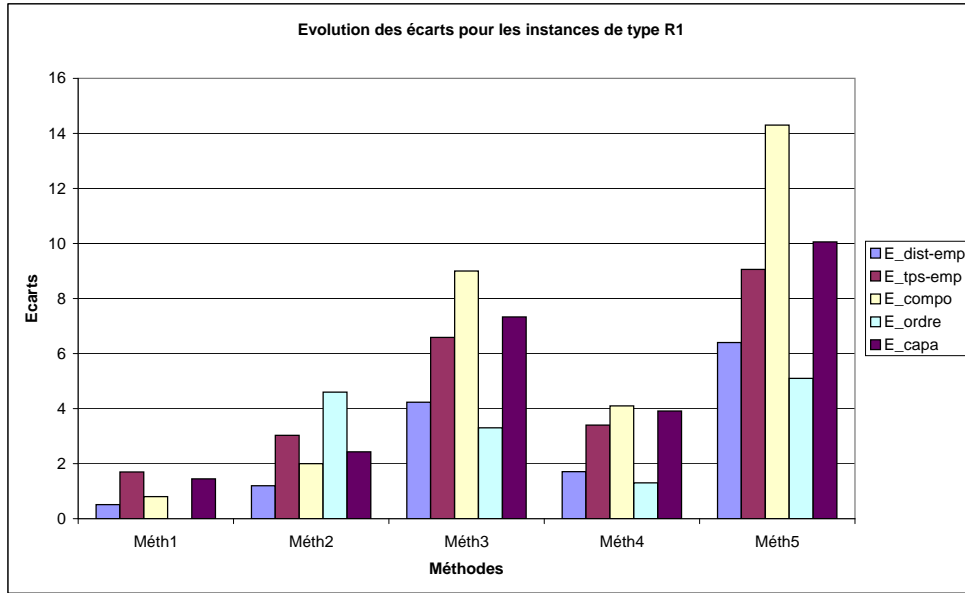


FIG. 4.17 – Comparaison des méthodes en termes de stabilité “employée” par type d’instances

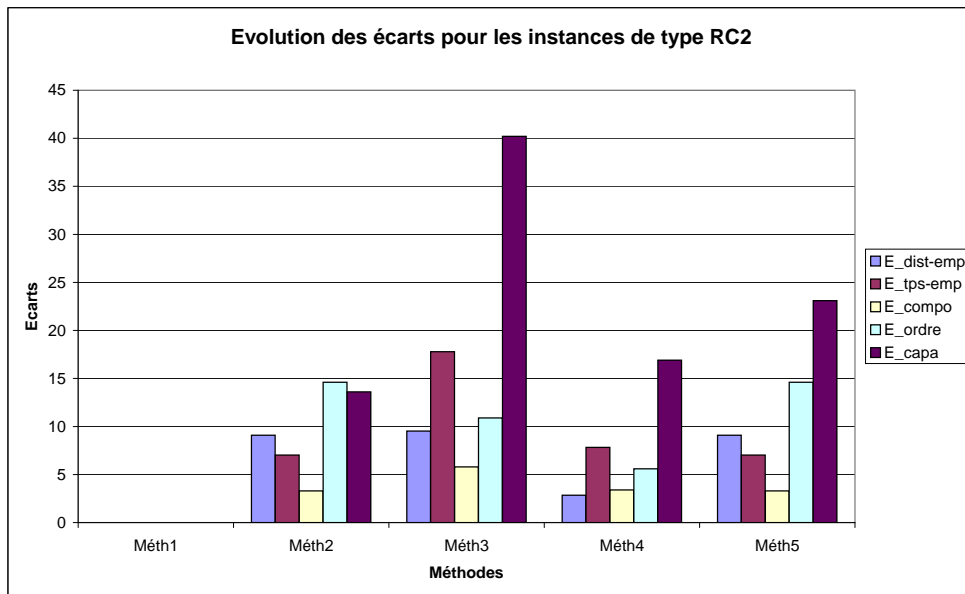
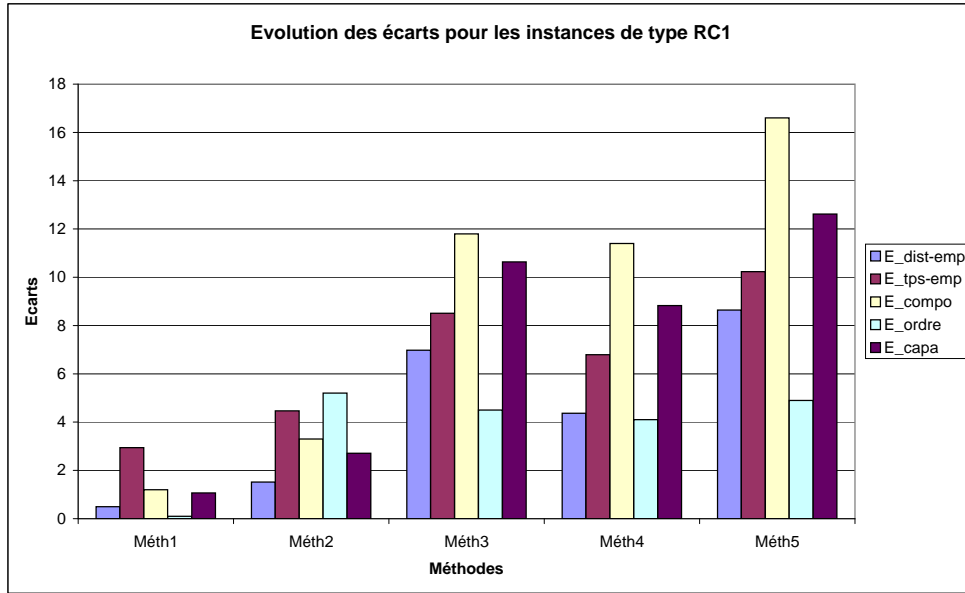


FIG. 4.18 – Comparaison des méthodes en termes de stabilité “employés” par type d’instances

Conclusion

En conclusion, comme nous pouvions nous y attendre la méthode A1 est très performante sur l'objectif de stabilité, de plus elle modifie peu ou pas les tournées déjà en place si celles-ci n'ont pas besoin de véhicules supplémentaires. La méthode A5 est quant-à-elle performante sur l'objectif de coût, mais donne en général les moins bons résultats sur l'objectif de stabilité. Pour ce qui est des autres méthodes, la méthode A4 donne en général un bon compromis entre l'objectif de coût et celui de stabilité. La méthode A2 tend vers la méthode A5, ceci est dû au fait que la méthode 1 modifie très faiblement les tournées lorsqu'aucun véhicule n'est ajouté et qu'on se retrouve donc dans la même situation que la méthode A5. Nous avons pu confirmer les résultats espérés lors de l'élaboration des différentes méthodes.

4.5 Discussion sur l'évolution continue des tournées

Dans ce chapitre nous avons étudié les évolutions de manière locale. C'est-à-dire que nous n'avons pris en compte qu'une seule évolution à la fois : apparition de nouveaux nœuds ou variation de quantité à collecter. Bien évidemment ces deux évolutions peuvent apparaître simultanément et complexifier d'autant plus l'obtention d'une solution de collecte proche de la solution courante.

Que peut-il se passer si les tournées évoluent continuellement selon une méthode qui maintient une bonne stabilité par rapport à la solution courante ?

D'après les résultats obtenus, nous voyons que les deux objectifs sont contradictoires : une méthode qui permet de trouver une solution avec un bon coût réorganise en général les tournées et donc offre une mauvaise stabilité et vice et versa. Ceci implique que lorsqu'on applique successivement les méthodes proposées ci-dessus, les tournées risquent de s'éloigner de plus en plus de celles de coût minimal (celle from "scratch"). Nous pouvons donc imaginer que les méthodes intermédiaires (les méthodes heuristiques suivis d'amélioration et la métaheuristique bi-objectifs) et notamment la métaheuristique bi-objectif, vont trouver un compromis entre ces deux extrêmes. Lors d'une itération nous obtenons un bon compromis entre le coût et la stabilité. Mais à force d'itération la métaheuristique bi-objectifs va devoir chercher des solutions de compromis de plus en plus éloignées de la solution de reconstruction complète de solution. Il serait donc intéressant de la comparer à chaque itération à celle qui reconstruit complètement les tournées en termes de coût. Plus il y a d'itérations, plus le coût sera important pour les méthodes proposées, tandis que la méthode de reconstruction complète continuera à conserver un bon objectif de coût. Par ailleurs ces évolutions peuvent avoir un impact sur la stabilité hebdomadaire des tournées étudiée au chapitre précédent.

La question est alors de savoir à partir de quand il s'avère nécessaire de réorganiser entièrement les tournées. En effet, la collectivité accepte de dégrader le coût pour maintenir une stabilité entre les tournées jusqu'à un certain point. Si la différence de coût entre les solutions est vraiment trop importante et que la construction de la solution initiale date depuis un certain temps (depuis plusieurs années), elle est prête à réorganiser complètement pour maintenir par la suite une solution relativement stable

durant une nouvelle “longue” période. Il est donc nécessaire de définir des indicateurs de reconstruction complète : une période d’évolution des informations relativement “longue” (plusieurs années) sur laquelle une stabilité doit être maintenue et un objectif de coût qui doit rester correct. Pour quantifier cela, nous n’avons pas pu faire d’expérimentations, mais nous pensons que la métaheuristique bi-objective va être un bon indicateur, notamment sur l’objectif de coût lorsque celui-ci sera comparé à celui obtenu par la méthode de reconstruction complète des tournées. A partir du moment où le coût sera fortement dégradé, en définissant par exemple un écart maximal entre le coût engendré par la métaheuristique et celui engendré par la méthode de reconstruction, il est peut être temps d’envisager une “refonte” des tournées. Il serait intéressant à ce niveau d’utiliser les notions de “regrets” pour qualifier les solutions. Ces regrets expriment le fait que si on ne choisit pas une solution, cela va nous impliquer un surplus sur le coût quantifiable. A partir du moment où les regrets deviennent trop importants par rapport au gain en stabilité que les solutions procurent, il serait peut être temps de reconstruire la solution entièrement.

4.6 Conclusion

Dans ce chapitre nous avons présenté une étude de l’évolution de certaines informations d’un *VRPTW*. Les évolutions sont deux types : des évolutions ajoutant de nouvelles variables (nouveaux points de collecte), et des évolutions sur les valeurs des données (quantité à collecter). Les méthodes heuristiques et métaheuristiques proposées pour prendre en compte ces deux évolutions sont similaires. Une première étape consiste à intégrer les nouvelles données dans les tournées existantes puis différentes étapes de post-optimisation guidées par certains objectifs. Pour comparer ces méthodes, nous considérons une méthode de reconstruction complète. L’ensemble des résultats présentés nous permettent de faire ressortir les avantages et les inconvénients de chacune des méthodes. Les premières méthodes proposées pour chaque évolution sont très bonnes en termes de stabilité, mais mauvaises en termes de coût et inversement pour les méthodes de post-optimisation guidées par le coût.

Malgré cela, certaines méthodes semblent donner des compromis intéressants, notamment la méthode utilisant une métaheuristique multi-objective donne des résultats de compromis, elle est performante sur chacun des objectifs. La difficulté ici réside dans la mesure de compromis a priori. Il faut fixer des coefficients pour les différents indicateurs et pour cela il faut connaître par avance les besoins du décideur. Il serait donc intéressant de proposer, comme dans le chapitre précédent les Front de Pareto pour avoir un ensemble de solution. Ceci n’a pas été fait dans ce chapitre, car nous nous sommes plus focalisés sur la comparaison de différentes méthodes d’évolution des tournées et donc la nécessité d’obtenir une solution par méthode.

Dans ce chapitre, chaque type d’évolution est traité séparément, or ces deux problématiques peuvent être couplées. Nous pouvons donc, de part les structures similaires pour les méthodes, agréger les méthodes pour n’en former plus qu’une seule qui traite de l’ensemble de la problématique.

Ces évolutions peuvent également être réitérées sur un horizon de temps plus important. En effet, les évolutions considérées sont régulières dans le temps, nous pouvons nous demander si au bout de

4.6. Conclusion

quelques années, par exemple, il ne serait pas plus judicieux (en terme de coût) de tout reconstruire plutôt que de faire évoluer la solution en place qui semble t-il perd en coût à chaque itération.

Conclusion et perspectives

Dans cette thèse, nous avons étudié le problème particulier de la collecte des déchets en se basant sur l'organisation effective d'une collectivité locale située proche de Toulouse (la CAM). La modélisation utilisée se base sur le modèle très connu qu'est celui des tournées de véhicules (Vehicle Routing Problem) avec collecte sur nœuds. Comme cité dans ce mémoire, ce problème particulier a déjà fait l'objet de nombreuses études souvent issues de cas pratiques. Ces études portent sur des problèmes avec leur spécificités tant au niveau de la géographie des lieux qu'aux aspects sociaux-techniques mis en jeu. Dans ce mémoire, nous avons étudié plus précisément deux problèmes émergents dus aux particularités et aux besoins de la collectivité partenaire que nous avons nommés problème d'organisation hebdomadaire des collectes et problème d'organisation dynamique des tournées. Le but de la collectivité est de proposer des tournées de collecte non seulement économique mais qui restent stables les unes par rapport aux autres. Les modifications d'un ensemble de tournée à un autre sont modélisées au sein d'un objectif appelé objectif de stabilité. Nous avons proposé une formulation de cet objectif en le divisant en trois parties caractérisant chacune un aspect de la stabilité : de manière globale, selon les usagers et selon les employés et cet objectif de stabilité caractérise l'éloignement existant entre deux solutions (la solution de départ et la nouvelle solution).

Le premier problème étudié concerne la construction à proprement parler des tournées sur une semaine. En effet, sur le territoire de la CAM une partie des points de collecte est ramassée deux fois par semaine (en début et en fin) et une autre une seule fois par semaine (en début). Le souci est de garder des tournées relativement "similaires" en début et en fin de semaine pour éviter de proposer aux usagers et aux employés des modifications trop importantes.

Le second problème étudié est celui de l'évolution des informations au sein de cette collectivité. En effet, non seulement son nombre d'habitation ne cesse d'augmenter (environ 2% par an), mais son taux de collecte par bac augmente lui aussi (environ de 2% par an et par habitant). Le but de cette seconde étude est de permettre d'intégrer ces nouvelles données sans trop perturber la solution courante.

Nous avons donc proposé d'étudier ces problèmes en tenant compte non seulement de l'objectif de coût mais également de celui de stabilité.

Nous avons, pour chacun des problèmes considérés, proposé différentes méthodes de conception des tournées à partir de tournées référentes, qui sont soit des méthodes heuristiques soit des méta-heuristiques prenant en compte un ou plusieurs objectifs.

Nous avons pu évaluer les avantages et les inconvénients inhérents à chacune de ces méthodes en

les testant sur des instances de la littérature.

L'étude proposée a montré les avantages et les inconvénients d'utiliser des méthodes heuristiques par rapport à des méthodes métaheuristiques. Les méthodes heuristiques sont des méthodes plus rapides, et qui s'avèrent de part leur conception, très efficaces en terme de stabilité, tandis que les métaheuristiques mono-objectifs sont des méthodes plus longues, moins bonnes sur le plan de la stabilité mais qui proposent de bonnes solutions en terme de coût. Enfin les métaheuristiques bi-objectifs proposent de bons compromis entre l'objectif de coût et de stabilité mais nécessitent un prétraitement pour fixer à priori des valeurs sur les indicateurs de stabilité. Par la suite, le décideur à la charge de choisir la solution qui lui convient le plus.

Les développements que nous avons proposés dans cette thèse sont destinés à être intégrés au sein d'une application pour la CAM qui a pour but de guider les décideurs pour la construction des tournées de collecte de déchets. Il faut donc intégrer en même temps les deux aspects développés dans ce mémoire, non seulement l'aspect hebdomadaire mais également l'aspect dynamique. D'une part il s'agit de construire deux ensembles de tournées, en début et en fin de semaine, mais il faut d'autre part faire évoluer les ensembles de tournées en fonction des évolutions d'informations de collecte. Au sein de cette application, la mise en valeur des résultats est primordiale, en effet le côté utilisation et pratique doit être mis en avant pour plus de facilité de compréhension des résultats par le décideur. Nous avons donc réfléchi à la mise en place d'indicateurs sur les résultats. Ils seront développés selon les objectifs utilisés et divisés comme dans l'étude :

- indicateur de coût ;
- indicateur de stabilité :
 - de manière globale ;
 - selon le point de vue des usagers ;
 - selon le point de vue des employés.

Chacun de ces indicateurs correspond à une agrégation des éléments le qualifiant. Par exemple le coût est somme des distances totales parcourues sur la semaine, celui concernant la stabilité des employés est l'agrégation de différents éléments qualifiant cette partie. Les indicateurs suggérés sont non seulement développés d'une solution s_t à une solution s_{t+1} , mais également d'une solution référente s_0 à une solution s' ayant subie plusieurs modifications (plusieurs évolutions d'informations). Ceci permet de garder un historique des modifications et de garder en mémoire chaque modification effectuée sur la solution référente.

Ces indicateurs doivent être pertinents et visuels mais doivent pouvoir s'adapter à d'autres types de collecte que celle de la CAM pour former une plateforme générique.

Pour ce qui est des méthodes utilisées, nous avons comparé des heuristiques et métaheuristiques. Les heuristiques utilisées sont des heuristiques dédiées aux problèmes évoqués dans les chapitres 3 et 4. Elles sont limitées en nombre et ont été élaborées dans le but de ne pas trop s'éloigner de la solution courante. Il pourrait être intéressant de modifier légèrement ces heuristiques et de proposer

pour chacune différentes notions de voisinage ou différentes notions d'insertion de nœuds. Pour l'instant, nous utilisons une procédure simplifiée pour l'insertion de nœuds, il pourrait être intéressant pour une même méthode de comparer les résultats pour différentes forme d'insertion plus complexes et prenant en compte le couplage avec la notion de voisinage d'une tournée et d'une autre. On pourrait développer une méthode d'insertion permettant d'insérer un nœud dans une tournée qui semble la plus proche et gérant la surcharge éventuelle en transférant des nœuds dans une tournée voisine.

Les métaheuristiques utilisées sont des métaheuristiques classiques basées sur la notion de voisinage. Nous avons pu constater dans le chapitre sur l'état de l'art que les algorithmes évolutionnaires ont été très utilisés, notamment pour les problèmes multi-objectifs. Il serait donc intéressant de développer une métaheuristique basée sur ces mécanismes pour obtenir des solutions à nos problèmes. Nous imaginons donc intégrer ce type de méthodes afin de comparer ses résultats avec les résultats obtenus par les heuristiques et les métaheuristiques à voisinage proposées. Ces méthodes évolutionnaires semblent particulièrement adaptées à la proposition de solution réparties sur l'espace de recherche de part leur structure et le fait qu'elles utilisent des populations de solutions. Il serait également intéressant dans un deuxième temps d'utiliser les points forts de chaque métaheuristique pour en former une seule. La diversification de population à l'aide de la méthode évolutionnaire et l'intensification de recherche dans un voisinage particulier à l'aide de la méthode tabou, par exemple.

Par la suite, nous devons pouvoir faire évoluer les solutions au cours du temps, comme évoqué au chapitre 4. Il semblerait intéressant de proposer des méthodes qui minimisent les écarts tout en gardant un coût correct, et qui indiqueraient qu'il est nécessaire de reconstruire les tournées plutôt que de les adapter. Pour cela, nous avons pensé à développer des indicateurs décrivant la notion de "regrets". Ces indicateurs permettraient de qualifier une solution non pas sur les écarts d'une solution à une autre, mais sur ce qu'elle provoquerait si elle n'est pas choisie par l'utilisateur. Ces regrets peuvent concerner chacun des objectifs, celui de coût et celui de stabilité et peuvent être calculés par rapport à des solutions "extrêmes" sur le Front Pareto. Ces solutions extrêmes sont celles donnant le meilleur coût et celles fournissant la meilleure stabilité sur le Front Pareto. Par exemple, concernant le coût, cet indicateur consisterait à calculer la différence entre le coût engendré par une méthode λ et celui engendré par la méthode de reconstruction. Pour celui concernant la stabilité, ce serait par exemple la comparaison avec la méthode I1 ou A1 dans le cas d'évolution des informations. En résumé, on fixe une borne maximale pour chacun des objectifs et on la compare avec le résultat obtenu sur chaque objectif pour chaque méthode. Le choix de la méthode se fait ici selon ce que le décideur va le plus regretter de détériorer.

Une autre prospective à ce travail, est de prendre en compte de nouvelles évolutions des informations. En effet, la collectivité partenaire est située proche de Toulouse, ce qui lui donne certaines particularités comme une baisse de population pendant la période estivale. Une proposition de gestion saisonnière semble indiquée pour ce type de particularité, et permettrait à la collectivité de gérer au mieux les périodes de vacances pour ses employés. Cet aspect combine à la fois la stabilité hebdomadaire et la stabilité dynamique et peut s'adapter à différentes type de collectivité. Cette problématique est assez

4.6. Conclusion

générique, il existe différents endroits qui n'ont pas la même population selon les périodes de l'année, et ceci peut s'avérer intéressant de proposer une gestion différente des tournées de collecte des déchets selon la période de l'année.

Bibliographie

- [1] <http://neo.lcc.uma.es/radi-aeb/webvrp/>.
- [2] ADEME. Gestion des déchets ménagers et assimilés : transport et logistique. 1998.
- [3] A.S. Alfa, S.S. Heragu, and M. Chen. A 3-opt based simulated annealing algorithm for vehicle routing problems. *Computers and Industrial Engineering*, 21 :635–639, 1991.
- [4] K. Altinkemer and B. Gavish. Parallel savings based heuristic for the delivery problem. *Operations Research*, 39 :456–469, 1991.
- [5] S.K. Amponsah and S. Salhi. The investigation of a class of capacited arc routing problems : the collection of garbage in developing countries. *Waste management*, 24(7) :711–721, 2004.
- [6] E. Angelelli and M.G. Speranza. The application of a vehicle routing model to a waste-collection problem : two cases studies. *Journal of the Operational Research Society*, 53(9) :944–952, 2002.
- [7] P. Baise. Tournées de véhicules d'une société coopérative : algorithmes séquentiels et parallèles. *stage au laboratoire PRiSM*, 1996.
- [8] S. Baptista, R.C. Oliveira, and E. Zuquete. A period vehicle routing case study. *European Journal of Operational Research*, 132 :220–229, 2002.
- [9] B. Baran and M. Schaerer. A multiobjective ant colony system for vehicle routing problem with time windows. *Fom Proceeding : Applied Informatics*, 378, 2003.
- [10] J. Bautista, E. Fernández, and J. Pereira. Solving an urban waste collection problem using ants heuristics. *Computers and Operations Research*, 35 :3020–3033, 2008.
- [11] J. Bautista and J. Pereira. Ant algorithms for urban waste collection routing. *Lecture Notes in Computer Science*, (3172) :302–309, 2004.
- [12] J.E. Beasley. Route-first cluster-second methods for vehicle routing. *Omega*, 11(4) :403–408, 1983.
- [13] A. Beham. Parallel tabu search and the multiobjective capacitated vehicle routing problem with soft time windows. *Lecture Notes in Computer Science*, 4739/2007 :829–836, 2007.
- [14] J.M. Belenguer, E. Benavent, P. Lacomme, and C. Prins. Lower and upper bounds for the mixed capacited arc routing problem. *Computers and Operations Research*, 33 :3363–3383, 2006.
- [15] E.J. Beltrami and L.D. Bodin. Networks and vehicle routing for municipal waste collection. *Networks*, 4 :65–94, 1974.

- [16] J. Berger and M. Barkaoui. A hybrid genetic algorithm for the capacited vehicle routing problem. *GECCO 2003 LNCS 2723*, pages 646–656, 2003.
- [17] J. Berger and M. Barkaoui. A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. *Computers and Operations Research*, 31 :2037–2053, 2004.
- [18] L.D. Bodin, B.L. Golden, A.A. Assad, and M.O. Ball. Routing and scheduling of vehicles and crews : the state of the art. *Computers and Operations Research*, 10(2) :63–211, 1983.
- [19] D. Bommessity, M. Dessouky, and L. Jacobs. Scheduling collection of recyclable material at northern illinois university campus using a two-phase algorithm. *Computers and Industrial Engineering*, 35 :435–438, 1998.
- [20] A. Van Breedam. An analysis of the behavior of heuristics for the vehicle routing problem for a selection of problems with vehicle-related, customer-related, and time-related constraints. *Ph.D. dissertation, University of Antwerp*, 1994.
- [21] A.V. Breedam. Improvement heuristics for the vehicle routing problem based on simulated annealing. *European Journal of operational Research*, 86 :480–490, 1995.
- [22] O. Bräysy. A reactive variable neighborhood search for the vehicle routing problem with time windows. *INFORMS Journal on Computing*, 15(4) :347–368, 2003.
- [23] O. Bräysy and M. Gendreau. Tabu search heuristics for vehicle routing problem with time windows. *Sociedad de Estadística e investigación Operativa, Top*, 10(2) :211–237, 2002.
- [24] O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, part ii : Metaheuristics. *Transportation Science*, 39 :119–139, 2005b.
- [25] B. Bullnheimer, R.F. Hartl, and C. Strauss. An improved ant system algorithm for the vehicle routing problem. *Annals of Operations Research*, 89 :319–328, 1999.
- [26] V. Cerny. Thermodynamical approach to the traveling salesman problem : an efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1) :41–51, 1985.
- [27] N.B. Chang, H.Y. Lu, and Y.L. Wei. Gis technology for vehicle routing and scheduling in solid waste collection systems. *Journal of Environmental Engineering*, 123 :901–910, 1997.
- [28] C.H. Chen, C.J. Ting, and P.C. Chang. Applying a hybrid ant colony system to the vehicle routing problem. *ICCSA 2005, LNCS 3483*, pages 417–426, 2005.
- [29] W.C. Chiang and R.A. Rusell. Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*, 63 :3–27, 1996.
- [30] N. Christofides, A. Mingozzi, and P. Toth. The vehicle routing problem. *Combinatorial Optimization*, Wiley, Chister :315–338, 1979.
- [31] G. Clarke and J.W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12 :568–581, 1964.
- [32] C.A. Coello. An updated survey og g.g. based multiobjective optimization techniques. *Rapport technique Lania-RD-98-08, Xalapa, Veracruz, Mexico*, 1998.

- [33] Y. Collette and P. Siarry. *Optimisation multiobjectif*. Eyrolles edition, 2002.
- [34] A. Coloni. Distributed optimization by ant colonies. *proceedings of the first European Conference on Artificial Life (ECAL 91)*, pages 134–142, 1992.
- [35] J.F. Cordeau and G. Laporte. The dial-a-ride problem (darp) : Variants, modeling issues and algorithms. *4OR : A Quarterly Journal of Operations Research*, 1(2) :89–101, 2002.
- [36] J.F. Cordeau, G. Laporte, and A. Mercier. A unified tabu search heuristic for the vehicle routing problems with time windows. *Journal Of the Operational Research Society*, 52 :928–936, 2001.
- [37] G.B. Dantzig and J.H. Ramser. The truck dispatching problem. *Operations Research, Management Sciences*, 6(1) :80–91, 1959.
- [38] C. Darwin. On the origin of species. *John Murray, Londres, Angleterre*, 1859.
- [39] Agence de l'Environnement et de la Maîtrise de l'Energie. www.ademe.fr.
- [40] G. Desaulniers, J. Desrosiers, A. Erdmann, M.M. Solomon, and F. Soumis. Vrp with pickup and delivery. *The Vehicle Routing Problem*, Philadelphia,PA : SIAM :225–242, 2002.
- [41] M. Desrochers and T. W. Verhoog. A matching based savings algorithm for the vehicle routing problem. *Working paper G-89-04, GERAD, Université de Montréal*, 56 :954–961, 1989.
- [42] C. Dhaenens-Flipo. Optimisation combinatoire multi-onjectif : Apport des méthodes coopératives et contribution à l'extraction de connaissances. *Habilitation à diriger des recherches de l'Université des Sciences et Technologies de Lille*, 2005.
- [43] H. Dickhoff. A typology of cutting and packing problems. *European Journal of Operational Research*, 44 :145–159, 1990.
- [44] M. Dorigo. Learning and natural algorithms. *Doctorat, Dipartimento di Elettronica e Informatica, Politecnico di Milano, Italie*, 1992.
- [45] M. Dorigo, V. Maniezzo, and A. Coloni. Positive feedback as a search strategy. rapport technique 910116. *Dipartimento di Elettronica e Informatica, Politecnico di Milano, Italie*, 1991.
- [46] M. Dorigo, V. Maniezzo, and A. Coloni. The ant system : optimization by a colony of cooperating agents. *IEEE Transactions on System, Man, and Cybernetics Part B : Cybernetics*, 26 :29–41, 1996.
- [47] M. Dror. Split delivery routing. *Naval Research Logistics*, 37 :383–402, 1990.
- [48] M. Dror. *Arc Routing : Theory, Solutions and Applications*. Kluwer Academic Publishers, 2000.
- [49] F.Y. Edgeworth. Mathematical physics. *P. Keagan, London*, 1881.
- [50] M. Ehrgott. Multicriteria optimization. *Lectures Notes in Economics and Mathematical Systems*, 491, 2000.
- [51] M. Ehrgott and X. Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spektrum*, 22 :425–460, 2000.

- [52] D.D. Eisenstein and A.V. Iyer. Garbage collection in Chicago : a dynamic scheduling model. *Management Science*, 43 :922–933, 1997.
- [53] N. El-Sherbeny. Resolution of a vehicle routing problem with multi-objective simulated annealing method. *Thèse, Faculté Polytechnique de Mons, Belgique*, 2001.
- [54] M.L. Fisher. Optimal solution of vehicle routing problems using minimum k-trees. *Operations Research*, 42(4) :626–642, 1994.
- [55] M.L. Fisher and R. Jaikumar. A decomposition algorithm for large-scale vehicle routing. *Working Paper Dept. of Decision Sciences, University of Pennsylvania, Philadelphia*, 1978.
- [56] M.L. Fisher and R. Jaikumar. A generalized assignment heuristic for vehicle routing. *Networks*, 11 :109–124, 1981.
- [57] P. Francis, K. Smilowitz, and M. Tzur. The period vehicle routing problem with service choice. *Transportation Science*, 40 :439–454, 2006.
- [58] L.M. Gambardella, E. Taillard, and G. Agazzi. A multiple ant colony system for vehicle routing problems with time windows. In : *Corne D. Dorigo M, Glover F, editors. New ideas in optimization. New York : McGraw-Hill*, chapitre 5 :63–76, 1999.
- [59] T.J. Gaskell. Bases for vehicle fleet scheduling. *Operational Research Quarterly*, 18 :281–295, 1967.
- [60] M. Gendreau, A. Hertz, and G. Laporte. A tabu search heuristic for the vehicle routing problem. *Management Science*, 40 :1276–1290, 1994.
- [61] M. Gendreau, G. Laporte, and J. Y. Potvin. Metaheuristics for the vehicle routing problem. *Les Cahiers du GERAD, G-98-52, Montreal, Canada*, 1998.
- [62] M. Gendreau, G. Laporte, and R. Séguin. New insertion and postoptimization procedure for the traveling salesman problem. *Transportation science*, 29(2) :143–155, 1992.
- [63] M. Gendreau, G. Laporte, and R. Séguin. Stochastic vehicle routing. *European Journal of Operational Research*, 88 :3–12, 1996.
- [64] B.E. Gillett and L.R. Miller. A heuristic algorithm for the vehicle dispatch problem. *Operations Research*, 22(2) :340–349, 1974.
- [65] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13 :533–549, 1986.
- [66] D.E. Goldberg. Genetic algorithms in search, optimization, and machine learning. *Addison-Wesley, reading, Massachusett*, 1989.
- [67] B. Golden, S. Raghavan, and E. Wasil. *The Vehicle Routing Problem*. Springer Editors, 2007.
- [68] B.L. Golden and A.A. Assad. Vehicle routing : Methods and studies. *Elsevier Science Publishers B.V., North-Holland*, 1988.
- [69] B.L. Golden, A.A. Assad, L. Levy, and F.G. Gheysens. The fleet size and mix vehicle routing problem. *Computers and Operations Research*, 11(1) :49–66, 1984.

- [70] A. Haghani and S. Jung. A dynamic vehicle routing problem with time-dependent travels times. *Computers and Operations Research*, 32 :2959–2986, 2005.
- [71] J.I. Van Hermet and J.A. La Poutré. Dynamic routing problems with fruitful regions : Models and evolutionary computation. *Parallel Problem Solving from Nature VIII*, pages 690–699, 2004.
- [72] H.Housroum. Une approche génétique pour la résolution du problème vrptw dynamique. *Thèse élaborée à l'Université d'Artois*, 2005.
- [73] J.H. Holland. Adaptation in natural and artificial system. *Ann Arbor, The University of Michigan Press*, 1975.
- [74] S. ichoua, M. Gendreau, and J.Y. Potvin. Exploiting knowledge about future demands for real-time vehicle dispatching. *Transportation Science*, 40 :211–215, 2006.
- [75] G.K. Janssens. Fleet size determination for waste oil collection in the province of antwerp. *Yugoslav Journal of Operational Research*, 3 :103–113, 1993.
- [76] N. Jozefowicz. Modélisation et résolution approchée de problèmes de tournées multi-objectif. *Thèse de l'Université des Sciences et Technologies de Lille*, 2004.
- [77] N. Jozefowicz, F. Semet, and E-G. Talbi. Multi-objective vehicle routing problems. *European Journal of operational Research*, 189 :293–309, 2008.
- [78] N. Jozefowicz, F. Semet, and E.G. Talbi. Enhancements of nsga ii and its application to the vehicule routing problem with route balancing. *Lectures Notes in Computer Science*, 3871 :131–142, 2006.
- [79] B.I. Kim, S. Kim, and S. Sahoo. Waste collection vehicle routing problem with time windows. *Computers and Operations Research*, 33 :3624–3642, 2006.
- [80] J.D. Kim, H.S. Choi, and D.H. Lee. Vehicle routing in a refuse collection system : a case study. *Computational Science and its Applications, ICCSA 2007*, pages 467–473, 2007.
- [81] G.A.P Kinderwater and M.W.P. Savelsbergh. Vehicle routing : Handling edge exchanges. *Asia-Pacific Journal of Operational research*, Chichester, UK :Wiley(E.H.L. Aarts and J.K. Lenstra, editors, Local Search in Combinatorial Optimization) :337–360, 1997.
- [82] K. Smith Korfmacher. Solid waste collection system in developing urban areas of south africa : an overview and case study. *Waste Management and Research*, 15 :477–494, 1997.
- [83] P.A. Koushki, U. Al-Duaij, and W. Al-Ghimlas. Collection and transportation cost of household solid waste in kuwait. *Waste management*, 24 :957–964, 2004.
- [84] S.O. Krumke, W.E. De Paepe, D. Poensgen, M. Lipmann, A. Marchetti-Spaccamela, and L. Stougie. On minimizing the maximum flow time in the online dial-a-ride problem. *Springer Berlin / Heidelberg*, 3879, 2006.
- [85] T. Kulcar. Optimizing solid waste collection in brussels. *European Journal of Operational Research*, 90 :71–77, 1996.

- [86] N. Labadi. Problèmes d'optimisation en tournées sur les arcs. *Thèse à l'université de technologie de Troyes*, 2003.
- [87] P. Lacomme, C. Prins, and W. Ramdane-Chérif. A genetic algorithm for the capacited arc routing problem and its extension. *Dans E.J.W. Boers at al. (eds.), Applications of evolutionary computing, Lecture Notes in Computer Sciences 2037, Springer*, pages 473–483, 2001.
- [88] P. Lacomme, C. Prins, and W. Ramdane-Chérif. Evolutionary algorithms for periodic arc routing problems. *European Journal of Operational Research*, 165(2) :535–553, 2005.
- [89] P. Lacomme, C. Prins, and M. Sevaux. A genetic algorithm for a bi-objective capacited arc routing problem. *Computers and Operations Research*, (33) :3473–3493, 2006.
- [90] G. Laporte and F.V. Louveaux. Solving stochastic routing problems with the integer l-shaped method. *In : Crainic, G., Laporte, G. (Eds.), Fleet Management and Logistics. Kluwer Academic Publishers, Boston*, pages 159–167, 1998.
- [91] G. Laporte and Y. Nobert. Exact algorithms for the vehicle routing problem. *Annals of Discrete Mathematics*, 31 :147–184, 1987.
- [92] G. Laporte and F. Semet. Classical heuristics for the vehicle routing problem. *Cahier du GERAD*, (G-98-54), 2005b.
- [93] A. Larsen. The dynamic vehicle routing problem. *LYNGBY, IMM-PhD-2000-73, Denmark*, 2000.
- [94] A. Larsen, O.B.G. Madsen, and M.M. Solomon. Classification of dynamic vehicle routing systems. *In V. Zeimpekis, G.M. Giaglis and C.D. Tarantilis and I. Minis, editors, Dynamic Fleet Management : Concept, Systems, Algorithms and Case Studies. Springer-Verlag*, 2007.
- [95] J.K. Lenstra and A.H.G. Rinnooy Kan. Complexity of vehicle routing and scheduling problem. *Networks*, 11 :221–227, 1981.
- [96] S. Lin. Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44 :2245–2269, 1965.
- [97] C. Liu, T.C. Chang, and L.F. Huang. Multi-objective heuristics for the vehicle routing problem. *International Journal of Operations Research*, 3(3) :173–181, 2006.
- [98] R. Álvarez Valdés, E. Benavent, V. Campos, A. Corberán, E. Mota, J. M. Tamarit, and V. Valls. A computerized system for urban garbage collection. *Topology*, 1 :89–105, 1993.
- [99] R. Minciardi, M. Paolucci, and E. Trasforini. A new procedure to plan routing and scheduling of vehicles for solid waste collection at a metropolitan scale. *Presented at Odysseys, Palermo, Italy*, 2003.
- [100] N. Mladenović and P. Hansen. Variable neighborhood search. *Computers and Operations Research*, 24 :1097–1100, 1997.
- [101] R.H. Mole and S.R.R. Jameson. A sequential route-building algorithm employing a generalized saving criterion. *Operational Research Quarterly*, 27(2) :503–511, 1976.

- [102] R. Montemanni, L.M. Gambardella, A.E. Rizzoli, and A.V. Donati. Ant colony system for a dynamic vehicle routing problem. *Journal of Combinatorial Optimization*, 10 :327–343, 2005.
- [103] M.C. Mourao and M.T. Almeida. Lower-bounding and heuristic methods for a refuse collection vehicle routing problem. *European Journal of Operational Research*, 121 :420–434, 2000.
- [104] M. Mourgaya and F. Vanderbeck. Problème de tournées de véhicules multipériodiques : classification et heuristique pour la planification tactique. *RAIRO Operations Research*, 40 :169–194, 2006.
- [105] T. Nuortio, J. Kytöjoki, J. Niska, and O. Bräysy. Improved route planning and scheduling of waste collection and transport. *Expert Systems with Applications*, 30 :223–232, 2006.
- [106] B.M. Ombuki-Berman, A. Runka, and F.T. Hanshar. Waste collection vehicle routing problem with time windows using multi-objective genetic algorithms. *Technical Report CS-07-04*, 2007.
- [107] H.L. Ong, T.N. Goh, and K.L. Poh. A computerised vehicle routing system for refuse collection. *Advances in Engineering Software*, 12 :54–58, 1990.
- [108] I. Or. Travelling salesman type combinatorial problems and their relation to the logistics of blood banking. *PhD Dissertation Department of industrial Engineering and Management Science. Northwestern University Evanston IL.*, 1976.
- [109] I.H. Osman. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41 :421–451, 1993.
- [110] I.H. Osman and G. Laporte. Metaheuristics : a bibliography. *Annals of Operations Research*, 63 :513–623, 1996.
- [111] M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33 :60–100, 1991.
- [112] V. Pareto. Cours d'économie politique. *Rouge, Lausanne*, 1896.
- [113] D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers and Operations Research*, 4638/2007 :187–191, 2007.
- [114] M. Polacek, K.F. Doerner, R.F. Hartl, and V. Maniezzo. A variable neighborhood search for the capacited arc routing problem with intermediate facilities. 2005.
- [115] J.Y. Potvin and J.M. Rousseau. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operations Research*, 66 :331–340, 1993.
- [116] C. Prins. A simple and ejective evolutionary algorithm for the vehicle routing problem. *Computer and Operations Research*, 31 :1985–2002, 2004.
- [117] J. Privé, J. Renaud, F. Boctor, and G. Laporte. Solving a vehicle-routing problem arising in soft-drink distribution. *Journal of the Operation Research Society*, 57 :1045–1052, 2006.
- [118] J. Privé, J. Renaud, F. Boctor, and G. Laporte. Solving a vehicle-routing problem arising in soft-drink distribution. *Journal of the Operation Research Society*, 57 :1045–1052, 2006.

- [119] Q. Jun, J. Wang, and B. J. Zheng. A hybrid multi-objective algorithm for dynamic vehicle routing problem. *Lecture Notes in Computer Science*, 5103/2008 :674–681, 2008.
- [120] W. Ramdane-Chérif. Problèmes d'optimisation en tournées sur les arcs. *Thèse soutenue à l'université de technologie de Troyes*, 2002.
- [121] W. Ramdane-Chérif. Evolutionary algorithms for capacitated arc routing problems with time windows. *12th IFAC Symposium on Information Control Problems in Manufacturing - INCOM*, 2006.
- [122] C. Rego and C. Roucairol. Using tabu search for solving a dynamic multiterminal truck dispatching problem. *European Journal of Operational Research*, 83 :411–429, 1995.
- [123] J. Renaud, F.F. Boctor, and L. Gilbert. An improved petal heuristic for the vehicle routing problem. *Journal of the Operational Research Society*, 47 :1156–1167, 1996.
- [124] Y. Rochat and E. Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1 :147–167, 1995.
- [125] R.A. Russell and W. Igo. An assignment routing problem. *Networks*, 9 :1–17, 1979.
- [126] A. Schrijver. On the history of combinatorial optimization (till 1960). *Handbook of Discrete Optimization (K. Aardal, G.L. Nemhauser, R. Weismantel, eds.)*, Elsevier, Amsterdam, pages 1–68, 2005.
- [127] P. Shaw. Using constraint and local search methods to solve routing problems. *In : CP-98, Fourth international conference on principles and practice of constraint programming, Lecture notes in computer science*, 1520 :417–431, 1998.
- [128] L.H. Shih and H.C. Chang. A routing and scheduling system for infectious waste collection. *Environmental Modeling and Assessment*, 6 :261–269, 2001.
- [129] M.M. Solomon. Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research*, 35 :254–265, 1987.
- [130] T. Stützle. Local search algorithms for combinatorial problems - analysis, algorithms and new applications. *DISKI - Dissertationen zur Künstlichen Intelligenz. Infix, Sankt Augustin, Germany*, 1999.
- [131] E. Taillard. Parallel iterative search methods for vehicle routing problems. *Networks*, 23 :661–673, 2005.
- [132] E.G. Talbi. A taxonomy of hybrid metaheuristics. *Journal of Heuristics*, 8(5) :541–564, 2002.
- [133] K.C. Tan, Y.H. Chew, and L.H. Lee. A hybrid multiobjective genetic algorithms for vehicle routing problem with time windows. *Computational Optimization and Applications*, 34 :115–151, 2006.
- [134] X. Tan, X. Luo, W.N. Chen, and J. Zhang. Ant colony system for optimizing vehicle routing problem with time windows. *Proceedings of the 2005 International Conference CIMCA-IAWTIC 05*, 2005.

- [135] C.D. Tarantilis and C.T. Kiranoudis. Boneroute : an adaptative memory-based maethod for effective fleet management. *Annals of operations Research*, 115(1) :227–241, 2002.
- [136] J. Teixeira, A.P. Antunes, and J.P. de Sousa. Recyclable waste collection planning - a case study. *European Journal of Operational Research*, 158(3) :543–554, 2004.
- [137] P.M. Thompson and H.N. Psaraftis. Cyclic transfer algorithms for the multivehicle routing and scheduling problems. *Operations Research*, 41 :935–946, 1993.
- [138] P. Toth and D. Vigo. *The vehicle routing problem*. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2001.
- [139] P. Toth and D. Vigo. The granular tabu search and its application to the vehicle routing problem. *INFORMS Journal on Computing*, 15(4) :333–346, 2003.
- [140] D.V. Tung and A. Pinnoi. Vehicle routing-scheduling for waste collection in hanoi. *European Journal of Operations Research*, 125 :449–468, 2000.
- [141] G. Verfaillie and N. Jussien. Constraint solving in uncertain and dynamic environments : A survey. *Constraints*, 3 :253–281, 2005.
- [142] S. Voss, S. Martello, I.H. Osman, and C. Roucairol. Meta-heuristics - advance and trends in local search paradigms for optimization. *Kluwer Academic Publishers, Dordrecht, The Netherlands*, 1999.
- [143] P. Wark and J. Holt. A repeated matching heuristic for the vehicle routing problem. *Operational Research Society*, 45 :1156–1167, 1994.
- [144] H. Xu, W. Fan, T. Wei, and L. Yu. An or-opt nsga ii algorithm for multi-objective vehicle routing problem with time windows. *Automation Science and Engineering, CASE 2008*, 2008.
- [145] P.C. Yellow. A computational modification to the savings method of vehicle scheduling. *Operational Research Quarterly*, 12 :281–283, 1970.
- [146] L. Zeng, H.L. Ong, and K.M. Ng. An assignment-based local search method for solving vehicle routing problems. *Asia-Pacific Journal of Operational research*, 22 i1 :84–104, 2005.

Annexe A

Résultats pour la stabilité hebdomadaire

A.1 Résultats hypothèse 1 : $|Z_{c_1}| < |Z_{c_2}|$

A.1.1 Résultats en termes de coût

Méthode	% c_1	Coût Ins- tances C1	Coût Ins- tances R1	Coût Ins- tances RC1	Coût Ins- tances C2	Coût Ins- tances R2	Coût Ins- tances RC2
1	10	2775	3037	3821	1445	2053	2575
2		2775	3048	3863	1419	2045	2554
3		2922	3116	3994	1490	2045	2554
4		2768	3057	3730	1601	2167	2569
5		2794	3104	3760	1675	2167	2662
6		2690	3003	3760	1430	2038	2513
1	20	2711	2951	3760	1395	1974	2512
2		3063	3041	3880	1369	1947	2486
3		3329	3054	3927	1369	1947	2486
4		2807	2974	3556	1802	2137	2649
5		2809	2995	3584	1805	2138	2667
6		2612	2860	3552	1457	1968	2466
1	30	2754	2843	3716	1427	1928	2428
2		3103	2948	3766	1361	1888	2437
3		3374	3004	3847	1863	2015	2614
4		2841	2884	3429	1768	2076	2684
5		2739	2907	3500	1769	2076	2724
6		2617	2733	3424	1400	1884	2423

TAB. A.1 – Résultats des méthodes en termes de coût par type d'instances

A.1.2 Résultats en termes de stabilité globale

Méthode	% c_1	Instances C1			Instances R1			Instances RC1		
		E_{nbv}	E_{dist}	E_{tps}	E_{nbv}	E_{dist}	E_{tps}	E_{nbv}	E_{dist}	E_{tps}
1	10	1	138	1263	1	144	187	0	156	186
2		0	149	772	0	122	111	0	114	126
3		0	210	754	0	110	197	0	108	153
4		1	183	1297	1	146	252	1	187	347
5		1	201	1191	1	201	362	1	236	392
6		0	208	648	1	122	243	2	183	359
1	20	1	207	1873	1	214	379	1	241	276
2		0	215	1246	1	168	341	0	151	189
3		1	436	1886	1	172	391	1	124	224
4		1	381	2460	2	339	677	1	315	505
5		2	385	2739	2	378	692	2	380	607
6		1	222	1911	3	243	542	2	352	632
1	30	1	290	2233	1	331	513	1	299	454
2		1	221	1943	1	254	517	1	256	382
3		2	317	3047	2	192	550	2	190	469
4		2	588	3887	4	545	1092	2	484	828
5		3	454	3976	5	555	1136	3	587	928
6		1	350	2892	4	381	883	2	510	895

TAB. A.2 – Résultats des méthodes en termes de stabilité globale pour les instances de type 1

A.1. Résultats hypothèse 1 : $|Z_{c_1}| < |Z_{c_2}|$

Méthode	% c_1	Instances C2			Instances R2			Instances RC2		
		E_{nbv}	E_{dist}	E_{tps}	E_{nbv}	E_{dist}	E_{tps}	E_{nbv}	E_{dist}	E_{tps}
1	10	0	154	1634	0	287	137	0	342	155
2		0	154	398	0	312	56	0	342	85
3		0	165	104	0	312	56	0	342	85
4		1	157	2402	1	153	847	1	184	800
5		1	230	3164	1	202	786	1	277	933
6		0	145	457	0	245	343	0	269	308
1	20	0	138	1279	0	344	369	0	364	283
2		0	192	315	0	391	172	0	392	111
3		0	191	315	0	391	172	0	392	111
4		1	290	3501	1	293	932	1	342	886
5		1	311	4001	1	315	1214	1	377	1077
6		0	81	2126	0	205	594	0	277	455
1	30	0	197	2613	0	329	493	0	459	526
2		0	213	805	0	417	167	0	479	329
3		0	425	1915	0	335	345	0	355	470
4		1	353	3795	1	373	1487	1	448	1497
5		1	382	5445	1	393	1559	1	498	1618
6		1	70	3054	0	259	727	1	242	734

TAB. A.3 – Résultats des méthodes en termes de stabilité globale pour les instances de type 2

A.1.3 Résultats en termes de stabilité du point de vue usagers

		Instances C1	Instances R1	Instances RC1	Instances C2	Instances R2	Instances RC2
Méthode	% c_1	E_h	E_h	E_h	E_h	E_h	E_h
1	10	1944	340	302	5094	1957	2167
2		1852	431	333	2251	1984	2106
3		2140	512	697	2089	1984	2106
4		1385	230	196	2603	1159	1884
5		728	230	138	0	0	178
6		7038	1530	1421	11585	9184	7206
1	20	2103	398	476	5517	4579	2660
2		6271	954	817	4926	4411	2572
3		4200	724	917	4926	4411	2572
4		1762	184	235	4043	2530	950
5		817	144	74	197	277	950
6		4834	1277	1091	10070	7152	5703
1	30	2967	802	472	5802	3952	3741
2		4761	1158	814	8338	4295	4890
3		4634	1128	767	10127	4747	4914
4		2236	338	499	1464	2414	1654
5		1439	180	245	1464	489	1112
6		4235	980	902	8559	6759	5834

TAB. A.4 – Résultats des méthodes en termes de stabilité “usagers” par type d’instances

A.1.4 Résultats en termes de stabilité du point de vue employés

Méthode	% c_1	Instances C1					Instances R1				
		$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}	$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}
1	10	154	495	6	16	484	112	140	4	17	440
2		94	655	6	16	565	73	149	2	17	507
3		236	841	7	19	560	192	226	8	23	484
4		78	607	0	9	528	93	131	0	12	447
5		57	543	0	6	505	72	101	0	8	385
6		372	1632	61	25	649	230	404	63	28	525
1	20	195	766	8	20	592	148	194	4	24	575
2		510	1650	36	26	764	238	444	20	33	673
3		776	1402	24	30	624	283	353	14	30	590
4		144	1075	0	14	634	125	249	0	15	550
5		126	647	0	7	517	101	156	0	10	442
6		313	1511	57	23	656	233	381	61	25	536
1	30	237	1186	5	27	864	186	309	4	33	802
2		486	1796	39	23	904	269	461	22	36	787
3		809	1651	29	27	714	363	480	20	34	680
4		209	1518	0	27	694	142	257	0	18	525
5		104	945	0	9	579	77	112	0	8	357
6		361	1704	55	20	742	241	404	59	19	550

TAB. A.5 – Résultats des méthodes en termes de stabilité “employés” pour les instances C1 et R1

A.1. Résultats hypothèse 1 : $|Z_{c_1}| < |Z_{c_2}|$

		Instances RC1					Instances C2				
Méthode	% c_1	$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}	$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}
1	10	65	145	2	16	145	154	376	13	13	356
2		86	172	4	17	619	67	212	2	11	558
3		274	298	13	28	578	25	184	0	11	607
4		72	139	0	12	549	57	133	0	9	553
5		59	129	0	8	492	59	1	0	0	327
6		287	484	63	32	604	104	495	11	22	545
1	20	133	198	4	26	777	106	488	7	20	721
2		231	328	14	34	828	51	395	0	21	901
3		335	358	14	35	757	51	395	0	21	901
4		117	174	0	19	689	110	484	0	12	691
5		83	165	0	12	595	13	37	0	2	353
6		300	459	63	25	677	172	620	21	21	542
1	30	167	341	3	32	982	185	650	13	22	630
2		236	384	18	35	995	95	585	5	24	1017
3		358	403	18	32	926	615	596	19	23	751
4		192	345	0	25	840	153	727	0	14	768
5		195	305	0	18	747	59	579	0	6	478
6		323	496	59	22	740	164	657	29	209	609

TAB. A.6 – Résultats des méthodes en termes de stabilité “employés” pour les instances RC1 et C2

		Instances R2					Instances RC2				
Méthode	% c_1	$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}	$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}
1	10	102	107	3	20	437	135	83	2	25	561
2		67	64	0	20	494	88	88	0	25	605
3		67	64	0	20	494	88	88	0	25	605
4		48	259	0	13	391	67	141	0	19	514
5		1	266	0	0	266	5	1	0	2	311
6		154	200	29	49	383	169	257	30	39	481
1	20	190	188	6	40	555	195	138	5	31	726
2		165	186	0	40	700	156	116	0	31	868
3		165	186	0	40	700	156	116	0	31	868
4		119	274	0	24	510	167	246	0	20	699
5		18	30	0	5	301	77	223	0	13	461
6		177	240	46	26	436	189	170	30	37	621
1	30	191	159	7	35	672	263	194	8	35	891
2		223	184	0	37	905	275	215	6	38	1046
3		290	190	9	35	795	390	225	12	37	961
4		143	515	0	24	586	147	422	0	20	776
5		40	188	0	6	284	50	97	0	7	360
6		130	212	32	32	590	223	183	37	27	716

TAB. A.7 – Résultats des méthodes en termes de stabilité “employés” pour les instances R2 et RC2

A.2 Résultats hypothèse 2 : $|Z_{c_1}| > |Z_{c_2}|$

A.2.1 Résultats en termes de coût

		Instances C1	Instances R1	Instances RC1	Instances C2	Instances R2	Instances RC2
Méthode	% c_1	E_{nbv}	E_{nbv}	E_{nbv}	E_{nbv}	E_{nbv}	E_{nbv}
1	60	2795	2732	3241	2271	2437	3313
2		3301	2742	3275	2597	2573	3490
1	70	2902	2659	3344	2402	2410	3117
2		3428	2673	3311	2524	2520	3249
1	80	2631	2562	3452	2441	2179	2905
2		3106	2537	3380	2745	2319	3017

TAB. A.8 – Résultats des méthodes en termes de coût par type d'instances

A.2.2 Résultats en termes de stabilité globale

		Instances C1			Instances R1			Instances RC1		
Méthode	% c_1	E_{nbv}	E_{dist}	E_{tps}	E_{nbv}	E_{dist}	E_{tps}	E_{nbv}	E_{dist}	E_{tps}
1	60	0	132	1385	6	766	1189	7	967	1397
2		0	592	3162	6	701	1124	7	905	1314
1	70	0	220	2110	6	711	1125	7	973	1391
2		0	669	4095	5	597	1019	7	925	1395
1	80	1	274	2504	7	758	1246	8	1088	1584
2		1	635	5555	5	537	1054	7	830	1335

TAB. A.9 – Résultats des méthodes en termes de stabilité globale pour les instances de type 1

		Instances C2			Instances R2			Instances RC2		
Méthode	% c_1	E_{nbv}	E_{dist}	E_{tps}	E_{nbv}	E_{dist}	E_{tps}	E_{nbv}	E_{dist}	E_{tps}
1	60	1	172	3079	1	414	842	2	748	1391
2		0	325	2756	1	626	1152	2	1038	1528
1	70	2	312	5532	2	370	828	2	618	1251
2		1	249	4055	1	695	1117	2	756	1322
1	80	2	514	6507	1	269	940	2	549	1385
2		1	521	6084	1	585	1226	2	831	1404

TAB. A.10 – Résultats des méthodes en termes de stabilité globale pour les instances de type 2

A.2.3 Résultats en termes de stabilité du point de vue usagers

		Instances C1	Instances R1	Instances RC1	Instances C2	Instances R2	Instances RC2
Méthode	% c_1	E_h	E_h	E_h	E_h	E_h	E_h
1	60	265	19	13	891	129	77
2		724	12	5	1079	134	72
1	70	242	7	15	644	39	62
2		446	17	11	919	45	46
1	80	135	18	15	196	5	46
2		391	6	15	528	33	44

TAB. A.11 – Résultats des méthodes en termes de stabilité “usagers” par type d’instances

A.2.4 Résultats en termes de stabilité du point de vue employés

Méthode	% c_1	Instances C1					Instances R1				
		$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}	$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}
1	60	280	1297	0	24	655	157	135	0	16	385
2		774	2787	0	28	1331	144	106	0	17	396
1	70	342	1709	0	22	815	160	144	0	16	374
2		842	3708	0	25	1388	154	151	0	17	456
1	80	396	1778	0	19	774	187	150	0	14	389
2		862	4420	0	23	1287	189	144	0	14	387

TAB. A.12 – Résultats des méthodes en termes de stabilité “employés” pour les instances C1 et R1

Méthode	% c_1	Instances RC1					Instances C2				
		$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}	$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}
1	60	132	110	0	12	353	217	510	0	18	374
2		168	134	0	13	402	286	970	0	19	698
1	70	166	169	0	15	427	306	207	0	16	379
2		164	186	0	14	457	347	678	0	16	583
1	80	173	194	0	13	435	256	542	0	12	381
2		220	145	0	13	351	555	1740	0	13	737

TAB. A.13 – Résultats des méthodes en termes de stabilité “employés” pour les instances RC1 et C2

A.2. Résultats hypothèse 2 : $|Z_{c_1}| > |Z_{c_2}|$

		Instances R2					Instances RC2				
Méthode	% c_1	$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}	$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}
1	60	190	227	0	11	280	237	177	0	12	432
2		370	412	0	12	695	365	300	0	12	712
1	70	256	260	0	10	334	228	210	0	12	387
2		377	427	0	9	796	321	324	0	12	719
1	80	270	365	0	8	522	222	202	0	10	339
2		414	568	0	9	869	433	461	0	10	801

TAB. A.14 – Résultats des méthodes en termes de stabilité “employés” pour les instances R2 et RC2

Annexe B

Résultats pour la stabilité dynamique

B.1 Résultats Insertion de nœuds :

B.1.1 Résultats en termes de coût

Méthode	% nœuds insérés	Coût Ins-tances C1	Coût Ins-tances R1	Coût Ins-tances RC1	Coût Ins-tances C2	Coût Ins-tances R2	Coût Ins-tances RC2
1	5	996	1780	2248	1269	1436	1675
2		964	1674	2073	1035	1332	1546
3		973	1597	1954	1069	1324	1549
4		924	1401	1614	865	1256	1419
5		953	1598	1871	1068	1353	1534
6		864	1398	1628	861	1094	1212
1	15	1035	1780	2279	1394	1517	1745
2		987	1642	2078	1116	1389	1580
3		971	1612	2018	1136	1372	1576
4		978	1384	1634	875	1302	1399
5		971	1578	1894	1083	1383	1556
6		864	1398	1628	861	1094	1212

TAB. B.1 – Résultats des méthodes en termes de coût par type d'instances

B.1.2 Résultats en termes de stabilité globale

Méthode	% nœuds insérés	Instances C1			Instances R1			Instances RC1		
		E_{nbv}	E_{dist}	E_{tps}	E_{nbv}	E_{dist}	E_{tps}	E_{nbv}	E_{dist}	E_{tps}
1	5	0	79	252	0	92	100	0	103	91
2		0	48	274	0	40	36	0	175	68
3		0	57	292	0	102	53	0	285	80
4		0	55	174	0	291	351	0	625	721
5		0	33	207	0	106	241	0	368	590
6		0	103	227	0	296	400	0	596	716
1	15	0	118	624	0	153	155	0	136	171
2		0	76	576	0	76	89	0	136	68
3		0	68	570	0	81	73	0	65	101
4		0	90	436	0	289	403	0	125	607
5		0	69	498	0	121	240	0	509	403
6		0	47	498	0	286	395	0	514	592

TAB. B.2 – Résultats des méthodes en termes de stabilité globale pour les instances de type 1

Méthode	% nœuds insérés	Instances C2			Instances R2			Instances RC2		
		E_{nbv}	E_{dist}	E_{tps}	E_{nbv}	E_{dist}	E_{tps}	E_{nbv}	E_{dist}	E_{tps}
1	5	0	77	168	0	126	68	0	157	81
2		0	185	130	0	99	97	0	59	55
3		0	151	92	0	105	87	0	51	98
4		0	327	658	0	135	173	0	110	259
5		0	124	432	0	83	176	0	34	173
6		0	332	296	0	217	140	0	306	62
1	15	0	250	384	0	240	122	0	271	198
2		0	111	342	0	120	95	0	107	139
3		0	94	391	0	119	100	0	109	136
4		0	269	295	0	122	354	0	165	223
5		0	112	312	0	114	168	0	91	166
6		0	282	211	0	184	142	0	263	80

TAB. B.3 – Résultats des méthodes en termes de stabilité globale pour les instances de type 2

B.1.3 Résultats en termes de stabilité du point de vue usagers

		Instances C1	Instances R1	Instances RC1	Instances C2	Instances R2	Instances RC2
Méthode	% nœuds insérés	E_h	E_h	E_h	E_h	E_h	E_h
1	5	760	30	7	886	9	14
2		2273	387	341	9983	1898	1482
3		2394	516	948	8240	1922	1752
4		3701	1709	1599	17731	4858	5784
5		4271	1577	1913	16786	4329	4690
6		6416	1654	1948	24518	9815	6982
1	15	1275	75	5	544	266	159
2		2577	467	590	7305	2658	2085
3		2709	653	639	6867	2690	2071
4		3535	1709	1216	16154	8184	8124
5		3784	1682	1609	15174	6515	6587
6		6411	1832	1400	20169	10496	7895

TAB. B.4 – Résultats des méthodes en termes de stabilité “usagers” par type d’instances

B.1.4 Résultats en termes de stabilité du point de vue employés

Méthode	% nœuds insérés	Instances C1					Instances R1				
		$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}	$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}
1	5	91	286	5	4	92	92	100	5	4	100
2		103	354	7	6	99	109	124	21	12	115
3		95	368	8	7	110	189	219	33	19	189
4		113	565	9	23	113	379	461	65	34	175
5		89	564	10	26	109	178	375	65	28	187
6		117	589	9	32	100	370	497	73	28	176
1	15	145	599	10	9	168	157	155	9	8	155
2		128	577	13	12	181	200	205	29	16	167
3		124	572	14	12	174	228	266	37	21	165
4		140	648	14	24	171	375	501	80	25	179
5		129	619	18	25	173	195	449	68	31	165
6		115	756	14	34	174	348	463	80	26	201

TAB. B.5 – Résultats des méthodes en termes de stabilité “employés” pour les instances C1 et R1

B.1. Résultats Insertion de nœuds :

Méthode	% nœuds insérés	Instances RC1					Instances C2				
		$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}	$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}
1	5	117	91	4	4	83	77	168	5	4	92
2		224	184	25	14	219	201	425	18	14	125
3		351	323	45	24	250	177	450	15	14	160
4		404	408	75	27	335	366	1089	22	41	282
5		257	429	81	27	311	141	874	32	36	270
6		418	464	84	26	295	375	689	21	43	158
1	15	136	171	10	8	181	250	384	10	7	169
2		209	289	34	21	289	203	670	24	14	206
3		257	361	43	25	378	192	631	22	14	196
4		404	429	79	26	403	339	1073	28	38	301
5		213	427	77	29	364	143	1073	38	35	311
6		405	465	87	23	426	337	853	29	40	240

TAB. B.6 – Résultats des méthodes en termes de stabilité “employés” pour les instances RC1 et C2

Méthode	% nœuds insérés	Instances R2					Instances RC2				
		$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}	$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}
1	5	126	68	5	4	60	157	81	5	4	81
2		153	161	16	12	128	117	89	15	10	143
3		161	172	17	12	142	143	163	18	12	167
4		289	297	29	30	220	386	330	23	47	310
5		133	219	27	27	167	131	385	23	39	181
6		277	390	34	46	135	310	144	19	55	131
1	15	247	122	9	10	135	272	204	9	8	193
2		205	188	9	23	176	208	229	19	13	208
3		222	208	9	25	194	207	232	20	14	221
4		302	467	10	39	250	432	529	34	44	347
5		153	317	9	35	223	156	330	34	42	222
6		251	366	9	41	173	306	284	28	50	208

TAB. B.7 – Résultats des méthodes en termes de stabilité “employés” pour les instances R2 et RC2

B.2 Résultats Augmentation de la quantité de nœuds :

B.2.1 Résultats en termes de coût

Méthode	% quantité diminuée	Coût Ins- tances C1	Coût Ins- tances R1	Coût Ins- tances RC1	Coût Ins- tances C2	Coût Ins- tances R2	Coût Ins- tances RC2
1	25	907	1455	1859	1130	1414	1564
2		857	1412	1789	861	1123	1212
3		867	1400	1695	840	1229	1363
4		881	1444	1761	1035	1377	1504
5		861	1398	1628	861	1123	1212
1	50	913	1441	1832	1130	1414	1564
2		864	1411	1793	861	1123	1212
3		869	1397	1708	836	1248	1406
4		883	1422	1729	1032	1394	1518
5		861	1398	1628	861	1123	1212

TAB. B.8 – Résultats des méthodes en termes de coût par type d'instances

B.2.2 Résultats en termes de stabilité globale

Méthode	% quantité diminuée	Instances C1			Instances R1			Instances RC1		
		E_{nbv}	E_{dist}	E_{tps}	E_{nbv}	E_{dist}	E_{tps}	E_{nbv}	E_{dist}	E_{tps}
1	25	0	0	0	1	54	99	1	103	185
2		0	50	163	1	40	155	1	64	194
3		0	39	87	0	19	80	0	86	89
4		0	26	126	0	50	126	0	70	144
5		0	64	179	0	48	109	0	128	162
1	50	0	7	14	1	46	89	1	111	176
2		0	48	214	1	35	113	1	85	197
3		0	44	131	0	25	76	0	72	136
4		0	30	135	0	39	92	0	53	118
5		0	71	236	0	55	183	0	99	174

TAB. B.9 – Résultats des méthodes en termes de stabilité globale pour les instances de type 1

B.2. Résultats Augmentation de la quantité de nœuds :

Méthode	% quantité diminuée	Instances C2			Instances R2			Instances RC2		
		E_{nbv}	E_{dist}	E_{tps}	E_{nbv}	E_{dist}	E_{tps}	E_{nbv}	E_{dist}	E_{tps}
1	25	0	0	0	0	0	0	0	0	0
2		0	267	220	0	287	111	0	352	62
3		0	312	612	0	191	149	0	150	214
4		0	106	280	0	43	82	0	51	77
5		0	393	698	0	286	114	0	349	66
1	50	0	0	0	0	0	0	0	0	0
2		0	269	220	0	287	111	0	352	62
3		0	306	751	0	165	156	0	168	145
4		0	117	389	0	36	65	0	39	60
5		0	370	390	0	291	121	0	354	84

TAB. B.10 – Résultats des méthodes en termes de stabilité globale pour les instances de type 2

B.2.3 Résultats en termes de stabilité du point de vue usagers

Méthode	% quantité diminuée	Instances C1	Instances R1	Instances RC1	Instances C2	Instances R2	Instances RC2
		E_h	E_h	E_h	E_h	E_h	E_h
1	25	0	482	467	0	0	0
2		4837	882	859	20790	7758	6493
3		2342	510	514	15068	3937	5330
4		2656	190	502	10317	1058	1490
5		5175	1130	1189	20790	7758	6493
1	50	66	560	579	0	0	0
2		4897	848	775	20790	7753	6493
3		2510	357	346	12511	4512	4973
4		2985	72	478	9536	2845	1823
5		5840	938	1115	20790	7753	6493

TAB. B.11 – Résultats des méthodes en termes de stabilité “usagers” par type d’instances

B.2.4 Résultats en termes de stabilité du point de vue employés

Méthode	% quan- tité dimi- nuée	Instances C1					Instances R1				
		$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}	$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}
1	25	0	0	0	0	0	26	84	4	0	75
2		50	254	0	25	0	66	148	10	24	116
3		42	153	0	17	18	243	393	54	18	389
4		30	193	0	15	11	82	156	20	5	188
5		65	312	0	27	11	348	486	75	25	476
1	50	9	36	0	0	12	25	86	4	0	70
2		57	292	0	26	12	54	155	10	22	127
3		54	257	2	19	37	180	266	36	15	344
4		40	249	2	20	37	89	184	21	8	203
5		77	398	1	28	38	292	420	68	26	530

TAB. B.12 – Résultats des méthodes en termes de stabilité “employés” pour les instances C1 et R1

Méthode	% quan- tité dimi- nuée	Instances RC1					Instances C2				
		$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}	$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}
1	25	20	147	6	1	46	0	0	0	0	0
2		85	231	19	27	149	269	308	7	42	70
3		328	389	60	25	452	329	1198	18	36	240
4		207	330	64	21	427	109	620	20	31	140
5		437	512	83	25	626	269	308	7	42	263
1	50	30	147	6	0	61	0	0	0	0	0
2		67	216	14	25	122	269	308	7	42	70
3		370	462	58	20	612	276	754	23	31	295
4		230	349	50	20	456	121	578	18	30	125
5		427	511	83	24	636	269	308	7	42	307

TAB. B.13 – Résultats des méthodes en termes de stabilité “employés” pour les instances RC1 et C2

B.2. Résultats Augmentation de la quantité de nœuds :

Méthode	% quan- tité dimi- nuée	Instances R2					Instances RC2				
		$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}	$E_{dist-emp}$	$E_{tps-emp}$	E_{compo}	E_{ordre}	E_{capa}
1	25	0	0	0	0	0	0	0	0	0	0
2		287	222	18	47	85	353	172	8	57	68
3		297	221	23	28	187	327	360	17	45	172
4		82	49	11	7	65	94	113	3	18	81
5		287	222	18	47	100	353	172	8	57	103
1	50	0	0	0	0	0	0	0	0	0	0
2		287	222	18	47	85	353	172	8	57	68
3		244	282	25	30	213	287	359	16	42	230
4		93	168	14	17	55	100	78	6	18	88
5		287	222	18	47	102	353	172	8	57	128

TAB. B.14 – Résultats des méthodes en termes de stabilité “employés” pour les instances R2 et RC2