Vehicle Routing for City Logistics

Diego Cattaruzza

NNT : *2014 EMSE 0737*

THÈSE

présentée par

Diego CATTARUZZA

pour obtenir le grade de
Docteur de l'Ecole Nationale Supérieure des Mines de Saint-Etienne

Spécialité : Génie Industriel

VEHICLE ROUTING FOR CITY LOGISTICS

OPTIMISATION DE TOURNÉES DE VÉHICULES POUR LA LOGISTIQUE URBAINE

Gardanne, 27 mars 2014

Membres du jury

Rapporteurs :	Michel GENDREAU	Professeur, Polytechnique Montréal, Canada
	Maria Grazia SPERANZA	Professeur, Université de Brescia, Italie
Examinateurs :	Stéphane DAUZÈRE-PÉRÈS	Professeur, ENSM-SE, France
	Alain QUILLIOT	Professeur, Université Blaise Pascal, France
	Daniele VIGO	Professeur, Université de Bologne, Italie
	Roberto WOLFLER CALVO	Professeur, Université Paris 13, France
Directeur de thèse :	Dominique FEILLET	Professeur, ENSM-SE, France
Co-Directeur de thèse :	Nabil ABSI	Chargé de recherche HDR, ENSM-SE, France

Spécialités doctorales :	Responsables :
SCIENCES ET GENIE DES MATERIAUX	K. Wolski Directeur de recherche
MECANIQUE ET INGENIERIE	S. Drapier, professeur
GENIE DES PROCEDES	F. Gruy, Maître de recherche
SCIENCES DE LA TERRE	B. Guy, Directeur de recherche
SCIENCES ET GENIE DE L'ENVIRONNEMENT	D. Graillot, Directeur de recherche
MATHEMATIQUES APPLIQUEES	O. Roustant, Maître-assistant
INFORMATIQUE	O. Boissier, Professeur
IMAGE, VISION, SIGNAL	JC. Pinoli, Professeur
GENIE INDUSTRIEL	A. Dolgui, Professeur
MICROELECTRONIQUE	

EMSE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

AVRIL	Stéphane	PR2	Mécanique et ingénierie	CIS
BATTON-HUBERT	Mireille	PR2	Sciences et génie de l'environnement	FAYOL
BENABEN	Patrick	PR1	Sciences et génie des matériaux	CMP
BERNACHE-ASSOLLANT	Didier	PR0	Génie des Procédés	CIS
BIGOT	Jean Pierre	MR(DR2)	Génie des Procédés	SPIN
BILAL	Essaid	DR	Sciences de la Terre	SPIN
BOISSIER	Olivier	PR1	Informatique	FAYOL
BORBELY	Andras	MR(DR2)	Sciences et génie de l'environnement	SMS
BOUCHER	Xavier	PR2	Génie Industriel	FAYOL
BRODHAG	Christian	DR	Sciences et génie de l'environnement	FAYOL
BURLAT	Patrick	PR2	Génie Industriel	FAYOL
COURNIL	Michel	PR0	Génie des Procédés	DIR
DARRIEULAT	Michel	IGM	Sciences et génie des matériaux	SMS
DAUZERE-PERES	Stéphane	PR1	Génie Industriel	CMP
DEBAYLE	Johan	CR	Image Vision Signal	CIS
DELAFOSSE	David	PR1	Sciences et génie des matériaux	SMS
DESRAYAUD	Christophe	PR2	Mécanique et ingénierie	SMS
DOLGUI	Alexandre	PR0	Génie Industriel	FAYOL
DRAPIER	Sylvain	PR1	Mécanique et ingénierie	SMS
FEILLET	Dominique	PR2	Génie Industriel	CMP
FOREST	Bernard	PR1	Sciences et génie des matériaux	CIS
FORMISYN	Pascal	PR0	Sciences et génie de l'environnement	DIR
FRACZKIEWICZ	Anna	DR	Sciences et génie des matériaux	SMS
GARCIA	Daniel	MR(DR2)	Génie des Procédés	SPIN
GERINGER	Jean	MA(MDC)	Sciences et génie des matériaux	CIS
GIRARDOT	Jean-jacques	MR(DR2)	Informatique	FAYOL
GOEURIOT	Dominique	DR	Sciences et génie des matériaux	SMS
GRAILLOT	Didier	DR	Sciences et génie de l'environnement	SPIN
GROSSEAU	Philippe	DR	Génie des Procédés	SPIN
GRUY	Frédéric	PR1	Génie des Procédés	SPIN
GUY	Bernard	DR	Sciences de la Terre	SPIN
GUYONNET	René	DR	Génie des Procédés	SPIN
HAN	Woo-Suck	CR	Mécanique et ingénierie	SMS
HERRI	Jean Michel	PR1	Génie des Procédés	SPIN
INAL	Karim	PR2	Microélectronique	CMP
KERMOUCHE	Guillaume	PR2	Mécanique et Ingénierie	SMS
KLOCKER	Helmut	DR	Sciences et génie des matériaux	SMS
LAFOREST	Valérie	MR(DR2)	Sciences et génie de l'environnement	FAYOL
LERICHE	Rodolphe	CR	Mécanique et ingénierie	FAYOL
LI	Jean Michel		Microélectronique	CMP
MALLIARAS	Georges	PR1	Microélectronique	CMP
MOLIMARD	Jérôme	PR2	Mécanique et ingénierie	CIS
MONTHEILLET	Franck	DR	Sciences et génie des matériaux	SMS
PERIER-CAMBY	Laurent	PR2	Génie des Procédés	DFG
PIJOLAT	Christophe	PR0	Génie des Procédés	SPIN
PIJOLAT	Michèle	PR1	Génie des Procédés	SPIN
PINOLI	Jean Charles	PR0	Image Vision Signal	CIS
POURCHEZ	Jérémy	CR	Génie des Procédés	CIS
ROUSTANT	Olivier	MA(MDC)		FAYOL
STOLARZ	Jacques	CR	Sciences et génie des matériaux	SMS
SZAFNICKI	Konrad	MR(DR2)	Sciences et génie de l'environnement	CMP
TRIA	Assia		Microélectronique	CMP
VALDIVIESO	François	MA(MDC)	Sciences et génie des matériaux	SMS
VIRICELLE	Jean Paul	MR(DR2)	Génie des Procédés	SPIN
WOLSKI	Krzystof	DR	Sciences et génie des matériaux	SMS
XIE	Xiaolan	PR0	Génie industriel	CIS

ENISE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

BERGHEAU	Jean-Michel	PU	Mécanique et Ingénierie	ENISE
BERTRAND	Philippe	MCF	Génie des procédés	ENISE
DUBUJET	Philippe	PU	Mécanique et Ingénierie	ENISE
FORTUNIER	Roland	PR	Sciences et Génie des matériaux	ENISE
GUSSAROV	Andrey	Enseignant contractuel	Génie des procédés	ENISE
HAMDI	Hédi	MCF	Mécanique et Ingénierie	ENISE
LYONNET	Patrick	PU	Mécanique et Ingénierie	ENISE
RECH	Joël	MCF	Mécanique et Ingénierie	ENISE
SMUROV	Igor	PU	Mécanique et Ingénierie	ENISE
TOSCANO	Rosario	MCF	Mécanique et Ingénierie	ENISE
ZAHOUANI	Hassan	PU	Mécanique et Ingénierie	ENISE

Mise à jour : 07/04/2013

PR 0	Professeur classe exceptionnelle
PR 1	Professeur 1ère classe
PR 2	Professeur 2ème classe
PU	Professeur des Universités
MA (MDC)	Maître assistant
DR	Directeur de recherche

Ing.	Ingénieur
MCF	Maître de conférences
MR (DR2)	Maître de recherche
CR	Chargé de recherche
EC	Enseignant-chercheur
IGM	Ingénieur général des mines

SMS	Sciences des Matériaux et des Structures
SPIN	Sciences des Processus Industriels et Naturels
FAYOL	Institut Henri Fayol
CMP	Centre de Microélectronique de Provence
CIS	Centre Ingénierie et Santé

Acknowledgement

With this thesis terminates a three-year period of my life. And as each time something finishes, I like to take a moment to stop by and look back into the ending adventure. To recall the joyful or difficult events, and, most of all, to thank the people that walked with me along this way and made this experience a *great* experience. In particular, I would love to thank:

- my supervisors Dominique and Nabil. This document would not have been possible without their wise guidance and all their suggestions. They were always available for discussions and they supported me in all the different steps of the work, enhancing my passion for the research. Working with them was always enjoyable and our rapport professional, but at the same time truly friendly. I will always be extremely grateful to them. It is scaring to think that from now on I will have to walk in the academic path without them holding my scientific hand. The only regret I have is to not have collected all the jokes they made on me and on Italians. It would probably be enough material for another publication.

- Jakey, with whom I arrived here in France almost at the same time and with the same level of French: null. This probably made us attaching to each other deeper and faster. I was lucky to find a person like Jakey on my way, with whom I could talk about everything, from technical, personal and professional problems to funny and stupid things, in the office as in front of a beer. I will definitively miss his constant and helpful presence.

- Elisabeth, that taught me most of the French I know. And it is funny to think that she is German. We initially shared the huge and (at that time) empty office. Until climatic problems pushed for an earlier split.

- Thierry and his way to be. Just a couple of minutes with him, and his good cheer can make your day turning positively. Thank to him and his accent I understood what it means to be a true *marseillais* (and learned that, apparently, one can recognize different brands of *pastis* only smelling the glasses).

- Federico and his amazing simplicity. Each morning he entered the office with a huge smile and started teaching me Spanish. I will not forget that smile. But I cannot say the same for the Spanish.

- Pierre, that introduced me into the soccer side of Marseille (even if not into the golden age of OM). Those who know how important is soccer for me, would understand

how important was to have a person like Pierre in the department. His encyclopedic knowledge of soccer gave us the possibility for long (and interesting) discussions.

- Dario and Vincenzo, that made me always feeling home. I would not imagine that the need of Italianity can be satisfied by an Emiliano with Neapolitan origins and a Sicilian. But with them, the fact that Italian pizza is the best pizza in the world is a dogma and does not require discussions. Moreover, there is no need of explaining that pasta *is not* a side dish.

- my old-time friends Pampa, Ricky, Paolino and Sepy. They represent the essence of friendship: each time we meet it is as we have never separated, as there is no physic and time distances among us.

- my two lovely grandmas Nonna Lida and Nonna Maria, that even from far away keep taking care of me, both of them with their unique style and strength. My aunt Albina: it does not matter how old I get, I will always be her *nini*.

- my sister Giorgia, for the link that is growing day after day. With the time ticking away, it increases the awareness of having on my side a person on which I can always count. And each day passes by, I miss her more.

- my parents Gavino and Gabriella, that always believed and will always believe in me. There was not a instant of my life in which I had not felt their support and love.

- two dear persons with whom I would have loved to share this moment of my life: my uncle Nello, that with his *keep going, little kid*, pushed me until here, and will push me in all the future adventures; my friend Agno, with which I should have shared the phD experience, if the destiny would have not decided differently.

- *ma chèrie*, as they say here in France, Stefy, that constantly support and sustain all my choices. Even if this means leaving a country, quitting a job, departing from family and friends. With her love life is brighter and without her, this, probably, would have never begun.

This thesis was supported by the French National Research Agency (ANR - Agence Nationale de la Recherche) and was part of the project MODUM - Mutualisation et Optimisation de la distribution Urbaine de Marchandises.

I thank all MODUM project members (in alphabetical order): Laurent Alfandari, Paolo Gianessi, Jesus Gonzalez-Feliu, Lucas Letocart, Rafik Makhloufi, Frédéric Meunier, Jean-Louis Routhier, Roberto Wolfler Calvo, and Tai-Yu Ma Collaborating with them I learned that the teamwork is probably not the easier way to work, but surely the richer. Moreover, I thank Thibaut Vidal and Daniele Vigo for the precious and inspiring collaborations.

Contents

1 Introduction **17**

2 Vehicle Routing Problems for City Logistics **23**

2.1 Introduction 23

2.2 City logistics: definitions, stakes and routing problems 24

2.2.1 Scope 24

2.2.2 The stakes of public authorities and private stakeholders 25

2.2.3 VRP for city logistics 26

2.3 Urban Goods Movements 29

2.3.1 UGM classification 29

2.3.2 Statistical analysis of UGM flows 32

2.3.3 UMG as routing problems 35

2.3.4 Discussion 35

2.4 Prominent families of vehicle routing problems for urban good movements . 36

2.4.1 VRPs with time-dependent travel times 37

2.4.2 Multi-level Vehicle Routing Problem 41

2.4.3 Dynamic vehicle routing problem 44

2.4.4 Multi-Trip VRP 48

2.5 Conclusion 51

3 The Multi Trip Vehicle Routing Problem: A Survey **55**

3.1 Introduction 55

3.2 Multi-Trip Vehicle Routing Problem 57

3.2.1 Heuristics 61

3.2.2 Exact methods . . . 63
3.2.3 Variants of the MTVRP . . . 63
3.3 MTVRP with Time Windows . . . 64
3.3.1 MTVRPTW with limited trip duration . . . 65
3.3.2 Variants of the MTVRPTW . . . 72
3.4 MTVRP in maritime . . . 73
3.5 Production, inventory and multi-trip transportation problems . . . 75
3.6 The MTVRP in practice . . . 76
3.7 Conclusions and perspectives . . . 78

4 A Memetic Algorithm for the Multi Trip Vehicle Routing Problem 81
4.1 Introduction . . . 81
4.2 Literature review . . . 82
4.3 A memetic algorithm for the MTVRP . . . 84
4.3.1 Solution representation and search space . . . 85
4.3.2 A Split algorithm for the multi-trip problems . . . 85
4.3.3 Crossover . . . 89
4.3.4 Local search - education and repair procedures . . . 90
4.3.5 Population structure, initialization and stopping criteria . . . 91
4.3.6 Survivor strategy . . . 92
4.4 Combined Local Search . . . 92
4.5 Computational results . . . 95
4.5.1 Parameter settings . . . 97
4.5.2 Discussion . . . 98
4.6 Conclusion and future work . . . 106

5 The Multi Trip Vehicle Routing Problem With Time Windows and Release Dates 111
5.1 Introduction . . . 111
5.2 Problem definition and notation . . . 112
5.2.1 Problem definition . . . 113
5.2.2 Notation . . . 114

5.2.3 Relationship with the VRPPDTW 114
5.2.4 Problem characterization 114
5.3 Literature Review 115
5.4 Method 116
5.4.1 Algorithm outline 116
5.4.2 Solution representation and search space 117
5.4.3 Local search 118
5.4.4 A Split algorithm for multi-trip problems with time windows and release dates 121
5.4.5 Survivor strategy 126
5.5 Computational results 127
5.5.1 Instances for the MTVRPTWR 127
5.5.2 Tuning 129
5.5.3 Setting of $\mathcal{L}_{\text{threshold}}$ 131
5.5.4 Results on MTVRPTWR instances 132
5.5.5 Comparison with Hernandez et al. [105] 136
5.6 Conclusions and perspectives 140

6 An Iterated Local Search for the Multi Commodity Multi Trip Vehicle Routing Problem with Time Windows 141

6.1 Introduction 141
6.2 Problem definition and notation 142
6.3 Algorithm 143
6.3.1 Objective function and search space 145
6.3.2 Giant tour creation 146
6.3.3 Local Search 146
6.3.4 *Repair* procedure 149
6.3.5 Crossover 149
6.4 *AdSplit* Procedure 150
6.4.1 Graph construction 150
6.4.2 Assignment of trips to vehicles 151
6.5 Computational results 152

6.5.1 Parameter setting 153
6.5.2 Results on Battarra et al.'s instances 153
6.5.3 MTVRP versus VRP in fleet dimensioning problems 155
6.6 Conclusions 164

7 MODUM Vehicle Routing Problem 167
7.1 Introduction 167
7.2 MODUM system 168
7.3 MODUM VRP: static version 169
7.4 MODUM VRP: dynamic version 173
7.5 Insights on the simulator 174
7.5.1 Client requests and instance requests 174
7.5.2 Handling of new client requests 174
7.5.3 Capacities at CDC 175
7.5.4 Day structure 176
7.5.5 Result analysis 176
7.6 Conclusions and perspectives 176

8 Conclusions and perspectives 177

List of Figures v

List of Tables vii

List of Algorithms xi

A New feasible solution for CMT4_T_H^1_7 xiii

B Detailed results on Solomon instances xv

Bibliography xvi

Résumé

Introduction

L'urbanisation est définie comme *le processus de transition d'un milieu rural à une société plus urbaine* (Rodrigue [179]) et a caractérisé le 20e siècle. La population urbaine mondiale a atteint 3,5 milliards d'habitants en 2010, soit 51,6% de la population totale (Rodrigue [179]). Les villes sont le centre de gravité de notre société, où toutes les activités se déroulent, activités commerciales ou du loisir.

Plus d'habitants vivants dans les villes signifie une plus grande quantité de marchandises et de personnes qui se déplacent dans les villes. Le transport entre en jeu, à la fois pour les personnes et à la fois pour les marchandises, et avec celle-là, se manifestent tous ses avantages et ses inconvénients. Le transport crée l'accès à des événements et rend les marchandises disponibles pour les utilisateurs. Tout cela avec un certain coût environnemental (bruit, pollution de l'air, pollution de l'eau) et un coût social (embouteillages, accidents).

Selon les prévisions, l'urbanisation continuera à croire: 84% de la population européenne devrait vivre dans les villes en 2050 (Commission européenne [153]), contre 72% en 2007. L'importance du transport des personnes et des marchandises a été comprise par les autorités publiques, les entreprises et les chercheurs qui mènent des études pour optimiser ces processus.

En particulier, la communauté scientifique a récemment adopté deux concepts fondamentaux. Le premier est celui de *city logistics* (Ruske [184], Kohler [122], Taniguchi et al. [200], [49]) définie par Taniguchi et al. [200] comme *le processus pour optimiser totalement les activités de logistique et de transport par des entreprises privées dans des zones urbaines tout en tenant compte de l'environnement, de la circulation, de la congestion, du trafic et de la consommation d'énergie dans le cadre d'une économie de marché.* Un concept beaucoup plus vaste est celui de *logistique urbaine* (Ambrosini et al. [4], Anderson et al. [5]), qui comprend l'organisation, le comportement, la réglementation, les éléments de financement ainsi que des approches de collaboration, pour étudier les processus logistiques et les mouvements de marchandises et les flux des services dans les zones urbaines. La plupart des auteurs se réfèrent également au concept de city logistics lorsqu'il s'agit de ce concept plus large. Dans cette thèse, nous allons suivre cet usage.

Cette thèse trouve sa position dans ce contexte et vise à étudier un système particulier de livraison mutualisée de marchandises dans les centres urbains. Son origine est le projet

MODUM [1]. MODUM étudie un système de distribution spécifique reposant sur un anneau de centres de distribution urbaine (CDU), situé dans la périphérie de la ville. La marchandise est d'abord transportée au CDU par les transporteurs, puis transférée aux véhicules respectueux de l'environnement, que nous appelons *vans*, en charge de la livraison finale aux clients. Des systèmes de livraison réels basés sur la présence de CDU ont été mis en œuvre. Certains ont été des réussites (par exemple le système de distribution mis en place à La Rochelle), d'autres ont échoués pour différentes raisons (par exemple, le CDU n'a pas été bien situés, ou par manque de soutien financier public). MODUM vise à étudier ce type de systèmes, pour mieux comprendre les raisons potentielles de succès et les raisons possibles d'échec. L'étude porte sur les différents aspects et impacts: économique (évaluation des gains et des coûts prévus), environnementaux (réduction des émissions de CO_2, et de la pollution sonore) et sociaux (réduction de la congestion).

Différentes universités sont impliquées dans MODUM: l'Université de Paris 13; l'Ecole des Ponts, Paris Tech; l'Université de Lyon; et l'Ecole Nationale Supérieure des Mines de Saint-Etienne. Ils partagent les quatre objectifs principaux du projet. En particulier, il a été prévu:

- de développer un outil d'aide à la décision pour la conception et le dimensionnement du système (déterminer le nombre de CDU et les localiser, déterminer le service sur l'anneau, la taille de la flotte de vans);
- de développer un outil pour déterminer la planification opérationnelle de la distribution;
- de développer un outil de simulation qui évalue un tel système de livraison;
- de recueillir les données nécessaires pour les trois points précédents et d'analyser les résultats de la simulation.

Cette thèse se concentre sur le second point et en particulier elle traite de l'organisation des opérations accomplies par les vans chaque jour. Plus précisément, notre objectif est de développer un outil capable de fournir des solutions de haute qualité pour le problème de tournées riche qui se pose à partir du projet.

Les problèmes de tournées (VRP) ont été très largement étudiés depuis le travail fondateur "The Truck Dispatching Problem" de Dantzig et Ramser [58] publié en 1959. Une planification à moindre coût pour servir tous les clients et en respectant les contraintes de capacité des camions doit être déterminée. La littérature sur le sujet est vaste et continue d'augmenter, année après année, de manière importante. Les spécialistes introduisent des caractéristiques nouvelles au problème d'origine afin de mieux représenter la situation particulière qu'ils doivent étudier. Fenêtres de temps, flotte hétérogène, plusieurs dépôts, niveau de stocks, sont seulement quelques-unes des plus courantes. Un grand nombre de problèmes se posent, soit en tenant compte de ces caractéristiques, individuellement, ou en combinant certaines d'entre elles.

[1] `http://www-lipn.univ-paris13.fr/modum`

La variété des recherches concerne aussi les méthodes de résolutions qui ont été proposées. D'un côté, nous pouvons trouver des méthodes exactes, telles que le branch-and-bound, le branch-and-price, le branch-and-cut, la programmation dynamique, entre autres. Le VRP a été prouvé $\mathcal{NP}$-difficile, et (à moins que $\mathcal{P} = \mathcal{NP}$) les méthodes exactes sont peu susceptibles d'être efficaces pour la résolution de cas de grande taille. Actuellement, au mieux, des instances avec une centaine de clients peuvent être résolues à l'optimalité dans des temps de calcul raisonnables.

Le $\mathcal{NP}$-difficulté du VRP justifie les efforts de la communauté mis dans le développement de méthodes heuristiques. Ces méthodes peuvent rapidement proposer une bonne solution mais sans garantir l'optimalité. Les méthodes varient d'heuristiques constructives simples (heuristiques de Clarke et Wright, Clarke et Wright [42], heuristique du sweep, Gillet et Miller [89]) éventuellement suivie par des méthodes d'amélioration locale, jusqu'à des schéma plus complexes. Les méthodes d'amélioration locale commencent à partir d'une solution et cherchent une solution meilleure en explorant les *voisins* de la solution qui peuvent être atteints en *perturbant* la solution courante. Ces perturbations sont généralement appelées *mouvements*. Des exemples de mouvements sont λ-opt, Or-opt, ou divers types d'échanges.

Une limite des méthodes d'amélioration locale repose sur le fait qu'elles peuvent facilement se retrouver piégées dans des *minima locaux*, c'est à dire, des solutions sans voisins strictement meilleurs. Différentes méthodes ont été proposées pour débloquer la recherche lorsque cette situation survient. Ces méthodes peuvent être essentiellement classées en trois groupes: les méthodes basées sur les trajectoires, les méthodes basées sur les populations, et les méthodes hybrides. Dans le premier cas, l'espace de recherche est exploré en acceptant, lors du passage d'une solution à une voisine, le déplacement vers des solutions de moins bonne qualité. Des exemples sont la recherche tabou, le recuit simulé, et la recherche à voisinage variable, parmi d'autres. Dans le second cas, de nouvelles solutions sont générées en combinant des solutions sélectionnées à partir d'un ensemble de solutions appelé population. Des exemples sont les algorithmes de colonies de fourmis et les algorithmes génétiques. Enfin, des algorithmes hybrides combinent les caractéristiques des algorithmes précédents pour profiter de la force de chaque méthode.

Un objectif de cette thèse est de développer un outil en mesure de fournir la planification quotidienne des véhicules circulant dans le système complexe considéré dans MODUM. Cela implique la considération d'un problème de tournées avec plusieurs contraintes secondaires comme la possibilité pour le véhicule de faire plusieurs trajets, de faire de la collecte et de la livraison, de considérer les fenêtres de temps, de prendre en compte plusieurs dépôts, et de considérer les dates de mise à disposition des marchandises. De plus, nous allons avoir à traiter des données représentatives de la situation d'une ville réelle de taille moyenne à grande. Il est alors raisonnable de s'attendre à des cas contenant des centaines de clients et des dizaines de véhicules. En outre, l'outil devra réagir rapidement aux nouvelles demandes de services qui peuvent se produire au cours des opérations.

Pour toutes ces raisons, cette thèse se concentre sur le développement d'algorithmes *heuristiques* pour des problèmes particuliers de tournées qui se posent dans le contexte urbain. Nous commençons par considérer le VRP avec trajets multiples, où une flotte de véhicules, basée en un dépôt central unique, doit servir un ensemble de clients. Les véhicules sont autorisés à être rechargés quand ils retournent au dépôt puis être réacheminés. Dans une deuxième étape, nous considérons un problème plus riche. En particulier, nous associons à chaque client une fenêtre de temps qui représente l'intervalle de temps où le service devrait avoir lieu et nous associons une date de mise à disposition à chaque marchandise. Cette date représente l'instant auquel la marchandise est disponible au dépôt. Dans une troisième étape, nous considérons la possibilité de livrer des produits différents et incompatibles. Certaines marchandises sont incompatibles et donc elles ne peuvent pas être transportées dans le même véhicule en même temps. D'autre part, tous les véhicules peuvent transporter toutes les marchandises. Dans cette étape, la taille de la flotte doit également être determinée. Dans la dernière étape, nous considérons le problème de tournées impliqué dans le projet MODUM.

Un travail parallèle a consisté en l'écriture de deux états de l'art. Le premier concerne les problèmes de tournées en milieu urbain et le second porte sur le problème de tournées avec trajets multiples. Dans la première étude, nous proposons une classification et une analyse de la circulation des marchandises dans un contexte urbain. Ensuite, nous considérons les principales caractéristiques des problèmes de tournées rencontrés lors de la livraison des marchandises dans les villes. Les conséquences algorithmiques de chaque caractéristique sont mises en évidence. L'état de l'art sur le problème de tournées avec trajets multiples propose un aperçu complet des recherches effectuées sur le sujet.

La section suivante détaille le schéma de la thèse et la liste de tous les chapitres et les sujets qui sont traités.

Structure de la thése

La thèse est composée de huit chapitres, y compris celui-ci. Ils sont énumérés dans la suite avec une brève explication sur leur contenu.

Chapitre 2: Vehicle Routing for City Logistics

Le transport de marchandises impacte profondément la qualité de vie des habitants, en particulier dans les zones urbaines. Afin de contribuer à la compréhension du transport de marchandises, ce chapitre donne un aperçu des mouvements de marchandises dans des zones urbaines de nos jours.

Nous avons analysé les documents qui ont étudié explicitement le transport de marchandises dans un contexte urbain tant du point de vue des entreprises que des autorités publiques. A partir de ces analyses, nous avons identifié les problèmes auxquels les chercheurs, les entreprises et les autorités doivent faire face lorsqu'ils s'intéressent aux livraisons de marchandises en ville. Dans un contexte urbain, des aspects particuliers sont considérés et les solutions proposées peuvent prendre en compte des aspects environnementaux pour améliorer l'habitabilité de la ville tout en déterminant un planning de livraison efficace. Quatre problèmes principaux ont été identifiés: le time-dependent VRP, le VRP multi-échelon, le VRP dynamique et le VRP avec trajets multiples. Une vue d'ensemble sur tous ces problèmes est alors donnée. Le document conclut en indiquant des directions pour la recherche et les développements futurs.

Ce chapitre a été produit en collaboration avec Jesus Gonzalez-Feliu du Laboratoire d'Economie des Transports, Lyon, France.

Chapitre 3: The Multi-Trip Vehicle Routing Problem: A Survey

Dans ce chapitre nous présentons un état de l'art sur le problème de tournées de véhicules avec trajets multiples. Ce problème est une extension du très connu problème de tournées de véhicules, où les véhicules sont autorisés à être rechargés au dépôt une fois qu'ils finissent un trajet. Il a été introduit il y a plus de 25 ans, et depuis lors, les chercheurs ont travaillé sur ce problème, mais aucun état de l'art complet n'a été proposé.

La contribution de ce chapitre est principalement de combler cette lacune, en proposant une synthèse exhaustive des travaux qui ont été consacrés à ce sujet. Les nombreuses applications pratiques qui sont mises en évidence devraient encourager les chercheurs et les experts à mettre leurs efforts sur ce problème.

Chapitre 4: A Memetic Algorithm for the Multi-Trip Vehicle Routing Problem

Nous considérons le problème de tournée de véhicules avec trajets multiples, dans lequel un ensemble de clients géographiquement dispersés doit être desservi par une flotte homogène de véhicules. Chaque véhicule peut effectuer plusieurs trajets au cours de la journée de travail. L'objectif est de minimiser le temps de voyage total tout en respectant les contraintes temporelles et de capacité.

Le problème est particulièrement intéressant dans le contexte de la logistique urbaine, où les clients sont situés dans les centres-villes. Des restrictions juridiques peuvent en effet favoriser l'utilisation de petits véhicules de capacité limitée pour effectuer les livraisons. Cela conduit à des trajets beaucoup plus courts que la journée de travail. Un véhicule peut alors retourner à son dépôt et être rechargé avant de commencer un autre trajet de service.

Nous proposons un algorithme génétique hybride pour le problème. Surtout, nous introduisons un nouvel opérateur de recherche locale basé sur la combinaison de mouvements et d'échanges classique dans le contexte du VRP. Notre procédure est comparée avec celles de la littérature et démontre sa performance par rapport aux algorithmes précédents en matière de qualité moyenne des solutions. En outre, une nouvelle solution réalisable a été trouvée ainsi que de nombreuses nouvelles meilleures solutions connues.

Ce chapitre a été produit en collaboration avec Thibaut Vidal de l'Université de Technologie de Troyes, Troyes, France et CIRRELT, Montréal, Canada.

Chapitre 5: The Multi-Trip Vehicle Routing Problem With Time Windows and Release Dates

Le problème de tournées de véhicules avec trajets multiples, avec fenêtre de temps et dates de disponibilité est une variante du problème de tournées de véhicules avec trajets multiples où une fenêtre de temps est associée à chaque client et une *date de disponibilité* est associée à chaque marchandise qui doit être livré à un certain client. La date de disponibilité représente le moment où la marchandise est disponible au dépôt pour la livraison finale.

Le problème est pertinent dans le contexte de la logistique urbaine, où les systèmes de livraison qui sont étudiés se basent sur des centres de distribution urbaine (CDU). Les camions arrivent au CDU pendant toute la journée de travail et livrent les marchandises qui sont transférées à des véhicules respectueux de l'environnement en charge de réaliser la livraison final aux clients.

Nous proposons un algorithme génétique pour le problème sur la base de la représentation des chromosomes comme tour géant ainsi qu'une procédure de *découpage* pour obtenir des solutions. En particulier, un graphe auxiliaire acyclique est construit sur la séquence de clients à desservir représentée par le tour géant. Chaque arc représente une tournée qui doit être affectée à un véhicule spécifique. La meilleure solution associée à chaque tour géant pourrait être obtenue seulement en considérant toutes les affectations possibles de toutes les tournées à tous les véhicules. Ceci n'est pas possible dans un temps de calcul raisonnable. Pour cette raison, nous calculons tout d'abord le plus court chemin sur le graphe auxiliaire. Cette première étape sélectionne des arcs sur le graphe qui correspondent à un ensemble de tournées. Dans un deuxième temps, nous affectons les tournées sélectionnées par le plus court chemin aux véhicules.

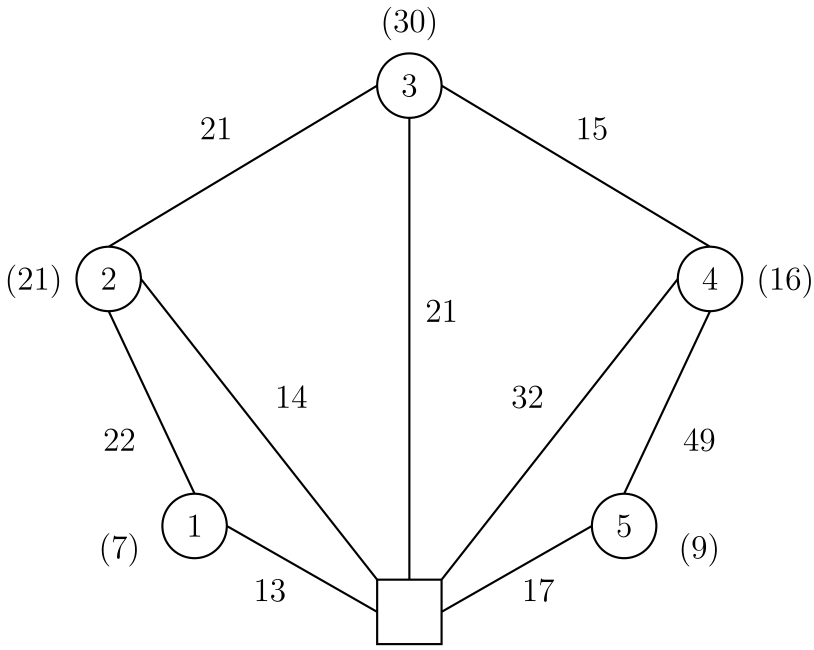


Figure 1: Exemple sur une petite instance avec cinq clients

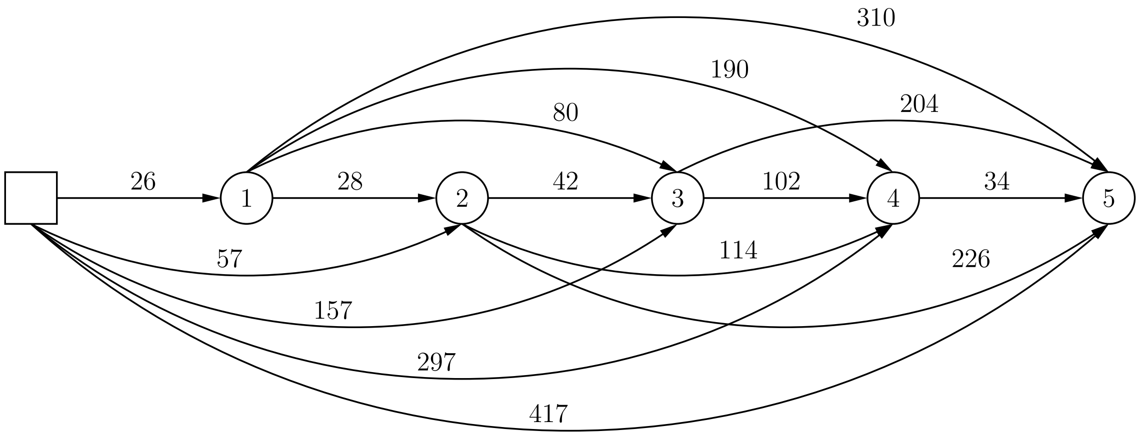


Figure 2: Exemple de graphe auxiliaire

Le problème ici traité est nouveau et est donc introduit pour la première fois dans le cadre de cette thèse. Pour cette raison, nous proposons un ensemble d'instances pour le problème même et une mesure pour caractériser leur difficulté de résolution, c'est à dire la difficulté à trouver un planning réalisable. Un planning réalisable est un planning qui respecte toutes les contraintes de temps et de capacité.

Chapitre 6: An Iterated Local Search for the Multi-Commodity Multi-Trip Vehicle Routing Problem with Time Windows

Le problème de tournées de véhicules avec produits multiples, fenêtre du temps et trajets multiples vise à déterminer un plan d'acheminement d'un ensemble de véhicules pour desservir un ensemble de clients qui exigent des produits *incompatibles* entre eux. Deux produits sont dit incompatibles s'ils ne peuvent pas être transportés ensemble dans le même véhicule. Par contre, les véhicules sont autorisés à effectuer plusieurs trajets au cours de la journée de travail et à transporter n'importe quel produit. L'objectif est de minimiser le nombre de véhicules utilisés.

Contrairement aux problèmes précédemment traités celui-ci est caractérisé par un objectif stratégique plutôt qu'un objectif opérationnel. De plus, les instances à traiter sont extraites d'un problème réel qu'une entreprise doit résoudre pendant une semaine de travail. Il s'agit donc de déterminer le planning de livraison pour trois à quatre cents clients en utilisant environs quatre-vingt véhicules. Une solution du problème est représentée dans la Figure 3.

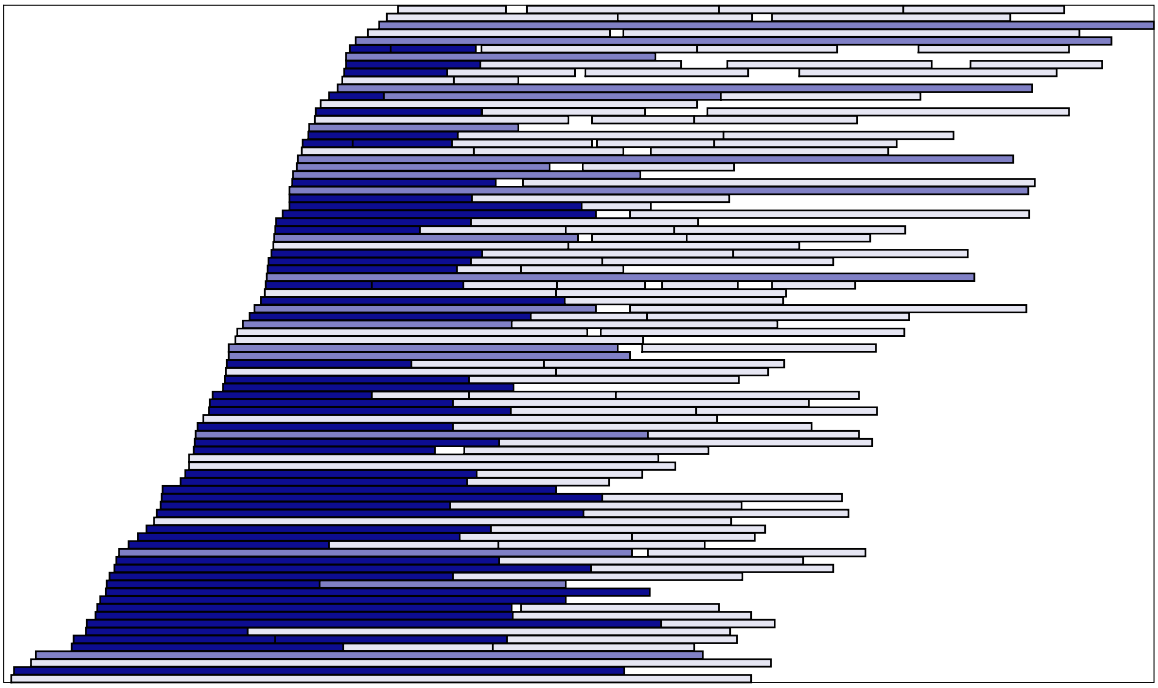


Figure 3: Tournées pour le cinquième jour: différentes couleurs représentent différents produits

Nous proposons une procédure de recherche locale itérative plus performante que le précédent algorithme proposé dans la littérature pour le même problème. De plus, nous effectuons une analyse sur le bénéfice que l'on peut obtenir en introduisant l'aspect trajets multiples au niveau du dimensionnement de la flotte de véhicules. Les résultats sur les instances classiques pour le VRPTW montrent que, dans certains cas, la flotte peut être réduite de moitié.

Ce chapitre a été produit en collaboration avec Daniele Vigo de l'Université de Bologne, Bologne, Italie.

Chapitre 7: MODUM Vehicle Routing Problem

Ce chapitre présente et définit formellement le problème considéré par le projet MODUM. Nous donnons le schéma général du simulateur qui est conçu pour évaluer les performances du système. Une heuristique pour faire face au problème général est proposée aussi. Elle sera intégrée dans le simulateur.

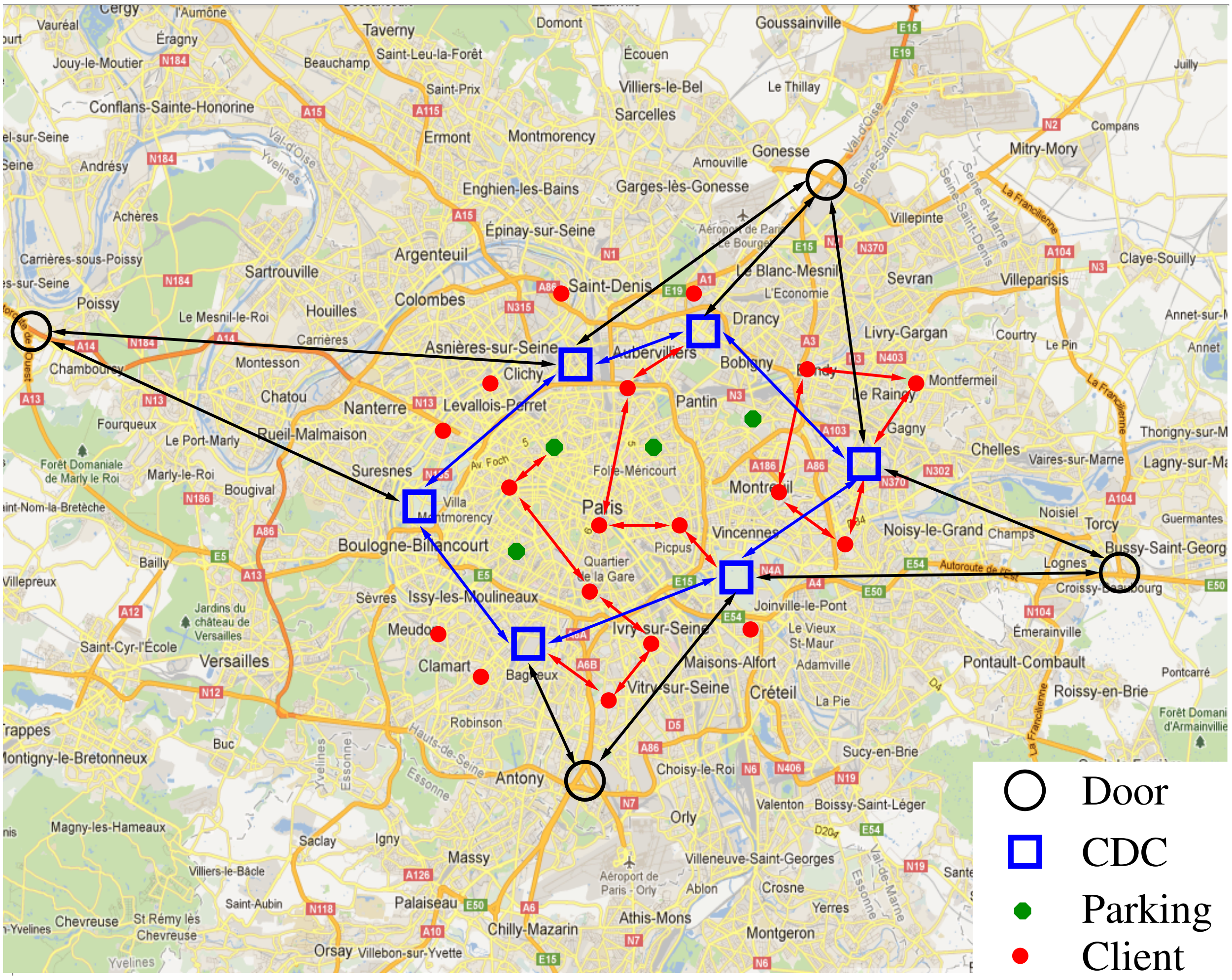


Figure 4: Schéma étudié dans le projet MODUM

Le développement du simulateur est géré par un autre partenaire du projet MODUM (l'Ecole Nationale des Pont et Chaussées) et n'est pas présenté dans la thèse. Comme le développement du simulateur et la collecte de données ne sont pas encore terminées, les résultats ne sont pas présentés ici.

Ce chapitre a été fait en collaboration avec tous les partenaires du projet.

Chapitre 8: Conclusion et perspectives

Ce chapitre termine le manuscrit en proposent des perspectives de recherche ouvertes par les travaux faits dans le cadre de cette thèse.

Description du projet MODUM et du système de livraison conçu

Le projet MODUM vise à étudier un système de distribution mutualisée basé sur un anneau constitué de centres de distribution urbaine (CDU) stratégiquement situés dans la banlieue de la ville. Les transporteurs qui doivent transporter aux clients leur marchandise, livrent les marchandises à l'un des CDU plutôt que directement au client. Le CDU est une plateforme logistique utilisée pour transférer les marchandises d'un véhicule à un autre. Habituellement, il est caractérisé par le fait de ne pas avoir la possibilité de stocker les marchandises ou de ne pouvoir la stocker que pour une très courte période.

Différentes études et projets ont été menés sur l'efficacité de ces systèmes de livraison en ville. Dans certains cas, les résultats ont été positifs et les perspectives prometteuses, d'autres projets ont échoué. Le succès dépend de plusieurs facteurs et acteurs. Les CDU doivent être bien situés: ils doivent être à proximité du centre-ville, mais aussi faciles d'accès pour les camions qui arrivent généralement par les autoroutes. L'ensemble du système doit être bien dimensionné afin d'être en mesure de recevoir toutes les marchandises, mais sans être pour autant trop coûteux. Les autorités doivent imposer et, en particulier, faire respecter la limitation d'accès au centre-ville, afin de forcer les camions à s'arrêter au CDU au lieu d'aller directement servir les clients. D'un autre côté, les transporteurs demandent un système efficace: les marchandises doivent être livrées aux clients à temps, et faire faire des économies (de temps et/ou d'argent) aux transporteurs. La livraison aux clients doit être effectuée par des véhicules respectueux de l'environnement, que nous appellerons *vans*, afin de réduire la pollution atmosphérique et le bruit.

Le projet MODUM doit évaluer un tel système de livraison, doit identifier les principaux leviers qui peuvent faire le succès du projet s'ils sont bien appliqués. En outre, il vise à développer un outil d'aide à la décision basé sur la simulation à événements discrets, pour aider les praticiens à dimensionner le système en ce qui concerne la ville où ils seraient situés.

Le système conçu fonctionne pendant la journée de travail et les camions amènent d'une façon continue les marchandises au CDU. Les nouvelles requêtes de service sont intégrées dans la planification actuelle en temps réel. Les requêtes non desservies, ou des marchandises déposées au CDU un jour à l'avance seront utilisées pour constituer la planification initiale des trajets des vans pour le lendemain. La simulation fournit différents indicateurs de performance qui sont utilisés pour évaluer le système.

A noter que la mise au point du simulateur est en dehors du cadre de cette thèse. Ce développement est dirigé par un autre partenaire dans le projet de MODUM. En outre, une mission importante liée à l'utilisation du simulateur est la conception et la collecte des données réalistes, qui sont gérées par un troisième partenaire. Lors de la rédaction de cette thèse, cette tâche n'est pas terminée non plus.

Le schéma général du système MODUM est représenté dans la Figure 4.

Le système considère à la fois le flux entrant et le flux sortant des marchandises. Nous commençons par décrire le premier type de flux. Les camions livrent des marchandises au

CDU (carrés bleus), d'où les *vans* sont utilisés pour accomplir les dernières livraisons vers les clients indiqués par des points rouges. Les camions sont censés entrer dans le système (approche de la ville) à partir des *portes* (cercles noirs). Une porte représente la source de la marchandise et peut être un aéroport, un port, une gare, une zone industrielle ou une bretelle d'autoroute. Le système considère aussi la présence de *parkings* (points verts) qui peuvent être utilisés par les vans.

Tous les CDU sont liés par une navette qui circule régulièrement et visite tous les CDU (flèches bleues dans la Figure 4) et transfère les marchandises d'un CDU à un autre. Par exemple, un camion décharge des marchandises au CDU situé au nord du centre-ville, et les marchandises doivent être livrées dans le sud de la ville. Il pourrait être pratique d'utiliser la navette pour déplacer la marchandise à un plus proche CDU, plutôt que d'envoyer un van de l'autre côté de la ville.

Dès que la marchandise arrive au bon CDU elle est transférée aux vans en charge de la réalisation de la livraison finale aux clients. Les tournées de livraison ont leurs origines à un certain CDU et se terminent au même CDU, à un autre CDU ou dans un parking. Les clients doivent être visités pendant leur fenêtre de temps et l'attente chez le client n'est pas autorisée. Un véhicule peut aller dans un parking et attendre.

Les flux sortants fonctionnent comme suit. Les marchandises sont collectées chez les clients et envoyées vers les portes en passant par les CDU. Les tournées de collecte finissent toujours à un certain CDU, mais elles peuvent commencer à partir du même CDU, d'un autre CDU ou d'un parking. Dans tous les cas, les vans sont initialement vides. Une fois que la marchandise est au CDU elle peut être déplacée vers un autre CDU plus pratique par la navette ou transférée dans un camion et envoyée vers une porte.

Nous supposons que les tournées sont soit tournées de livraison soit tournées de collecte (c'est-à-dire, seules des livraisons ou des collectes sont effectuées tout au long d'une tournée). Lorsque la distinction n'a pas besoin d'être faite, le mot *service* sera utilisé pour indiquer l'un des deux types de voyages. Chaque van est affecté à un *trajet*, c'est à dire, à un ensemble de services que le van lui-même doit effectuer en séquence.

Le système considère la possibilité de louer les vans. Dans ce cas, un utilisateur peut louer un van, en précisant les endroits (CDU ou parking) et les horaires auxquels il souhaite obtenir et rendre le véhicule. Nous parlerons de em libre-service.

Sur le plan opérationnel, on peut supposer que le transporteur appelle le *centre opérationnel* (OPC) et fournit les informations telles que le CDU où il va déposer la marchandise et à quel moment il atteindra ce CDU. En outre, il doit préciser la destination finale des marchandises. Le OPC exécute un algorithme rapide pour évaluer la possibilité d'intégrer la requête dans la planification actuelle afin d'accepter ou de rejeter la requête. Si elle est acceptée, elle doit être insérée dans le plan d'exécution actuel. Tous les CDU sont considérés comme points de départ potentiels des services. Si le meilleur CDU n'est pas celui où le camion est arrivé, la navette est utilisée pour déplacer les marchandises à un autre plus adapté. Cela rend les produits disponibles pour être chargés dans un van et commencer la livraison finale.

Une planification est créée avant que la journée de travail ne commence en utilisant toutes les requêtes connues. Cette planification est construite par la résolution d'un problème statique. Au cours de la journée de travail, de nouvelles requêtes sont insérées dans la planification et un problème dynamique est résolu.

Conclusion

La logistique urbaine est devenue un sujet de recherche important au cours des dernières années. Des systèmes de livraison efficaces sont étudiés et de nouvelles solutions sont recherchées. Ces solutions cherchent à réaliser des économies, à respecter l'environnement, et à créer des villes agréables sans pénaliser les activités du centre-ville. Dans ce contexte, le projet MODUM étudie un nouveau système de livraison efficace, sur la base d'un anneau de centres de distribution urbains (CDU) situés autour de la ville.

Une première contribution de cette thèse a consisté en l'analyse des mouvements de marchandises dans le contexte urbain. L'enquête met en évidence la façon dont les mouvements des marchandises dans un contexte urbain sont effectués: les trajets de livraison desservent plusieurs clients. Cela laisse de l'espace pour l'optimisation du planning de livraison de la journée, qui peut induire des réductions du temps de trajet total, de la distance parcourue et réduire aussi la pollution.

Néanmoins, lorsqu'il s'agit de systèmes urbains, une telle planification efficace n'est pas facile à réaliser. L'environnement métropolitain a des caractéristiques particulières comme les heures de pointe, les embouteillages, les politiques de restriction ou d'autres caractéristiques qui sont renforcées dans ce contexte, comme les accidents de voiture, qui affectent profondément le planning et doivent être prises en compte. La conception d'un planning efficace passe par la compréhension de l'environnement urbain.

Cette thèse contribue à cette tâche en proposant une analyse des travaux réalisés par des chercheurs dans le contexte de la logistique urbaine et en les compilant dans un état de l'art. En outre, à partir de ces travaux, nous extrapolons les principales caractéristiques qu'un problème de tournées doit prendre en compte pour produire une planification de la livraison urbaine efficace. Nous avons examiné chaque caractéristique dans l'état de l'art et, en plus, nous avons proposé une synthèse des travaux liés à chaque sujet. Le but de ce travail est de fournir aux futurs chercheurs un aperçu sur le tournées en ville, et de guider le lecteur vers une plus grande et plus précise analyse.

Une caractéristique particulière que nous avons détectée dans la phase précédente, est l'aspect *trajets multiples*. Les tournées de livraison en centre-ville sont souvent faites par des véhicules petits et respectueux de l'environnement, que nous appelons *vans*. L'autonomie ou capacité des vans est limitée et, en conséquence, limite la longueur des tournées qui sont normalement plus courtes que la journée de travail. Les vans peuvent alors être réutilisés plusieurs fois, afin d'exploiter tout l'horizon de temps. Le problème académique qui se pose est le Multi-Trip Vehicle Routing Problem (MTVRP).

Ce *modus operandi* est commun aux systèmes de logistique urbaine: l'agencement structurel des villes (en raison de l'héritage médiéval) et les motivations écologiques forcent les professionnels à prendre en compte les vans pour les livraisons finales. Cependant, le contexte urbain n'est pas le seul dans lequel plusieurs trajets par jour sont considérés. La livraison de marchandises aux supermarchés, le réapprovisionnement des stations service, la collecte des ordures et le transport du bétail sont quelques exemples d'applications de tournées dans lesquels l'aspect trajets multiples a été pris en compte dans la littérature. Par ailleurs, des travaux récents admettent plusieurs trajets par véhicule dans des problèmes de production-scheduling et de l'inventory-routing.

En dépit de son intérêt pratique, le MTVRP n'a pas été intensivement étudié par les chercheurs et la littérature n'est pas aussi riche que l'on pourrait imaginer. Cette thèse contribue à combler cette lacune et propose des méthodes de résolution heuristiques pour le MTVRP, pour le MTVRP avec fenêtres de temps et dates de mise à disposition, et pour le MTVRP avec produits multiples et fenêtres de temps. Une contribution supplémentaire de cette thèse dans le contexte des tournées multiples est de rassembler tous les travaux effectués sur le sujet dans un deuxième état de l'art. Il s'agit de la première synthèse complète sur le MTVRP. Cet état de l'art montre la limite des algorithmes exacts, compare les résultats des méthodes heuristiques et donne des références aux instances classiques du problème. De plus, une section examine l'intérêt pratique des modèles. Cette partie du travail devrait encourager d'autres chercheurs à mettre leurs efforts dans ce domaine particulier de problèmes de tournées.

Comme déjà mentionné, nous avons développé trois algorithmes pour des problèmes de tournées avec trajets multiples. Nous avons d'abord développé un algorithme génétique pour le MTVRP où les chromosomes sont des permutations des clients, généralement appelés *tour géant*. Une procédure de découpage (basée sur les travaux de Prins [166]) transforme les chromosomes en solutions. En outre, nous avons proposé un opérateur de recherche locale adapté au problème. A notre connaissance, à ce jour, seuls les opérateurs propres au VRP ont été utilisés dans le contexte des tournées multiples, à la seule exception du remplacement et des échanges de trajets entre les véhicules. L'opérateur que nous avons proposé combine mouvements classiques pour le VRP qui n'améliorent pas la solution actuelle, avec l'échange de tournées entre les différents véhicules, à la recherche d'une amélioration globale. Ceci est la première étape du développement d'opérateurs spécialisés qui exploitent la structure particulière du problème. Les résultats obtenus définissent l'état de l'art sur les instances classiques du problème.

Un deuxième travail a introduit un nouveau problème: le MTVRP avec fenêtre du temps et dates de disponibilité (MTVRPTWR). Une date de disponibilité est associée à chaque marchandise et elle représente l'instant où la marchandise elle-même devient disponible pour la livraison au dépôt. Elle modélise la dépendance entre les flux *externes* et les flux *internes*. Les flux externes sont les déplacements effectués par les camions lourds qui apportent les marchandises à un CDU. Les flux internes sont les déplacements effectués par les vans d'un CDU vers les clients ou vice-versa. Des discussions avec des praticiens d'entreprises privées ont certifié l'intérêt pratique du problème, ainsi que l'absence de travaux académiques sur le sujet.

Nous avons proposé un algorithme génétique hybride pour le MTVRPTWR. Les tours géants sont transformés en solutions au moyen d'une procédure de labellisation qui généralise la procédure de découpage que nous avons présentée pour le MTVRP. Nous avons aussi créé un ensemble d'instances pour le problème. Nous avons utilisé les fameuses instances de Solomon comme instances de base. En particulier, la même date de disponibilité peut être associée à différents clients: ainsi est simulée l'arrivée d'un camion au dépôt. Différentes familles d'instances sont présentées, chacune caractérisée par une différence moyenne entre la date de disponibilité et la borne supérieure de la fenêtre de temps.

Avec l'introduction du problème MTVRPTWR, plusieurs directions de recherche sont maintenant ouvertes et laissées pour des travaux futurs. L'une est le développement de méthodes exactes. Même si les méthodes exactes sont peu susceptibles d'être efficaces dans la résolution des problèmes de taille réelle, elles peuvent être utilisées pour évaluer les algorithmes heuristiques. Une deuxième direction naturelle est l'introduction des aspects dynamiques. La marchandise arrive au dépôt par les axes routiers et les rues urbaines. Ainsi les camions peuvent souffrir de retards dus à des conditions de circulation défavorables. Les conducteurs peuvent communiquer leur retard au centre d'opération qui peut modifier la date de disponibilité et mettre à jour la planification en conséquence. Les aspects dynamiques peuvent être introduits, même du fait de la possibilité que certains camions peuvent notifier leur arrivée au cours de la journée de travail. Dans les deux cas, l'utilisation des nouvelles technologies communicantes permet d'établir un lien entre les conducteurs de camions et le centre d'opération qui coordonne les opérations de livraison. Une troisième sous-classe des problèmes qui peuvent découler du MTVRPTWR considère des aspects stochastiques. Des temps de déplacement stochastiques peuvent être considérés au niveau des flux internes ainsi qu'au niveau des flux externes. Dans ce dernier cas, l'incertitude du temps de déplacement sur les routes parcourues par les camions peut être reflétée dans les dates de sortie.

La dernière variante du MTVRP que nous avons traité dans la thèse, organise le planning de livraison de produits incompatibles, à savoir, qui ne peuvent pas être transportés dans le même véhicule en même temps. Cette variante a été appelés le MTVRP avec produits multiples et fenêtre de temps, et a été introduite par Battarra et al. [17]. Le problème est plutôt stratégique qu'opérationnel: le nombre de véhicules utilisés doit être minimisé. Une procédure itérative (ILS) est proposée pour un ensemble d'instances qui se pose dans un contexte réel. Les résultats sont le nouvel état de l'art pour le problème: la flotte est réduite pour toutes les instances et peut atteindre 10%. En outre, une analyse est menée sur l'avantage potentiel de permettre aux véhicules d'effectuer plusieurs tournées dans le problème de dimensionnement de la flotte. Sachant que le MTVRP avec produits multiples et fenêtre de temps généralise le MTVRP avec fenêtres de temps, nous utilisons notre procédure ILS sur les instances classiques de Solomon et les instances de Gehring et Homberger conçues pour le VRPTW avec l'objectif de minimiser la flotte en premier et la distance parcourue en second. Les résultats ont montré que, dans certains cas, la flotte peut être réduite de moitié, seulement en laissant les véhicules effectuer plusieurs tournées, tandis que dans d'autres, il est peu probable que l'introduction de l'aspect trajets multiples puisse réduire le nombre de véhicules. Une analyse plus poussée a montré que la raison de la réduction possible de la taille, est que l'horizon de temps n'est pas bien exploité et que donc les véhicules restent non utilisés dans le dépôt sur une partie de l'horizon de temps.

La contribution de cette partie du travail n'est donc pas seulement algorithmique, avec le développement d'une procédure efficace. Les résultats obtenus et leur analyse devraient rendre le lecteur conscient de l'importance de l'aspect trajets multiples dans la réalisation d'une planification efficace de livraison. Ici, avec "efficace", nous ne référons pas seulement au niveau opérationnel, mais aussi au niveau stratégique, où nous regardons comment minimiser la taille de la flotte et les coûts connexes.

L'axe de recherche qui découle de ce travail et de l'analyse des résultats que nous avons obtenus est l'intérêt qu'aurait la conception d'algorithmes permettant de résoudre efficacement à la fois un problème VRP et son homologue avec trajets multiples. La procédure commune actuelle est d'évaluer les algorithmes conçus pour des problèmes avec tournées multiples sur des instances où la capacité limitée des véhicules les oblige à effectuer plusieurs trajets. Les méthodes ne sont pas évaluées sur des instances admettant une solution optimale pour le VRP. Par conséquent, leur capacité à trouver des solutions de type VRP n'est pas évaluée. L'analyse que nous avons menée a montré que, dans certaines situations, une solution MTVRP est nécessaire pour exploiter efficacement l'horizon de temps, dans d'autres cas, une solution du VRP est suffisante. De là découle le besoin d'algorithmes efficaces pour ces deux problèmes simultanément.

La thèse se termine par une définition formelle du problème de tournées riche qui se pose dans le projet MODUM et qui a guidé notre travail. Le problème concerne plusieurs dépôts et des parkings utilisés pour garer les vans. Les vans font plusieurs tournées au cours de la journée de travail et font à la fois de la collecte et de la livraison (bien que dans des tournées séparées). Les tournées commencent à un certain dépôt et se terminent au même ou à un autre dépôt ou, aussi, dans un parking. Une date de disponibilité est associée à chaque marchandise à livrer, alors que des dates d'échéances sont associées aux marchandises à collecter chez le client.

Comme la conception du système MODUM a inspiré l'introduction des problèmes de tournées avec dates de mise à disposition, ces systèmes peuvent être la motivation pour l'introduction de problème de tournées avec des dates d'échéance associées aux marchandises qui doivent être collectées chez les clients et ramenées au dépôt. Les problèmes de tournées avec date de mise à disposition modélisent un problème de livraison pur, les problèmes avec dates d'échéance modélisent des problèmes de collecte purs et peuvent définir une autre classe intéressante de problèmes de tournées. Une extension naturelle serait alors de considérer simultanément des dates de disponibilité et des dates d'échéance dans le même problème, avec par conséquent l'introduction d'une troisième classe de problème à l'intersection des deux précédentes.

Au moment de la conclusion de cette thèse, le simulateur en charge de l'évaluation du système était en cours de développement. L'objectif est d'évaluer le système dans différents scénarios. L'évaluation sera basée sur différents indicateurs comme le coût du système, la quantité d'émissions de CO_2, le nombre de fenêtres de temps violées, le nombre de camions utilisés, le facteur de charge moyen. Nous n'avons pas tenu compte de temps de déplacements qui varieraient en fonction du temps, bien que cela caractérise généralement les centres-villes et les zones urbaines. Des temps de déplacement constants sont utilisés pour une première évaluation du système. Cependant, d'autres recherches devraient examiner la manière d'intégrer des temps de déplacement fonctions du temps ainsi que des temps de parcours stochastiques pour une meilleure représentation de l'environnement urbain. En outre, le système MODUM considère une navette qui relie tous les dépôts et déplace les marchandises d'un dépôt à un autre, par exemple plus adapté à la livraison finale. Les recherches futures peuvent étudier l'organisation de la navette et évaluer la possibilité d'optimiser la planification de la navette en se basant sur la demande exacte plutôt que d'avoir un service préprogrammé. Chacune des directions que nous avons énumérées peut améliorer le système considéré dans MODUM et améliorer sa fonctionnalité. Nous sommes optimistes quant au fait que la simulation puisse démontrer l'efficacité du système, à condition que les choix tactiques et stratégiques aient été bien effectués. Les travaux futurs devraient se poursuivre avec l'objectif d'amélioration de ces systèmes qui, selon nous, offrent une opportunité concrète pour la construction de villes durables et orientées vers les personnes.

Chapter 1

Introduction

Urbanization is defined as *the process of transition from a rural to a more urban society* (Rodrigue [179]) and has characterized the 20th century. The world's urban population reached 3.5 billion people in 2010, representing 51.6% of the total population (Rodrigue [179]). Cities are the center of gravity of our society, where all the activities take place, from commercial to leisure.

More inhabitants living in cities means a bigger quantity of merchandise and people moving into the cities. Transportation comes into play, both for people and merchandise, and with it, come into play all its benefits and disadvantages. Transportation connects people, creates access to events, makes goods available to users. All this with a certain environmental (noise, air and water pollution) and social cost (congestion, accident deaths).

Forecasts predict that urbanization will continue growing: 84% of the European population is expected to live in cities by 2050 (European Commission [153]), against 72% in 2007. The importance of both people and goods transportation has been understood by authorities, companies and scholars and studies to optimize the process have been carried out.

In particular the scientific community has recently adopted two main concepts. The first is that of *city logistics* (Ruske [184], Kohler [122], Taniguchi et al. [200], [49]) defined by Taniguchi et al. [200] as *the process for totally optimizing the logistics and transport activities by private companies in urban areas while considering the traffic environment, the traffic congestion and energy consumption within the framework of a market economy.* A much larger concept is that of *urban logistics* (Ambrosini et al. [4], Anderson et al. [5]), which includes all the organizational, behavioral, regulation and financing elements, as well as collaborative approaches, to study the logistics processes and the movements of goods and service flows in urban areas. Most authors however also refer to city logistics when dealing with this larger concept. In the thesis, we will follow this usage.

This thesis finds its position into this context and aims at studying a particular system of mutualized merchandise delivery in city centers. Its origin is the MODUM project[1]. MODUM studies a specific delivery system based on a ring of city distribution centers (CDC), located in the city outskirts. Merchandise is first transported to the CDC by carriers and then transferred to eco-friendly vehicles, that we call *vans*, in charge of the final delivery

[1] `http://www-lipn.univ-paris13.fr/modum`

to customers. Actual delivery systems based on the presence of CDC have been implemented. Some cases were successful (for example the delivery system implemented in La Rochelle), other failed for different reasons (for example the CDC was not well located, or for lack of public financial support). MODUM aims at studing this kind of systems, to better understand the potential reasons of success and the possible reasons of fail. The study addresses different aspects and impacts: economic (evaluation of the expected gains and costs), environmental (reduction of CO_2 emission and noise pollution) and social (reduction of congestion).

Different universities are involved in MODUM: University of Paris 13; École des Ponts, Paris Tech; University of Lyon; and the École Nationale Supérieure des Mines de Saint-Etienne. They share the four main objective of the project. In particular it was planned:

- to develop a decision support tool for designing and sizing the system (determine the number of CDC and locate them; determine the service on the ring; size the fleet of vans);
- to develop a tool for determining the distribution operational planning.
- to develop a simulation tool that evaluates such a delivery system;
- to collect the data needed for the three previous points and analyze the results of the simulation.

This thesis focuses on the second point and in particular it deals with the organization of operations that vans need to accomplish each day. Specifically, our objective is to develop a tool able to provide high-quality solutions for the rich routing problem that arises from the project.

Routing problems have been deeply studied since the seminal work “The Truck Dispatching Problem” of Dantzig and Ramser [58] published in 1959. A least-cost planning to serve all the customers and respecting truck capacity constraints needs to be determined. The literature on the subject is large and keeps growing year after year. Scholars introduced different characteristics into the original problem in order to better represent the particular situation they were facing. Time windows, heterogeneous fleet, multiple depots, backhauls, inventories are some of the most common. A large number of problems arise, either by considering these characteristics individually or by combining some of them.

The variety of the research concerns as well the solution methods that have been proposed. On one side we can find exact methods such as branch and bound, branch and price, branch and cut, dynamic programming, among others. The VRP has been proven to be $\mathcal{NP}$-hard, then (unless $\mathcal{P} = \mathcal{NP}$) exact methods are unlikely to be efficient in solving large-size instances. Currently, at best, instances with hundred customers can be solved to optimality within reasonable computational time.

The $\mathcal{NP}$-hardness of the VRP justifies the efforts the community put into the development of heuristic methods. These methods can quickly propose a good solution but without the guarantee of the optimality. Methods vary from simple constructive heuristics (Clarke and Wright heuristic, Clarke and Wright [42], sweep heuristic, Gillet and Miller [89]) eventually followed by local improving methods, to more complex schemes. Local improving methods start from a solution and look for a better one exploring *neighbour* solutions that can be reached *perturbing* the current solution. Perturbations are usually called *moves*. Examples of moves are λ-opt, Or-opt, exchanges, interchanges.

Limitation of local improving methods is that it can easily get trapped in *local minima*, i.e., solutions without any strictly better neighbour. Different methods have been proposed to push the search out of this trap. These methods can mainly be classified in three groups: trajectory-based method, population-based methods, and hybrid methods. In the first case, the search space is explored moving from one solution to a neighbour one, accepting moving toward worse solutions. Examples are tabu-search, simulated annealing, variable neighbourhood search among others. In the second case, new solutions are generated combining solutions selected from a pool. Examples are genetic and ant colony algorithms. Finally, hybrid methods combine characteristics of the previous algorithm to take advantage from the strength of each.

One objective of this thesis is to develop a tool able to provide the daily planning of vehicles operating in the complex system considered in MODUM. This implies to consider in the routing problem several side constraints as multi-trip possibility for the vehicles, pickups and deliveries, time windows, multiple depots, release dates on the merchandise. Moreover, we will have to deal with realistic data representative of the situation of a medium-large city. It is then reasonable to expect instances containing hundreds of customers and dozens of vehicles. In addition, the tool will need to promptly react to new service requests that can occur during operations.

For all these reasons, this thesis will focus on the development of *heuristic* algorithms for particular routing problems that arise in the urban context. We start by considering the Multi-Trip VRP, where a fleet of vehicles, based on a central and unique depot, need to serve a set of customers. Vehicles are allowed to be re-load when they go back to the depot and be re-routed. In a second step we consider a richer problem. In particular we associate with each customer a time window that represents the time interval where service should take place and we associate a release date with each merchandise. This release date represents the instant the merchandise becomes available at the depot. In a third step we consider the possibility of delivering different incompatible commodities. Commodities are incompatible when they cannot be transported into the same vehicle at the same time. On the other hand all the vehicles can transport all the commodities. In this step, the fleet size needs to be determined. In the last step, we consider the whole routing problem involved in MODUM.

A parallel work concerns the composition of two surveys. One is about routing problems in city centers the second is about Multi-Trip Vehicle Routing Problems. In the first survey we propose a classification and an analysis of the goods movement in an urban context. Then, we consider the major characteristics of routing problems encountered when delivering goods in cities. Peculiarities of each characteristic are highlighted. The survey on the Multi-Trip Vehicle Routing Problem proposes a full overview of research done on the subject.

The following section gives the scheme of the thesis listing all the chapters and the topics that are dealt.

Thesis structure

The thesis is made by eight chapters including this one. They are listed in the following with a brief explanation of the content.

Chapter 2: Vehicle Routing for City Logistics

The impact of merchandise transportation deeply impacts the life quality of inhabitants, especially in urban areas. In order to contribute to the understanding of good transportation, this paper gives a picture of nowadays urban good movements that are here classified and described.

We surveyed the papers that explicitly studied urban good transportation from the enterprise and from public authority point of view. From these papers, we identified the problems that scholars, privates and authorities face while studying urban deliveries or aspects that are considered when solutions are proposed to simultaneously improve delivery efficiency and city livability. Four main problems are identified: Time-dependent VRP, Multi-level VRP, Dynamic VRP and Multi-trip VRP. An overview on all these problems is then given. The paper concludes given directions for further research and development.

This chapter has been done in cooperation with Jesus Gonzalez-Feliu from Laboratoire d'Économie des Transports, Lyon, France.

Chapter 3: The Multi-Trip Vehicle Routing Problem: A Survey

This paper presents a survey on the Multi-Trip Vehicle Routing Problem. This problem is an extension of the well-known Vehicle Routing Problem, where vehicles are allowed to be re-loaded and re-routed once they end a trip at the depot. It was introduced more than 25 years ago, and since then, researchers have been working on it, but no extensive survey has been proposed.

The contribution of this paper is mainly to fill this gap, proposing a full collection of the works that have been done on the subject. The wide practical applications that are highlighted should encourage academics and practitioners to put their efforts on this problem in further research.

Chapter 4: A Memetic Algorithm for the Multi-Trip Vehicle Routing Problem

We consider the Multi-Trip Vehicle Routing Problem, in which a set of geographically scattered customers have to be served by a fleet of vehicles. Each vehicle can perform several trips during the working day. The objective is to minimize the total travel time while respecting temporal and capacity constraints.

The problem is particularly interesting in the city logistics context, where customers are located in city centers. Road and legal restrictions favor the use of small capacity vehicles to perform deliveries. This leads to trips much shorter than the working day. A vehicle can then go back to the depot and be re-loaded before starting another service trip.

We propose an hybrid genetic algorithm for the problem. Especially, we introduce a new local search operator based on the combination of standard VRP moves and swaps between trips. Our procedure is compared with those in the literature and it outperforms previous algorithms with respect to average solution quality. Moreover, a new feasible solution and many best known solutions are found.

This chapter has been done in cooperation with Thibaut Vidal from Université de Technologie de Troyes, Troyes, France and CIRRELT, Montreal, Canada.

Chapter 5: The Multi-Trip Vehicle Routing Problem With Time Windows and Release Dates

The Multi-Trip Vehicle Routing Problem with Time Windows and Release Dates is a variant of the Multi-Trip Vehicle Routing Problem where a time windows is associated with each customer and a *release date* is associated with each merchandise to be delivered at a certain client. The release date represents the moment the merchandise becomes available at the depot for final delivery.

The problem is relevant in city logistics context, where delivery systems based on city distribution centers (CDC) are studied. Trucks arrive at the CDC during the whole working day to deliver goods that are transferred to eco-friendly vehicles in charge of accomplish final deliveries to customers.

We propose a population-based algorithm for the problem based on giant tour representation of the chromosomes as well as a *split* procedure to obtain solutions from individuals.

Chapter 6: An Iterated Local Search for the Multi-Commodity Multi-Trip Vehicle Routing Problem with Time Windows

The Multi Commodity Multi-Trip Vehicle Routing Problem with Time Windows calls for the determination of a routing planning to serve a set of customers that require products belonging to *incompatible* commodities. Two commodities are incompatible if they cannot be transported together into the same vehicle. Vehicles are allowed to perform several trips during the working day. The objective is to minimize the number of used vehicles.

We propose an Iterated Local Search that outperforms the previous algorithm designed for the problem. Moreover, we conduct an analysis on the benefit that can be obtained introducing the multi-trip aspect at the fleet dimensioning level. Results on classical VRPTW instances show that, in some cases, the fleet can be halved.

This chapter has been done in cooperation with Daniele Vigo from the University of Bologna, Bologna, Italy.

Chapter 7: MODUM Vehicle Routing Problem

This chapter formally introduces and defines the MODUM problem. The main scheme of the simulator that is built to evaluate the performances of the system is given. A heuristic to face it is proposed as well. It will be embedded in a simulator.

The development of the simulator is managed by another partner of the MODUM project and is not presented in the thesis. As the development of the simulator and the data collection are not finished yet, no results are presented here.

This chapter has been done in cooperation with all the project partners.

Chapter 8: Conclusion and perspectives

This chapter concludes the thesis and proposes direction for further research.

Chapter 2

Vehicle Routing Problems for City Logistics

Abstract

The impact of merchandise transportation deeply impacts the life quality of inhabitants, especially in urban areas. In order to contribute to the understanding of good transportation, this paper gives a picture of nowadays urban good movements that are here classified and described.

We surveyed the papers that explicitly studied urban good transportation from the enterprise and from public authority point of view. From these papers, we identified the problems that scholars, privates and authorities face while studying urban deliveries or aspects that are considered when solutions are proposed to simultaneously improve delivery efficiency and city livability. Four main problems are identified: Time-dependent VRP, Multi-level VRP, Dynamic VRP and Multi-trip VRP. An overview on all these problems is then given. The paper concludes given directions for further research and development.

2.1 Introduction

Transportation has huge economic, social and environmental impacts. In 2009, 7% of the gross domestic product (GDP) in the EU was due to the transport industry that offered over 5% of total employment. Public revenues benefit as well from transportation: 0.6% of the GDP is collected from vehicle taxes and the biggest part of energy taxes (that counts 1.9% of the GDP) comes from taxes on fuel (European Commission [153]). Environmentally speaking, transportation has been the sector with the biggest growth rate of greenhouse gas (GHG) emissions compared to 1990 (European Commission [153]). 60% of the global oil consumption and 25% of energy consumption are due to transportation (Rodrigue [179]). Moreover, road transport caused 39,000 deaths in EU in 2008 (European Commission [153]).

On the other side, urbanization continues growing. In 2007, 85% of the EU's GDP was generated in urban areas where 72% of the European population lived (European Commission [151]). The urban population rate will keep growing and it is expected to reach 84% in 2050 (European Commission [153]).

Adding to these facts that nine out of ten Europeans believe that the traffic situation in their area should be improved (European Commission [151]), that 69% of road accidents occur in cities, that 25% of the CO_2 emission of the whole transport sector comes from urban transport (European Commission [154]), and that more than half of the weight of goods in road transport are moved over distances below 50 km and more than three quarters over distances below 150 km (White paper - European Commission [154]), we can easily understand the importance of transportation in urban areas for private people, public authorities and enterprises.

Especially, the nuisances of urban freight transport, mainly related to congestion, GHG emissions, pollution and noise, have become a priority for several public and private stakeholders. Since 1992 (Ogden [155]), many projects have studied the different aspects of urban goods movement (UGM) and logistics activities in urban areas. The aim of this survey is to analyze the vehicle routing problems (VRPs) that are faced when dealing with the transportation of goods in cities and to discuss the directions in which future researches should be conducted in this field.

In the survey we adopt the following methodology. In Section 2.2 we define the concept of city logistics, we introduce the actors involved and their interests, then we present the literature on vehicle routing optimization in cities. Section 2.3 classifies and analyzes the logistic flows in cities, and describes how vehicle routes are organized, providing some statistics on these routes.

The structure of the second part of the paper is more sequential. In Section 2.4, for each one of the major characteristics previously detected, we review the associated literature. We underline the main issues raised by the considered characteristic and present how they are addressed in the papers published on the subject. Section 2.4.1 deals with VRPs with time-dependent travel times. Section 2.4.2 considers multi-level VRPs. Section 2.4.3 is about dynamic VRPs. Section 2.4.4 reviews multi-trip VRPs.

In Section 2.5 we conclude the paper and open some perspectives on future works on route optimization in the context of urban good movements.

2.2 City logistics: definitions, stakes and routing problems

2.2.1 Scope

The scientific community has recently adopted two main concepts. The first is that of *city logistics* (Ruske [184], Kohler [122], Taniguchi et al. [200], [49]) defined by Taniguchi et al. [200] as "*the process for totally optimizing the logistics and transport activities by private companies in urban areas while considering the traffic environment, the traffic congestion and energy consumption within the framework of a market economy*". A much larger concept is that of *urban logistics* (Ambrosini et al. [4], Anderson et al. [5]), which includes all the organizational, behavioral, regulation and financing elements, as well as collaborative approaches, to study the logistics processes and the movements of goods and service flows in urban areas. Not only the retail distribution is concerned in that definition, but also shopping trips, civil works and city services' maintenance, among others.

In this survey, we will retain the more general framework and will consider as UGM all movements of goods related to urban logistics as defined above. In this sense UGM include all the existing flows of goods, from factories to wholesalers, from wholesalers to retail distribution, and also from shops to the households (Segalou et al. [189]). Especially, UGM include not only a large part of the commercial transport, a part of the individual transport, but it concerns also the service trips (of firms and of the individuals) and the flows generated by the supplying of the building sites and the maintenance of the networks (electricity, water, sewer system, routing of domestic waste). In the following of the paper, we will use the term *city* instead of *urban* when referring to logistics concept because it is generally adopted in the community.

2.2.2 The stakes of public authorities and private stakeholders

In the definitions of city and urban logistics given in Section 2.2.1, three main objectives can be identified. The first, and the most common one, is reducing congestion and increasing mobility of freight transportation services in urban areas. The second is to contribute positively to the environment and to the sustainable development, mainly by contributing to reach the Kyoto objectives in terms of GHG emissions, by reducing pollution and noise or by improving living conditions of city inhabitants. The last objective outlines how it is crucial to preserve city center activities (mainly commercial, tourist and tertiary).

However, since several stakeholders are seen in urban areas, the main goals of city and urban logistics differ depending on the actors involved (Taniguchi and Van Der Heijden [201]). Public authorities stakes are mainly related to collective utility. Their objectives can be in conflict with the individual performance and the goals of private stakeholders, who seek mainly to increase their economic benefits by both reducing their costs and ensuring a good service quality. For that reason, it is important that public and private stakeholders collaborate to urban logistics actions, in order to find consensual directions and increase the success rates of urban logistics projects. The main stakes are listed in the following.

- Regarding public authorities:
  - To revitalize the economic activity of the urban areas, particularly the town centers (with retail shops and various services);
  - To master urban sprawl, controlling strategic town planning;
  - To reduce congestion issues in the most dense (central) urban areas;
  - To decrease the impacts of nuisance due to GHG and other atmospheric pollutant emissions and to reduce noise levels;
  - To implement convenient good distribution services for the different socio-economic categories (like home deliveries for reduced mobility inhabitants or comfortable pickup points for very mobile inhabitants, among others).
- Regarding private stakeholders:
  - To provide quality services (commercial and tertiary activities) to the customers;

– To reduce the economic costs related to the last mile management;
– To make sure that sustainable policies of the firms coincide with urban transport practices.

In the next section, we describe the literature published on vehicle routing problems where the application to city logistics is explicitly underlined. We focus on papers that address collectivity attempts and papers based on company objectives. In both cases, vehicle route optimization can be a valuable tool.

2.2.3 VRP for city logistics

Collectivities can benefit from the use of vehicle routing optimization to validate policies to implement. For example, Taniguchi and van der Heijden [201] evaluates the benefit of the implementation of advanced information systems and cooperative freight transportation systems on CO_2 emissions of delivery trips.

Transportation optimization can lead to significant savings for the enterprises. These savings are quantified to be from 5% to 30% by Hasel and Kloster [103] when routing tools are used. Similar results are reported in Toth and Vigo [204] that estimate savings of 5%–20% when computerized planning is adopted. Remembering that transportation represents 10% of the final cost of goods (Rodrigue [179]), these figures show how potential gains offered by route optimization are important.

Before detailing the literature, we first shortly come back to the types of VRPs encountered in cities.

Routing problems in cities

Different transportation activities that occur in cities can be modeled using well-known routing problems. Those problems can be *tactical* or *operational*. Tactical can be considered problems where the planning is done once and then kept for a long period. For example, the postman performs each day the same trip, regardless the mail he has to deliver. He can only skip some stops if needed. At the operational level, customers served and (consequently) delivery trips can deeply differ from day to day. Route planning is carried out daily or even in real-time. The former is the case, for example of *express* parcel delivery. The latter, the case of *dynamic fleet management* where vehicles are dispatched in real-time for pickup or delivery purposes. At a tactical level, the quality of the solution obtained plays a major role. Since the plan is going to be maintained for a long period, one is willing to spend hours or even days to obtain a (near-) optimal solution. The use of exact methods can be considered. On the other side, at the operational level, time is prior. The quickness of the algorithm is a necessary condition, however quality needs to be guaranteed as well. Usually, (meta-) heuristics are preferred.

Collectivities

Urban goods transport is a vital activity for cities, but it is seen as a nuisance for the livability of the city itself. Trucks use the same road network as public transport and private cars, contributing to traffic congestion, air and noise pollution, road accidents. Municipalities can introduce regulations that aim to maintain the living environment in urban areas while facilitating smooth and safe traffic flows. Such measures can be, for example, the introduction of loading/unloading zones, the banning of freight vehicles in cities during night or their restriction of movement according to time, size or weight (OECD [81], Quak and de Koster [172]).

Access limit could be modeled using time windows (TWs) that differs from those in VRPTW, because they restrict the access to the overall concerned area (usually called restricted zone). Hence, they do not concern delivery time but the time vehicles have access to the zone. Also, as they are not imposed by customers but by local authorities, they do not depend on the customer. The introduction of these TWs is motivated by seeking to reduce congestion and pollution in central areas, but imposes extra cost to the carriers, forcing them to use more vehicles (Muñuzuri et al. [144], Muñuzuri et al. [143]). When TW imposed by municipalities are considered, the problem that arises is called Vehicle Routing Problem with Access Time Windows (VRPATW).

Limited access could be combined with the construction of a city distribution center (CDC) where goods are loaded in appropriate vehicles for final delivery. CDC are implemented in most of European cities with contrasting results (OECD [81], van Rooijen and Quak [211], Browne et al. [30]). Transshipment in the outskirts of the city can avoid big trucks from entering city centers. Moreover, small vehicles can be fully loaded in the CDC decreasing the number of (almost) empty or not fully loaded trips. From the other side, however, more friendly vehicles can be needed to replace trucks. Crainic et al. [54] study how the use of CDC can reduce global costs comparing VRP solution costs with two-echelon VRP (2E-VRP, introduced in Section 2.4.2) solution costs. Results show the potential benefit of using CDCs, especially when CDCs are judiciously located.

Quak and de Koster [172] study the impact of TW and vehicle restriction on companies, showing that those policies increase companies' costs and pollutant emissions.

Crainic et al. [55] study the possibility of using CDC for consolidate goods to be delivered in the center of Rome. Trucks would deliver merchandise to CDC that is consolidated into *city-freighters* limiting direct distribution to customers. Experiments show that it would reduce the number of truck-kilometers in the city center by about 70%. These saved kilometers are replaced by city freighter moves that are expected to be more environmental friendly.

A different policy is to distribute financial incentives to customers in order to foster off-hour deliveries (Silas et al. [191]). It is outlined how the potential shift of New York City to off-hour deliveries (that is around 20%–40%) can lead to economic savings in the range of $100–$200 million per year, while reducing congestion and environmental pollution.

Companies

Delivering merchandise in urban areas involves facing rush hours and congestion, unexpected events like accidents, vehicle crashes and changes in weather condition. Moreover, companies have to respect all the restrictions that local authorities introduce to maintain the livability of the city. Very often these measures are not harmonized among cities, causing inefficiencies and planning difficulties (OECD [81], Muñuzuri et al. [143]).

Planners should take into account the strong relationship in urban areas between time of the day and travel times. Considering time-dependent travel times is important to better plan vehicle routes (Taniguchi and Shimamoto [199], Ando and Taniguchi [6], Ehmke and Mattfeld [69], Kritzinger et al. [125]). Moreover, considering time-dependent travel times can reduce CO_2 emissions and time windows violations (Figliozzi [75]).

To consider time-dependent travel times, the planner needs information on the expected situation of the road-network. Data can be provided by a taxi fleet (Kritzinger et al. [125], Ehmke and Mattfeld [68], Ehmke et al. [70]) or could be collected from Google Maps and the use of highway sensors to improve the quality (Conrad and Figliozzi [44]). For sensor location problems on traffic networks, we refer to Gentili and Mirchandani [87]. If interested on how time-dependent travel times for city logistics application are collected we refer to Ehmke and Mattfeld [68], Ehmke et al. [70].

Monitoring the fleet during the working time gives the possibility to handle unforeseen events with intelligent re-routing of vehicles (Zeimpekis and Giaglis [220], Novaes et al. [150], Qureshi et al. [174]). A prerequisite for these systems is possible communication between drivers and a central unit and the use of new technologies such as global positioning system (GPS) or Intelligent Traffic System (ITS). Project SMILE ([152]), implemented in a courier company working in Malmö, Sweden, has shown how unloaded trips decrease and the number of delivers per trip increases when the transport monitoring center knows the real-time location of all vehicles.

Cooperative distribution systems can improve the quality of the service and reduce the cost for the carriers (Thompson and Hassall [203]). Consolidation of goods from different carriers in shared vehicles, that can be those owned by the carriers (Qureshi and Hanaoka [173]), leads to a higher load factor and thus enables decreasing the number of empty trips and the driven kilometers. In the Netherlands and Belgium a collaborative system is implemented with 16 companies (TransMission). Quak [171] outline that without that collaborative system in the city of Amsterdam four times more trucks would be needed. That could have a good impact on the quantity of CO_2 emission.

The structure of city centers in most of European cities is inherited from Middle Ages: streets are narrow with no or few parking lots and are often one-way (Crainic et al. [55], Muñuzuri et al. [143]). Then, trucks cannot get into centers due to structural limitations (different than limitations introduced by authorities discussed in Section 2.2.3) and small vehicles must be used. Due to their limited capacity, these vehicles go back to their depot several times a day and are reloaded for new tours (Browne et al. [29], Delaître and De Barbeyrac [60]).

Browne et al. [29] presented the case of supply company operating in the City of London. The company studied the possible benefit of using a micro-consolidation urban center together with electrically assisted cargo tricycles and electric vans instead of a suburban depot (located 29 km away from the London suburbs) and 3.5 tons gross weight diesel vans. From the original suburban depot only one truck is going to the micro-consolidation urban center instead of being the starting point of all the diesel van delivering routes. Due to the small size of tricycles and electric vans, they perform several trips during each day. The analysis of the system outlined the reduction of the total distance traveled and CO_2 emissions per parcel delivered (respectively by 20% and 54%) for the same operating cost.

Finally, Chang and Yen [36] consider the case of a city-courier company in Taipei and considered strict-TW since there is no-parking possibility along most city streets. Moreover, waiting is not allowed on the whole network.

2.3 Urban Goods Movements

In this section we illustrate the Urban Good Movements (UGM). As already introduced in Section 2.2.1, the UGM that we consider in this survey include all goods' flows as well as service, supply and maintenance trips. Section 2.3.1 proposes a detailed classification of the UGM, while in Section 2.3.2 UGM are statistically analyzed.

2.3.1 UGM classification

A detailed classification of the movements (following the definition of Segalou et al. [189]) is given in Figure 2.1 to clearly understand what exactly are UGM. In this figure, the first column exhibits the different types of UGM. A second column illustrates these UGM with some examples. A third column describes the type of flow. A fourth column finally indicates the general modeling framework used for each type of UGM.

This classification is discussed below:

- Inter-establishment movements (IEM): IEM are pickup and delivery trips related to the urban area's economic activities. Three main organizational modes exist for these movements (Routhier and Toilier [182]):
  - Third party transport: Transport is carried out by a third-party service provider. Two main strategies are identified here: full truckload (FTL) or less-than-truckload (LTL). FTL approaches mainly concerns hypermarket distribution, urban industry and agriculture. LTL is related to retailing and tertiary activities distribution. Typical examples are parcel delivery services, express delivery services, supermarket and medium stores distribution, but also distribution for restaurants, hotels, food franchising, clothing retailers and so on. Other activities, like non-hypermarket grocery, wholesalers and pallet distribution companies can also combine FTL and LTL transport schemes. FTL schemes are in general modeled by the well-known Transportation Problem (TP), introduced by Hitchcock [108], and LTL schemes are in general modeled as VRPs.
  - Sender's own account: This category refers to transport flows carried out directly by producers, artisans, craftsmen or distribution companies without involving a

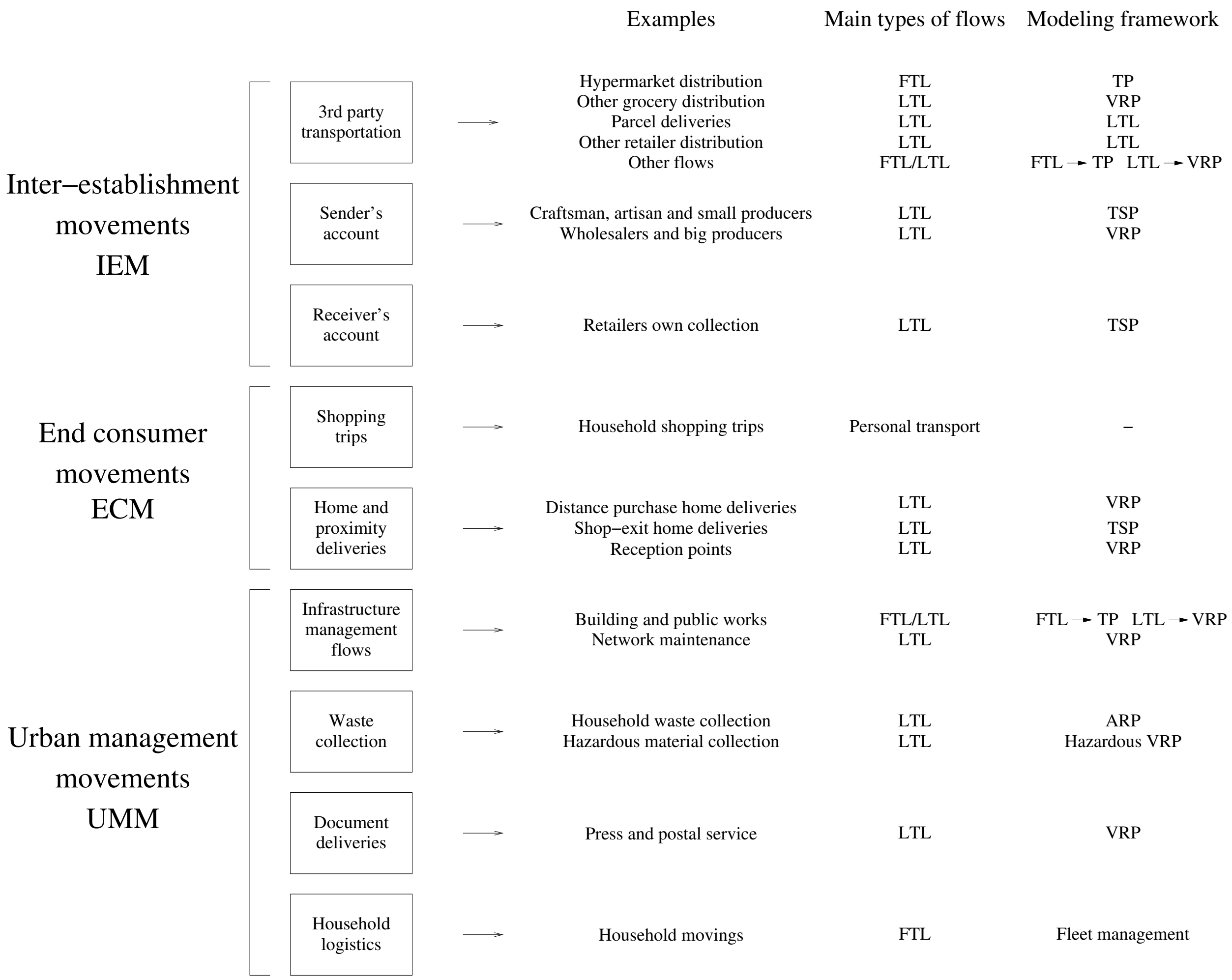


Figure 2.1: UGM classification

transport carrier. Concerned routes are in general similar to small LTL circuits, although they are under-optimized with respect to third party transport services and have only one departure point (consolidation or multiple depot approaches are less deployed in own account transport). Note that craftsmen and small enterprises have often one single vehicle. In this case, transportation schemes can be modeled as traveling salesman problems (TSPs). For medium and large companies with several vehicles, VRP models apply.

  – Receiver's own account: Transport for this category is performed by the receiver. Examples are the distribution companies collecting goods at their suppliers, or the retailers going to gross companies for their supply needs. Again, these flows are in general mono-vehicle. They are assimilated to pickup routes, easily identifiable as TSP routes. They additionally have the following common characteristic: purchased goods are selected at the supplier location, which implies important pickup times at each step of the routes.

- End-consumer movements (ECM): ECM refer to the trips that permit the (physical) junction of the goods and of the consumers. They consist of two main categories:
  – Shopping trips: Traditionally, ECM flows were reduced to shopping trip chains made by private cars (Gonzalez-Feliu et al. [93]). Individually, these flows are not formally optimized since they derive from behavioral patterns. Globally, they are modeled with classical four-steps models (Ortúzar and Willumsen [157]).
  – Home and proximity deliveries: Home delivery includes business-to-customer flows from the parcel delivery sector (for example e-commerce shopping) and other types of deliveries like grocery home deliveries. For the former, LTL third-party transport is used. In both cases, VRP routes have to be optimized. When goods are physically purchased at stores and delivered home, distribution rather takes the form of a TSP. Proximity delivery relies on proximity reception points where goods are delivered. Again, they are organized via VRP, with the trend that the number of delivery points is decreased thanks to the customer aggregation provoked by the reception point network.
- Urban management movements (UMM): UMM flows are related to the development of a city, public maintenance and other functional needs of the city. They are of various nature and characteristics, and can be grouped into four main sub-families:
  – Infrastructure management flows: These flow derive from building and public works. They are non-periodic, non-systematic flows which take place in different parts of urban areas and depend strongly on the urban planning policies. Most of them are FTL flows when related to building and public works, although some LTL routes can be defined (remember that FTL flows are generally modeled via TP while LTL via VRP). Network maintenance (phone, electricity, water, optic fiber, etc.) is another example, which is in general modeled as a VRP.
  – Waste collection flows: These flows concern garbage collection for individuals and professionals. Flows are organized differently depending on the type of waste (household, recyclable, hazardous, etc.). For household waste collection, arc routing models are generally used (Del Pia and Filippi [59]) so as to aggregate collection points located in a same street and that will be served successively. Garbage

collection for professionals or for recyclable waste are organized as LTL routes. They can be sub-contracted to companies specialized in reverse logistics, that might be able to mutualize waste collection with deliveries. Models can then be complex pick-up and delivery VRPs. Transportation of hazardous waste also gives raise to specific VRPs, with risk minimization objectives (Tarantilis and Kiranoudis [202]).

- Document deliveries: This type of flows include press and postal services. They can be modeled as VRP (or arc routing problems, ARP, for postal services) with a large number of customers. Routes are generally stable over relatively long periods of times (except that customers might possibly be skipped). However difficulties stem from the huge variability of the quantities to be delivered to customers, especially in the context of press distribution.
- Household move logistics: These flows are provoked by individual moves. They are very heterogeneous and difficult to anticipate. They are mainly FTL routes or FTL shuttles (when for an intra-city move, a truck makes more than one trip to relocate all goods). The most important issues here are related to fleet management rather than vehicle dispatching.

2.3.2 Statistical analysis of UGM flows

In this section, we provide some statistics on UGM flows. The objective is to give some insights on the relative importance of the different categories of flows in terms of volume in the city. Also, information is given on the size of the routes and, thus, the characteristics of the solutions that can be expected from vehicle route optimization.

Table 2.1 compares IEM, ECM and UMM on the basis of road occupancy. All ratios are extracted from Segalou et al. [189], Bonnafous [23] and Gonzalez-Feliu et al. [93].

	Traffic road occupancy	Park road occupancy	
		Whole urban area	Inner city
IEM	40-45%	23%	62%
ECM	45-55%	67%	31%
UMM	8-10%	10%	7%

Table 2.1: Road occupancy rates

Traffic road occupancy rates are indicators that reflect the occupancy of infrastructures by running vehicles. They are expressed in km $\times$ equivalent vehicle (generally car-equivalent units). Impacts on road traffic of good movements are mainly shared between IEM and ECM, with more or less equivalent rates. ECM traffic is essentially provoked by shopping trips, with private cars. Other types of ECM flows are indeed more limited (see, *e.g.,* Durand and Gonzalez-Feliu [64]). UMM traffic road occupancy rate is less than 10%.

Park road occupancy rates are indicators that reflect the occupancy of infrastructures by vehicles when they stop. They are expressed in hours $\times$ equivalent vehicle. They include and do not differentiate among the different types of parking (on delivery areas, on the street network, etc.). The issue of park road occupancy is essentially related to ECM (shopping trips) in the whole urban area. However, when focusing on city centers, the question of parking for IEM becomes prominent.

Tables 2.2–2.4 focus on IEM flows. As stated before, these flows represent the most important part of the UGM once shopping trips (that cannot be addressed with route optimization strategies) are removed. The data shown in these tables are new statistics retrieved from the French National Surveys on Urban Goods Movement database (Routhier [181], Patier and Routhier [159]). The database compiles three surveys made in late 90's in France, respectively for the urban areas of Bordeaux, Marseille and Dijon. These surveys contain three nested questionnaires: one at the establishment level, one at the operation level and one at the vehicle level. We exploit here this last set of data, which was less explored so far. This set however suffers from certain lacks: from about 2100 collected routes, about two thirds of them cannot be exploited because of missing details or since they concern marginal types of goods (*e.g.*, cattle transportation). The size of the sample is thus not sufficient to be too affirmative on the results. For this reason, we limit our analyses here to the main trends that can be observed.

From the selected route dataset, four categories of transport have been defined, according to French practice: third-party transport is divided in two categories (classical delivery services and small parcel delivery services), as well as own-account transport (sender's and receiver's own account transport). Categories of routes are defined according to their number of deliveries, with 5 categories: 1 delivery, 2 to 10 deliveries, 11 to 20, 21 to 30 and more than 31 deliveries. Tables 2.2–2.4 display information on the percentages involved with each category of transport and route with regard to the number of deliveries, the number of routes and the weight delivered, respectively. In these tables "-" indicates that no route included in the dataset fits the category.

	Third-party transport		Own account		
Route size category (in number of deliveries)	Classical delivery service	Small parcel delivery service	Sender	Receiver	Total
1 delivery (FTL routes)	0.21%	-	0.34%	0.07%	0.62%
2 to 10 deliveries	2.78%	0.63%	5.77%	0.11%	9.22%
11 to 20 deliveries	4.36%	3.92%	9.24%	0.29%	17.80%
21 to 30 deliveries	4.92%	6.40%	6.40%	-	17.72%
31 deliveries and more	5.79%	47.30%	1.48%	-	54.46%
Total	18.05%	58.25%	23.23%	0.47%	100.00%
	76.30%		23.70%		

Table 2.2: Percentage of deliveries with respect to the total number of deliveries

A first conclusion that can be drawn from Tables 2.2–2.4 is that third-party transport manages about 75% of IEM in terms of number of deliveries, and aroud 60% in terms of number of routes and weight transported. The rest part of IEM are moslty due to sender own account, while receiver own account plays a marginal role.

It can be observed that classical and small parcel delivery services are almost equivalent in terms of routes generated, but they are unbalanced with respect to the number of deliveries and weight. Moreover, the unbalance reverses. As suggested by the name of the categories, this is explained by the fact that parcels delivered by small parcel delivery system weigh less than those transported by classical service. Also, small parcel delivery involves route

	Third-party transport		Own account		
Route size category (in number of deliveries)	Classical delivery service	Small parcel delivery service	Sender	Receiver	Total
1 delivery (FTL routes)	4.01%	-	4.66%	1.29%	9.96%
2 to 10 deliveries	9.18%	1.55%	18.50%	0.65%	29.88%
11 to 20 deliveries	5.43%	4.79%	11.13%	0.50%	21.85%
21 to 30 deliveries	3.75%	4.79%	4.79%	-	13.33%
31 deliveries and more	2.98%	21.22%	0.78%	-	24.98%
Total	25.35%	32.35%	39.86%	2.44%	100.00%
	57.70%		42.30%		

Table 2.3: Percentage of routes with respect to the total number of routes

	Third-party transport		Own account		
Route size category (in number of deliveries)	Classical delivery service	Small parcel delivery service	Sender	Receiver	Total
1 delivery (FTL routes)	6.70%	-	6.86%	2.16%	15.71%
2 to 10 deliveries	31.21%	0.13%	16.58%	0.85%	48.77%
11 to 20 deliveries	16.91%	0.52%	7.77%	0.13%	25.32%
21 to 30 deliveries	2.45%	0.58%	1.52%	-	4.55%
31 deliveries and more	2.42%	3.18%	0.05%	-	5.65%
Total	59.70%	4.41%	32.76%	3.13%	100.00%
	64.11%		35.89%		

Table 2.4: Percentage of weight delivered with respect to the total delivered weight

with much more stops in average than classical delivery services. Hence, more than 50% of the weight delivered in cities originates from classical third-party logistics with less-than-20-deliveries routes, representing less than 20% of the routes and 10% of the deliveries. Conversely, more than 50% of the deliveries comes from small parcel delivery with more-than-20-deliveries routes, while representing less than 5% of the total weight.

Receiver's own account represents only a very small percentage of the goods transported in the city. Routes involve a limited number of stops, with a single customer visit in more than 50% of the cases.

Regarding sender's own account, as already mentioned, it represents about 40% of the routes. The structure of these routes approximately matches the one of classical third-party: most routes imply between 2 and 20 deliveries. However, contrary to third-party, the chances that these routes are finely optimized are limited. One can suspect that place for a lot of optimization is left here.

Papers surveyed in Sections 2.2.3–2.2.3 mostly refer to third party transportation. The analysis conducted justifies researcher's behavior to focus on that specific transportation segment. Indeed, third party transportation is part of the IEM (Section 2.3.1) that counts for 40-45% of traffic road occupancy and for 62% of park road occupancy in the inner city. The other relevant part of movements with respect to traffic road occupancy is due to ECM that contain all the shopping trips that are made by privates. Consequently these movements are out of optimization possibilities. Moreover, third party transportation is prominent into IEM regarding all the considered classifications (see last row in Tables 2.2–2.4).

It follows that optimization of third party transportation can definitely lead to traffic improvements, congestion and pollution reduction, and cost savings. These are some of the stakes we listed in Section 2.2.2 for public authorities and privates. Related routing problems are studied by scholars in order to carry out optimized planning as well as validate logistic measures. These problems are examined in the next section.

2.3.3 UMG as routing problems

UGM listed above, can be represented using classical routing problems that are widely studied in literature. IEM can be modeled with various vehicle routing problems (VRP, Toth and Vigo [204], Golden et al. [90]) as well as ECM not made by private cars. If goods need to be picked-up and delivered it can be the case of the VRP with pickup and deliveries (VRPPD) or the VRP with backhauls (VRPB). In the former case, each request is characterized by a pickup location and a delivery location (and eventually the weight or size of the request). In the latter, vehicles first deliver merchandise stored at the depot to customers. Afterwards, they can pickup other goods to bring back to the depot. The multi-depot VRP (MDVRP) and the periodic VRP (PVRP) can arise in this context as well. In the MDVRP, vehicles are located at different depots, while in the PVRP customers need to be visited up to n times during a period of n time units (once per day, twice per week, etc.).

Sender's and receiver's own account movements as well as ECM made by private cars can be associated with traveling salesman problems (TSP, Gutin and Punnen [99]). The sender (receiver) accomplishes the deliveries (pickups) with his own vehicle. The private person as well, gets his own car to visit all the stores he needs. Most of the time trips are under-optimized.

Finally, waste collection and postal service can be modeled as arc routing problems (ARP). In the ARP, the network is still represented as a graph, but a service is expected on arcs instead of nodes. It can be noticed that these problems are tactical. Since they correspond to some specific categories of routing problems, that have their own literature, they will be ignored in this survey. The reader interested to the ARP is referred to Dror [63], Wøhlk [217].

2.3.4 Discussion

From the above sections, one can see the high importance, the huge variety and the originality of vehicle routing in cities. Vehicle routing is faced for a large part of UGM. Very different products, organizations and time-scales are concerned. The literature on the subject strongly reflects this variety. Case-study papers address a set of very different topics, with different organizational or technological prerequisites.

It is a general opinion that a complete organization and control of UGM into cities can enhance livability as well as transportation efficiency, possibly satisfying both privates and public interests (Section 2.2.2). Recent works (see the MODUM[1] project for an example) study different delivery system and the research process is going toward a scheme where the whole stream of merchandise is controlled by a central management center. Goods are first

[1] `http://www-lipn.univ-paris13.fr/modum`

delivered to CDC located around the city. This is done in order to forbid heavy trucks to enter into the city center seeking for pollution, noise and congestion reduction. Merchandise is then loaded into electrical or eco-friendly vehicles. Limited autonomy as well as city center's configuration force the use of small vehicles that naturally perform several serving trips during the working day. High quality service can be obtained carrying out optimization that considers time-dependent travelling times in order to avoid congested zones in the city as well as dynamic re-optimization that takes into account unexpected events and new requests.

Analyzing the described scheme and the papers previously cited, one can extract different categories of vehicle routing problems that appear significant for city logistics and relevant from an academic point of view. They are listed in the following.

A first noticeable characteristics is the time-dependency. Efficient transportation in urban areas should consider traffic congestion and rush hours, even if many cities reflect on how UGMs could be transferred to more quiet hours (*e.g.*, at night).

A second important topic is multi-level distribution. Many papers highlight how new distribution schemes based on distribution centers at the outskirts of the city and satellites inside the city could limit nuisances and costs of UGM.

A third subject is related to the transition from large trucks to small environmentally-friendly vehicles. A direct consequence of using small vehicles is a decrease of route sizes, which implies both quick access to the customers from the depots (which relates to the use of multi-level distribution or the move of distribution centers towards city centers) and multiple returns of the vehicles to their depots during the day. In terms of vehicle routing, the latter refers to the class of multi-trip VRPs.

Finally, the dynamics of the cities and the development of new communicating technologies motivate the study of dynamic VRPs, where vehicle routes can be re-optimized according to different types of information (actual travel times, new requests, unexpected events).

In the second part of this paper, we will separately consider each one of these four families of vehicle routing problems and review them. In each case, we will enlarge the scope and also consider papers not directly connected to city logistics. We will describe which additional difficulties are implied and how they are handled in the literature. We start in Section 2.4.1 with VRPs with time-dependent travel times. Section 2.4.2 continues with multi-level VRPs. Section 2.4.3 is about dynamic VRPs. Section 2.4.4 completes the review considering multi-trip VRPs.

2.4 Prominent families of vehicle routing problems for urban good movements

The four families of variants of VRPs surveyed in this section strongly differ in nature. Multi-level VRPs assume a different design of the distribution scheme, with additional facilities and transshipments. Multi-trip VRPs rather consider a different organization of the routing. No facilities are introduced; instead, the daily management of the fleet is modified. For VRPs with time-dependent travel times, the novelty concerns the sharpness of the modeling. The physical organization is the same as in standard VRPs but models better capture how routing will actually be performed. Finally, with dynamic VRPs, the optimization setting changes as data are revealed in the course of time.

A first consequence of these differences is that the four families are not exclusive. Actually, all combinations are possible. Hence, one should have in mind while reading this section that the issues are treated separately, but might be faced simultaneously. Also, the literature devoted to the different families sometimes intersects and it is possible that some papers appear twice or more.

A second consequence is that the topics that are worth being underlined are not the same for the different families. For this reason, the structure of the four subsections will vary a lot.

2.4.1 VRPs with time-dependent travel times

Traditional vehicle routing problems consider that travel times between locations depend only on distances, while in practice they usually (and especially in an urban setting) continuously vary during the day. Addressing time-dependent travel times raises different issues that are developed here.

Motivation

A first important question is: does it worth it to consider time-dependent travel times? The methodology used to answer this question is to compute two series of solutions: solutions obtained with a classic VRP formulation and solutions where time-dependent travel times are taken into account in the solution framework. Then, both types of solutions can be evaluated using time-dependent data and compared. Several papers applied this methodology.

In the context of the distribution system of an electrical goods wholesaler in South West UK, Maden et al. [138] outlined that 65% of the routes obtained considering constant travel times (with no congestion) would exceed the 10 hours maximum working time, requiring averagely 57 minutes of extra time. Even reducing the average speed by 10% in the model produces 44% of routes that become infeasible in the time-dependent context. It is also observed that considering time-dependent travel times leads to saving 7% of CO_2 emissions.

A case study from a Taiwanese company is faced in Kuo et al. [127]. The company has to serve 25 retail stores located nearby a depot in the south of the country. Solutions obtained considering constant speeds and evaluated using time-dependent travel speed, decreases operation time up to 16.77% compared to the company's planning strategy. A further 6% can be saved by considering time-dependent speeds during the search.

Kuo [126] studied the fuel consumption when time-dependent travel times are taken into account. The time dependency of travel times is reflected in the fuel consumption that is time-dependent as well. It is pointed out how minimizing fuel consumption increases transportation time between 21.96% and 23.37% and transportation distance between 37.96% and 40.26%. From the other side, fuel consumption is reduced by 22.69% and 24.61%.

Ehmke et al. [71] studied the impact of time-dependent travel times in city distribution planning. It is underlined that non time-dependent routing can underestimate or overestimate total travel time, introducing an error in the range of ±20%. Underestimation, in particular, occur when routing is performed during the day.

Data collection and management

Until the 90s, time-dependent travel times (or speeds) were not addressed for two main reasons: parameter estimation efforts and data storage requirements were prohibitive for computers at that time (Hill and Benton [107]).

Data collection is still a very hard job: the rush hours can be known, but different arcs of the network can be affected quite differently (Fleischmann et al. [79]). Advancements in technology allow easier data collection: mobile phone moves or sensors installed on taxi fleets for example can be tracked, thus providing real-time information and sources for historical data on traffic (Ehmke [66], Fleischmann et al. [79]). However, having a precise view on the traffic for the whole network is still a challenge.

The management of the data also still raises difficulties. van Woensel et al. [212] use a queueing approach to get time-dependent travel speeds from traffic flow data collected by the Flemish government. Ehmke et al. [71] discuss how storage and access to data can be improved by clustering network segments into homogeneous groups according to their relative variation of daily speeds. Also, Ehmke [67] explain how including the time dimension in the input data affect the computation of the distance-matrix, the algorithms used for the calculation of the shortest path between locations and the network representation in terms of graphs.

Modeling of travel times

Travelling times usually vary continuously, but for modeling purpose the planning horizon is discretized in time slots. These time slots can have different length (Fleischmann et al. [79]) and to each slot can be assigned a speed (Ichoua et al. [112], Donati et al. [61]) or a travel time (Fleischmann et al. [79]).

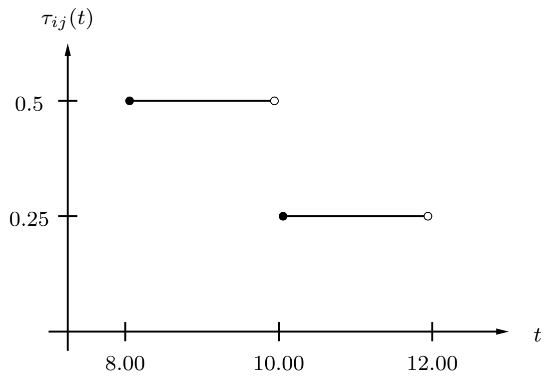
Hill and Benton [107], model the time-dependency by assigning a speed to each node for each time period. That speed can be seen as the average speed for the area around the location during a period t. Thus, the traveling time through an edge can be estimated using the speed assigned to the incident nodes. In Malandraki and Daskin [139] the travel time is a step function of the time over the planning horizon.

In both previous works, the First In First Out (FIFO) property (called as well non-passing property) does not hold. FIFO property, firstly defined in Ahn and Shin [2], can be stated as follows. A function $A_{ij}(t)$ is defined as the arrival time at node j when service starts at time t at node i and arc (i, j) is covered. If for each pair of nodes i and j and any two service start times t_1 and t_2 such that $t_1 > t_2$, $A_{ij}(t_1) > A_{ij}(t_2)$ holds, then the FIFO property holds. Roughly speaking, identical vehicles covering an edge (i, j) must reach location j in the order they leave location i.

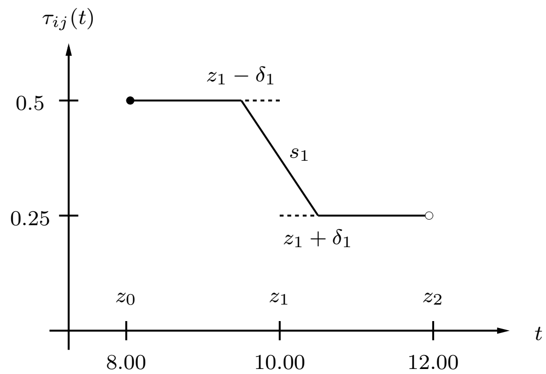
It seems natural to expect from a modeling that the FIFO property holds. However, one should be conscious that it is not necessarily true when data are dynamic (which is not the case usually: travel times are varying but are assumed to be known). Indeed, arcs represent shortest paths in the street network and it is possible that two vehicles following the same arc do not travel through the same streets. Thus, a vehicle leaving earlier a node i might sometimes be stuck in an unexpected congestion while a vehicle departing later will be able to avoid it.

The example of Figure 2.2a) depicts a situation where the FIFO property does not hold. In this example, travel times are modeled with step functions, as in Malandraki and Daskin [139]. Starting from i between 8.00 a.m. and 10.00 a.m. takes 30 minutes while leaving i between 10.00 a.m. and 12.00 takes 15 minutes. In this case, a vehicle leaving i at 9.55 will reach j at 10.25, while a vehicle leaving i at 10.05 will reach j at 10.20. Different modelings have been proposed that ensure the FIFO property.

Fleischmann et al. [79] propose a method to obtain smoothed travel time functions satisfying FIFO property from step functions. The jumps between time slots $Z_k = [z_{k-1}, z_k[$ are linearized in an interval $[z_k - \delta_k, z_k + \delta_k]$ with appropriate parameter δ_k and slope s_k. They demonstrate that if *1)* $\delta_k > 0$, *2)* the intervals do not overlap, and *3)* the slope $s_k > -1$, the non-passing property holds. In Figure 2.2b, travel time illustrated in Figure 2.2a are smoothed accordingly. The vehicle that leaves location i at 9.55 arrives at j around 10.18, while the vehicle that leaves location i at 10.05 arrives at j at 10.26 ($z_1 = 10.00$, $\delta_1 = 0.5$, $s_1 = -0.25$). FIFO property is respected.



(a) FIFO property does not hold



(b) Traveling times smoothed as proposed by Fleischmann et al. [79]: FIFO property holds

Figure 2.2: Time-dependent travel time $\tau_{ij}(t)$ for an edge (i, j). Time and traveling time in hours

Impact on solution methods

Algorithms designed for the VRP cannot straightforwardly be adapted to the TDVRP: local changes on a route have complex consequences on the whole planning of the route. Ahn and Shin [2] however explain how FIFO property can help. Assuming FIFO property, arrival time function A_{ij} is monotonic and so it possesses an inverse A_{ij}^{-1}. Thus, given a feasible route, the latest time at which a customer service can start keeping feasibility, can be linearly calculated using backward relations. Then, checking feasibility for inserting unrouted nodes, combining distinct routes and exchanging nodes can be quickly done.

Another difficulty arises from the fact that the TDVRP is an asymmetric problem (driving

Authors	year
Ahn and Shin [2]	1991
Hill and Benton [107]	1992
Malandraki and Daskin [139]	1992
Kaufman and Smith [118]	1993
Horn [110]	2000
Ichoua, Gendreau and Potvin [112]	2003
Fleischmann, Gietz and Gnutzmann [79]	2004
Woensel, Kerbache, Peremans, and Vandaele [212]	2008
Hashimoto, Yagiura and Ibaraki [102]	2008
Donati, Montemanni, Casagrande, Rizzoli and Gambardella [61]	2008
Figliozzi [74]	2009
Soler, Albiach and Martínez [195]	2009
Kuo, Wang, and Chuang [127]	2009
Kok, Hans and Shutten [123]	2009
Maden, Eglese and Black [138]	2010
Kok, Hans, Shutten and Zijm. [124]	2010
Figliozzi [76]	2012
Ehmke, Steinert and Mattfeld [71]	2012

Table 2.5: Articles cited that concern TDVRP

to the city center could take more than driving from the city center along the same road in the morning, and the opposite could happen during evening, Kok et al. [123]). Local search moves as $k-opt$ exchange or insertion heuristic have to be carefully designed in this context. For example, the $k - opt$ exchange tries to improve a solution by exchanging k links while ensuring feasibility. When the method is applied to the VRP, only the cost of the exchanged links has to be taken into consideration. Conversely, when the TDVRP is considered, the cost of more links must be taken into account. That happens both because some links are covered in the opposite sense and because the start times changed (Malandraki and Daskin [139]).

Finally, differently than in the classic VRP, leaving the depot as soon as possible is not necessarily the optimal choice. When driving time is considered in the problem (in the objective or because it is constrained), time savings can be obtained by simply waiting at the depot. This introduces a complication to the problem: the *optimal start time problem* (Hashimoto et al. [102]) for each vehicle needs to be solved and can be modified by any change of the routes.

Based on the previous ingredients, different heuristics and meta-heuristics have been proposed to solve the problem: a modified Clark Wright and 2-opt heuristic (Hill and Benton [107]), nearest-neighbor heuristic and a cutting plane heuristic (Malandraki and Daskin [139]), iterated LS (Hashimoto et al. [102]), simulated annealing (Kuo [126]), multi ant colony system (Donati et al. [61]), tabu search (Maden et al. [138], Kuo et al. [127]), parallel tabu search (Ichoua et al. [112]). Moreover, in Ehmke et al. [71] classical heuristics for the TSP are adapted to the TDTSP and their performances are compared. Another direction explored by Soler et al. [195] is to get rid of travel-times variations by adapting the graph structure. They show the TDVRP with TW can be transformed into a Asymmetric Capacitated VRP (without TW). Table 2.5 lists the papers cited in this section.

2.4.2 Multi-level Vehicle Routing Problem

Traditional routing problems consider direct delivery from a central depot to a set of customers. Recently, mostly in city logistics context, different distribution systems have been studied where goods are dispatched to intermediate depots before reaching the final destinations. Research has mostly focused on rather simplified systems and it is here summarized.

Motivation

In the last years, enterprises understood the need to redesign their distribution scheme in cities mainly for two reasons. Firstly, in order to improve the quality of their service, to better match customer's requests, to improve their image and eventually to be more effective. Secondly, traffic restrictions have been introduced in most of the big European cities. Such limitations, for example, avoid big trucks to enter city centers, limit the number of trips a vehicle is allowed to do in special zones and encourage the use of green or environmental friendly vehicles. Enterprises then need to adapt to this new environment or to anticipate possible future restrictions. Moreover, municipalities have to face constant growth of urban freight transportation, related to the increment in urban population, that contributes in traffic congestion, air pollution, noise pollution and accidents into city centers. Solution to those problems can be found implementing city distribution centers (CDC), where *transshipment*, *consolidation collaboration* and *synchronization* are considered.

Following Rushton et al. [183], a CDC can be a finished goods warehouse that holds stock from factories, a transshipment site or cross-docking facility (freight is unloaded from inbound vehicles and directly loaded in outbound vehicles and no storage is offered, see Van Belle et al. [209] for a survey on cross-docking). In city logistics context, the term *satellite* is normally used to indicate a small CDC where freight can be stored only for a short period (or cannot be stored at all) and vehicle-waiting is not offered. Then synchronization of the inbound and outbound flows is a crucial point. In particular, a satellite can be a public parking or municipal bus garage (Crainic [49]).

From a city logistics point of view, CDCs are installed to achieve high degree of collection in the good flows providing efficient streams to city center from CDC itself and vice-versa, aiming at the reduction of traffic congestion (BESTUFS [65]).

Several projects have been implemented worldwide. Some have failed due to different reasons. For example, companies are sometimes reluctant to lose the contact with customers or to transport their goods with the competitors due to sharing of information; in other cases, the CDC was located too far from highways and centers or the municipality adopted wrong supporting policy measures (BESTUFS [65], van Rooijen and Quak [211]). Despite that, practitioners are still convinced that the use of CDC in distribution system can improve traffic and environmental condition of city center and make distribution more efficient.

In what follows the term *logistic platform* (LP) is used for both CDC and satellites. The context will make clear which platform is being considered.

Problem description

An LP is usually located in strategical position that can be the airport, close to the train station or in the seaport. Furthermore, an LP can be located along the highway ring just outside the city where big trucks can easily have access. Numerous parameters can characterize each LP as storage capacity, storage period limit, vehicle accessibility, number of vehicle that can access it at the same time, possibility of vehicle-waiting and so on.

In real frameworks, firms store goods in LP from where they are dispatched to the final users directly or via transshipment, i.e., goods are delivered to other (intermediate) LPs. Moreover, different enterprises can agree in consolidating their products in the same vehicle that will be delivered to common customers, sharing transportation costs and respecting traffic limitations. LPs facilitate these operations where goods from different enterprises arrive and are stored or loaded in smaller vehicles that can have access to the city center and reach the customers.

The most studied system is the Two-Echelon VRP (2E-VRP) depicted in Figure 2.3. Goods are initially in 1-st level LP(1-LP). 1-LP are represented by black squares. They are

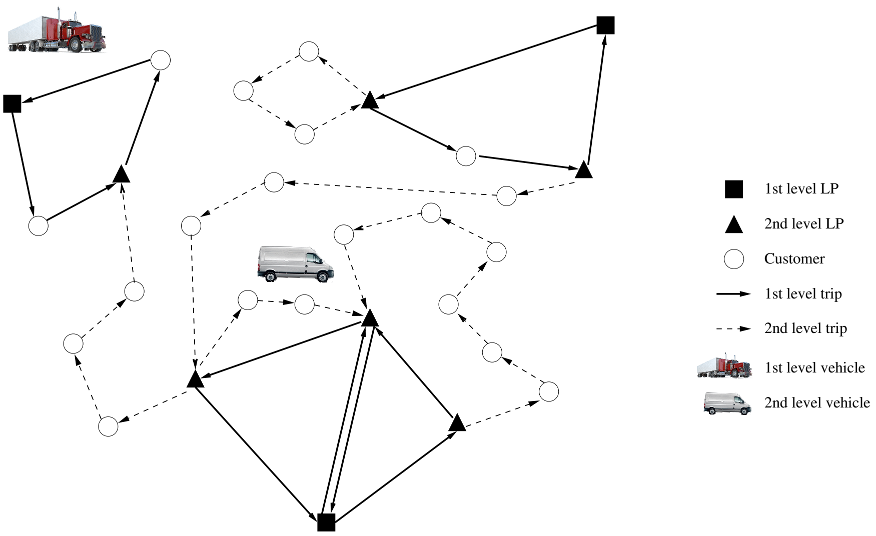
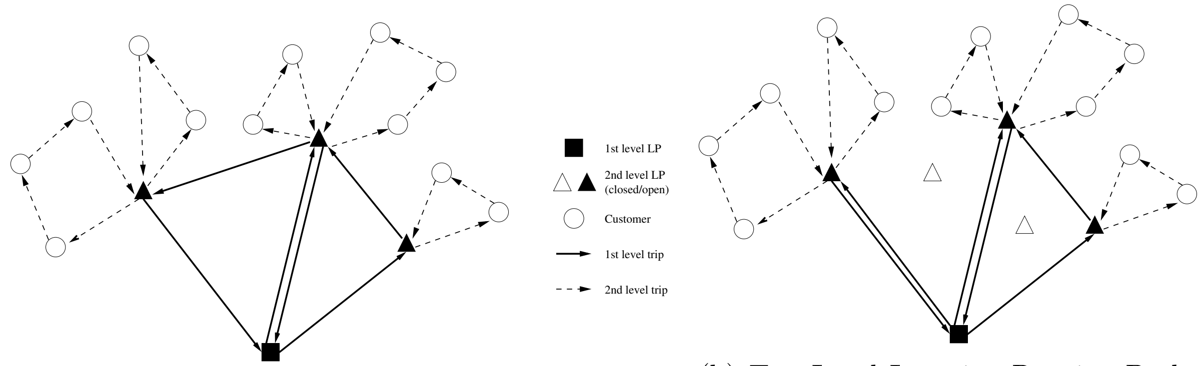


Figure 2.3: General Two-Level distribution system

the *source* of the merchandise: airports, ports, stations, industrial parks, etc. Products are delivered to 2-level LP (2-LP) by 1st level vehicles (1-vehicles). Then they are consolidated into 2nd level vehicles (2-vehicles) that perform final delivery to customers. Direct delivery to customer made by 1-vehicle is considered if access to the customer is not limited. Moreover, the quantity to deliver to 2-LP can be split among different 1-vehicles (then 2-LP can be visited by more than one 1-vehicle) and 2-vehicles can end the trip at a 2-LP different than the one from where they have started. It is assumed that each vehicle can perform several serving trips during the working day (multi-trip aspect). Apart from Crainic et al. [56] who proposed a model that generalizes this system (for example, time dependent travel times and TW associated with customers are considered as well), studies are on simplified schemes. Generalizations of the 2E-VRP with more than 2 levels can be considered.

Common assumptions in the problem definition of the 2E-VRP are the following. There is a unique 1-LP (the central depot) where goods are initially placed. Customers cannot be served directly (for example, due to limitation on vehicles that can access city centers), then goods are delivered to 2-LP by 1-vehicles. 2-LP are usually associated with a capacity. Deliveries from 1 vehicles cannot exceed this capacity. Quantities to deliver to 2-LP can be split among different 1-vehicles. Customers can be served only by 2-vehicles and split in the deliveries is not allowed at that level. Moreover, vehicles can perform only one serving trip that must start and end at the same 2-LP(Crainic et al. [52], Perboli et al. [160], and Crainic et al. [50], Hemmelmayr et al. [104]). This situation is depicted in Figure 2.4a.



(a) Two-Level Routing Problem

(b) Two-Level Location Routing Problem

Figure 2.4: Two-Level Problems

When the 2-LPs that are used need to be chosen among a set of possible 2-LPs, the Two-Echelon Location Routing problem (2E-LRP) arises. Usually a cost associated with the usage of a certain 2-LP (opening cost) is given and the objective is to minimize opening and routing costs. It is noteworthy that when all the opening costs are zero, 2E-VRP and 2E-LRP coincide. Nguyen et al. [149] forbid splitting deliveries even when serving 2-LPs (Figure 2.4b), while Jepsen et al. [117] allows it. The interested reader is referred to Nagy and Salhi [146] for a recent and complete survey on location routing problems.

An implicit assumption of multi-level systems is the sharing of information. To properly manage the merchandise flow from 1-LP to customers, knowledge is needed on the exact destination of each 1-vehicle, the merchandise it is carrying and to which customer it will be delivered by a specific 2-vehicle. This allow goods to be rightly moved into the LP from the arrival.

It should be noticed that, in the previous models, the only interaction between 1- and 2-vehicles regards capacity constraints: 2-vehicles need to be enough in order to serve the customers and deliver the quantity of goods unloaded from 1-vehicles at 2-LPs. In fact, as pointed out in Crainic et al. [56], in advanced systems, 1- and 2-level vehicles are synchronized for cross-docking purposes (goods are unloaded from 1-vehicle and loaded in 2-vehicles almost immediately). Moreover, when 2-LPs do not offer storage possibility, vehicles should arrive at the LP on time.

Synchronization between 1- and 2-vehicles has received little attention until now. We are aware of the the work of Grangier et al. [94] where the two levels are simultaneously considered. 2-LPs do not offer storage possibilities, thus, good transfer at 2LPs requires the simultaneous presence of 1- and 2-vehicles.

Vehicle synchronization is considered in routing problems that differ from the 2E-VRP considered here. For example, in Lee et al. [130] and Wen et al. [216] vehicles are located at the 2-LP. Merchandise needs to be picked-up at supplier locations and delivered to retailer (by the same vehicles) via the 2-LP where consolidation takes place. While Lee et al. [130] force all the vehicles to arrive at the 2-LP simultaneously, Wen et al. [216] associate to products a *ready date* that consists of the sum of the arrival time of merchandise plus the unloading time. The vehicle considered for final deliver needs to wait until the ready date for loading the product. For a survey on synchronization in VRP, we refer to Drexl [62].

Algorithms for the 2E-VRP

We start this section by observing that if customers are assigned to a specific 2-LP, one has to solve as many VRP problems as the number of 2-LP. Moreover, the assignment of customers to 2-LP determines as well the quantity that has to be delivered to each of the 2-LPs by 1-vehicles. 1-level operations are determined by solving either a split delivery VRP (when splitting is allowed) or a capacitated VRP. Then, the problem is naturally decomposed in $|\text{2-LP}| + 1$ routing problems.

Following this decomposition of the problem determined by its structure, different two-phase heuristics have been proposed for the 2E-VRP. Crainic et al. [52] decompose the 2-level problem in smaller and independent VRPs clustering customers and assigning them to 2-LPs. The 2-level distribution is solved as well as Multi-Depot VRP. 1-vehicles deliver goods to 2-LPs accordingly to customer assignment. The second phase improves the initial solution. Perboli et al. [160] propose a flow-based model for the 2E-VRP. They assign customers to 2-LPs regarding the solution of the linear relaxation of the model. In the second phase they solve the corresponding $|\text{2-LP}| + 1$ VRPs. Crainic et al. [50] find an initial assignment of customers to 2-LP and improves it. The solution is improved by means of a multi-start approach.

Other procedure used are GRASP with path-relinking (Crainic et al. [53]), adaptive large neighborhood search (Hemmelmayr et al. [104]), or Branch-and-Cut (Jepsen et al. [117]). Baldacci et al. [15] proposes an exact method that can be summarized in three steps: first, the set $\mathcal{R}$ of all the 1-level routes is generated. Second, all the subsets of $\mathcal{R}$ that can be part of an optimal solution are generated. For each of these subsets, the corresponding 2-level routing problem is solved.

We refer to Gonzalez-Feliu [92] for a recent survey on Two-Level distribution systems.

2.4.3 Dynamic vehicle routing problem

In classical routing problems all the information is supposed to be known before the operations start and the complete information is used to produce the planning. Differently, in dynamic context, information is revealed during the operations. New information need to be treated and re-optimization executed.

Authors	year
Jacobsen and Madsen [116]	1980
Crainic, Ricciardi and Storchi [55]	2004
Crainic [49]	2008
Crainic, Mancini, Perboli and Tadei [52]	2008
Gonzalez-Feliu and Morana [91]	2008
Gendron and Semet [86]	2009
Crainic, Ricciardi and Storchi [56]	2009
Perboli, Tadei and Vigo [160]	2011
Crainic, Errico, Rei and Ricciardi [50]	2011
Hemmelmayr, Cordeau and Crainic [104]	2012
Baldacci, Mingozzi, Roberti and Wolfler Calvo [15]	2013
Jepsen, Spoorendonk and Ropke [117]	2013

Table 2.6: Articles cited that concern multi-level distribution

Definitions

Before starting to analyze this family, it is useful to clarify the concept of *dynamism*. We will adopt the following definition (Psaraftis [169]): *"vehicle routing problem is* dynamic *if information (input) on the problem is made known to the decision maker or is updated concurrently with the determination of the set of routes. By contrast, if all inputs are received before the determination of the routes and do not change thereafter, the problem is termed* static*"*.

This definition is in contrast, for example, to the one given by Ghiani et al. [88], that reads as follows: *"a vehicle routing problem is said to be* static *if its input data (travel times, demands, ...) do not depend explicitly on time, otherwise it is* dynamic*. Moreover, a VRP is* deterministic *if all input data is known when designing vehicle routes, otherwise it is stochastic"*.

The second definition introduces a possibility of confusion, since the time-dependent VRP (TDVRP), would be considered as a dynamic problem as well as the dynamic VRP (DVRP), even if the former would be deterministic while the latter stochastic. To avoid this confusion, we adopt the first definition of dynamism and we consider the TDVRP as static.

In the static case, all the data are supposed to be known in advanced, i.e., before the planning takes place, and do not change afterwards. Vice versa, in the dynamic case part (Gendreau et al. [84], Ichoua et al.[111], Potvin et al.[164]) or all the information becomes available when the vehicles are already routed. The dynamic elements can involve requests, travel times or both (Potvin et al. [164], Fleischmann et al. [80]).

Motivation

The interest in the DVRP (in the literature the word dynamic is often replaced by *real time* or *on-line*) has rapidly grown with the development in communication technology as GPS, mobile phones and geographic information system, giving the opportunity of using real-time information. Then, vehicle activity can be constantly monitored and dynamic demand can be efficiently and rapidly handled since a constant link can be kept between the dispatch center and drivers. Moreover, the possibility of parallel computation allows quick and sophisticated responses.

Güner et al. [98] pointed out that over 50% of travel time delays are due to non-recurrent (and then unpredictable) events. Kim et al. [119] show how using real-time information on congestion can produce cost savings up to 3.65% and reduction of vehicle usage up to 6.88% compared to planes made using travel times based on historical data.

Moreover, results show the benefit of using dynamic knowledge in routing planning. For example, Grzybowska and Barceló [97] implemented a method for the dynamic pickup and delivery VRPTW using the downtown of Barcelona as simulation testing area. Static planning generates solutions that are a third worse than dynamic ones obtained considering time-dependent travel times as well.

Optimization frameworks

In dynamic context, regarding the strategy, requests must be accepted (as, for example, in Fleischmann et al. [80] for taxi cab service or Chen and Xu [38]) or can be refused (Barkaoui and Gendreau [16], Yang et al. [218]). A request can be rejected in the case it is not convenient to serve it or there are no feasible insertions along the planned routes. However, quick decisions have to be made in order to settle if a new demand should be accepted (Gendreau et al. [84], Yang et al. [218]), since customers are willing to wait for answer a short amount of time. When a request is accepted it has to be assigned to a vehicle and its position along the route has to be determined.

While in the VRP the classical objective function is the minimization of the travel distance and/or time, in the dynamic case different objective can be more adequate. For example, Barkaoui and Gendreau [16] minimize customer service denial, total lateness at customer location and total traveled distance.

Usually, to manage dynamic requests, static algorithms are used. The static demands form an initial static problem and each time a new request appears, a new static problem is defined and has to be solved. When rejection is allowed, a quick insertion procedure is run in order to determine whether the request should be accepted or discarded. Then, the customer can know in real time whether its demand will be served. A re-optimization procedure can be used to determine the new routes and is run between dynamic events arrival. In order to save computational time, a simple insertion procedure can also be used but it may produce myopic solutions.

Re-optimization can take place each time a new event occurs, or at pre-fixed *decision epochs* (Chen and Xu [38]). In Gendreau et al. [84] each time a new request arises, the procedure stops and tries to insert the new customer in one of the solutions kept in memory. If no feasible insertion is found the request is rejected, otherwise the next destination of each vehicle is identified regarding the best solution computed so far. Since the routes can often change, drivers just know their next location. In Attanasio et al. [8] each time a request arises a quick procedure called *feasibility check* is performed (they suppose the request is done by telephone or Internet and a customer is not willing to wait more than few seconds). In case of positive response, a re-optimization procedure is run in order to reduce the routing cost as much as possible. Ichoua et al. [111] run a procedure (based on the parallel tabu search developed by Gendreau et al. [84]) for δ time units each time a new request is accepted.

Waiting, diversion and relocation strategies

Since new requests can arise during the planning horizon, it can be worth to let vehicles wait at some strategic locations in order to reduce the overall traveled distance or maximizing the probability to serve a demand. This is in contrast with the static version of the VRP, where the only appropriate strategy requires a vehicle to drive as soon as it is feasible. A *waiting strategy* is an assignment of waiting times to the customer of a tour (including the depot) and their sum does not exceed the slack time, i.e., the difference between the traveled time and the time horizon (Branke et al.[26]). Gendreau et al. [84], force each vehicle to wait at their location if some waiting time is expected at the next destination. The move is then performed as the latest possible time to allow for last minute changes to the planned routes due to the arrival of new requests. Branke et al.[26] implemented different waiting strategies comparing them with the *nowait* strategy (i.e., no waiting times are introduced). The results show that waiting at the depot as long as possible gives worse results than *nowait* strategy while waiting at the farthest location from the depot as long as possible results in shorter routes. Moreover, waiting at each location for the same amount of time (the slack time divided by the number of the routed customers) or for a time proportional to the routed distance allows more customers to be served in addition to decrease the tour length. Mitrović-Minić and Laporte [141] consider the Pickup and Delivery Problem with time windows and propose three waiting strategies to compare with the *nowait* strategy. The first strategy let a vehicle wait at the current location as long as possible. In the other two strategies a route is divided in service zones. After customers in the same service zone are served, vehicles wait as long as possible in the second strategy and for a time proportional to the service zone time in the latter case. The last strategy results to be the best among the others with respect to the route length and the number of vehicle used. Ichoua et al. [113] consider DVRP when some stochastic information is available regarding dynamic demands. In particular, the requests unfold over time according to a Poisson process. In the proposed waiting strategy a vehicle may wait at its current position for some amount of time if the probability for a request to occur in the vehicle's neighbor is higher or equal to a given threshold. Results show that significant improvements are obtained using the waiting strategy especially in the case of small fleet size and high request arrival rates.

In Gendreau et al. [84], when a vehicle move to the next location it cannot be redirected to another site to serve a request that just occurred in its vicinity, i.e., *diversion* is not allowed. Ichoua et al. [111] show that diverting a vehicle to another destination can be beneficial. When a new request is accepted, re-optimization is performed. If in the outcome solution the next destination of a vehicle is different to the current destination, it is diverted (note that the vehicle can be diverted to a location different to the new one). Results show that diversion strategy reduces the total distance traveled, the total lateness and unserved customers.

Bent and Hentenryck [19] show that using *relocation strategies*, i.e., moving a vehicle to an arbitrary location, can improve the quality of the solution when the objective is to maximize the served requests and stochastic information is available for future demands. In particular, they generate sampled requests regarding the stochastic information and consider

Authors	year
Gendreau, Guertin, Potvin, Taillard [84]	1999
Ichoua, Gendreau, Potvin [111]	2000
Attanasio, Cordeau, Ghiani, Laporte. [8]	2004
Fleischmann, Gnutzmann, Sandvoß [80]	2004
Mitrović-Minić and Laporte [141]	2004
Yang, Jaillet, Mahmassani [218]	2004
Branke, Middendorf, Noeth, Dessouky [26]	2005
Chen and Xu [38]	2006
Ichoua, Gendreau, Potvin [113]	2006
Potvin, Xu, Benyahia [164]	2006
Bent and Hentenryck [19]	2007
Ichoua, Gendreau and Potvin [114]	2007
Larsen, Madsen, Solomon [129]	2008
Pureza and Laporte [170]	2008
Berbaglia, Cordeau, Laporte [20]	2010
Grzybowska and Barceló [97]	2012
Pillac, Gendreau, Guéret and Medaglia [162]	2013

Table 2.7: Articles cited that concern DVRP

those requests as real. Then, the vehicle will move to the request with the best evaluation even if that request is forecasted and can not materialize. Results are compared with those obtained using a generalization of the parallel tabu search proposed by Gendreau et al. [84] and a waiting strategy in which vehicles can wait at their location if a request can materialize in the vicinity. Relocation strategy produces the best results.

Different methods are used to tackle these problems, such as adaptive memory tabu search (Gendreau et al. [84]), parallel tabu search (Attanasio et al. [8], Grzybowska and Barceló [97]), dynamic column generation (Chen and Xu [38]), evolutionary algorithm (Branke et al.[26], Barkaoui and Gendreau [16]).

The interested reader is referred to Ichoua et al. [114], Larsen et al. [129], Berbaglia et al. [20] and Pillac et al. [162] for recent surveys on DVRP. Table 2.7 summarizes the articles cited in this section.

2.4.4 Multi-Trip VRP

The Multi-Trip VRP (MTVRP) has been investigated only in the last two decades and the literature is still scarce. However, its interest is recently growing, especially because in city logistics context short trips are common and re-loading necessary. This is a consequence of limited vehicle autonomy when, for example, electrical vehicles are considered, or limited vehicle capacity due to road narrowness or law traffic limitation.

Differently than the other families, MTVRP does not involve network design and data collection problems. Differences arise only at the operational level. For that, in this section the review mainly focuses on the algorithmic aspects.

Multi-Trip VRP

The MTVRP is a variant of the VRP, where vehicles are allowed to perform more than one serving trip in the working day. Then, in the MTVRP, routes have to be determined *and assigned* to vehicles. The assignment part is implicit in the VRP, where each route is associated with one vehicle. Most of the procedures proposed in the literature for the MTVRP, first determine the routes by means of a VRP algorithm. Then, using a bin packing heuristic, a solution is created.

Fleischmann [78] was the first that addressed the problem in his working paper in 1990. He proposed a modification of the saving algorithm and used a bin packing problem heuristic to assign routes to the vehicle. In Taillard et al. [198], VRP solutions are generated using a tabu search algorithm (Taillard [197]). The routes forming the VRP solutions are stored in a list. From that list a subset of routes is selected and a MTVRP solution is constructed using a bin packing heuristic. Petch and Salhi [161] proposed a multi-phase algorithm that minimizes the maximum overtime. A pull of solutions is constructed by the parameterized Yellow saving algorithm. For each solution in the pull, a MTVRP solution is constructed using a bin packing heuristic. The MTVRP solutions are improved using 2-opt, 3-opt moves, combining routes and reallocating customers. Olivera and Viera [156] used an adaptive memory approach to tackle MTVRP. A memory M is constructed with different routes that form VRP solutions generated with the sweep algorithm. Each route is labeled with its overtime value and its cost and are sorted using a lexicographic order. Probabilistically selecting routes in M, new VRP solutions are generated and then improved by a tabu search algorithm. New VRP solutions are used to upload M. From the best VRP solution a MTVRP is obtained using a bin packing heuristic.

Cattaruzza et al. [33] propose an efficient population based algorithm where individuals are evaluated by an extension of the well-know split procedure proposed by Prins [166] for the VRP, that both determines the routes and assigns them to vehicles. Moreover, a new tailored local search operator has been introduced. It detects deteriorating moves (among those usually considered in the VRP context) that together with a swap of trips assigned to different vehicles, yields to a global improvement.

Mingozzi et al. [140] propose an exact method for the MTVRP based on two set partitioning-like formulations. 52 instances with up to 120 customers and with a known feasible solution (without overtime) are faced and in 42 cases the optimal solution is found.

MTVRP with time windows

MTVRP with time widows (MTVRPTW) is faced as well. Azi et al. [11] proposed an exact algorithm for solving the single vehicle MTVRPTW with limited trip duration. The problem is faced via an approach that exploits an elementary shortest path algorithm with resource constraints. In the first phase all non dominated trips are calculated. Then the shortest path algorithm is applied to a modified graph where each node is a non-dominated trip and two nodes are connected whether it is possible to serve the two trips consequently and they do not have customers in common. Solomon instances are used with different values of time horizon. 16 instances out of 54 with 100 customers are solved to optimality. Azi et al. [12] addressed the MTVRPTW with limited trip duration. A column generation approach embedded within a branch-and-price algorithm is developed. A set packing formulation is given for the master problem and each column represents a working day. Since each pricing problem is an elementary shortest path with resource constraints, a similar approach to the one proposed in Azi et al. [11] is applied. As in Azi et al. [11], Solomon instances are considered and a time horizon is introduced. Due to the limitation of the algorithm, the authors focused on instances formed by the first 25 or 40 customers of each Solomon instance. The number of vehicles is set to 2. Their results are overcame by Macedo et al. [136] and Hernandez et al. [106]. The former paper proposes a minimum flow model, while in the latter a set covering formulation is given for the problem and each column represents a trip instead of a working day.

In Crainic et al. [51], Nguyen et al. [148] and Cattaruzza et al. [32] the dependency between the the different distribution levels in two-echelon distribution system is studied. Crainic et al. [51] and Nguyen et al. [148] studied the Multi-Zone MTVRP. This problem deals with synchronization problems that arise in two-echelon systems, where 1st-level vehicles and 2nd level vehicles (see Section 2.4.2 for those definitions) should meet at 2nd level LPs in order to transfer freight. In the Multi-Zone MTVRP, a vehicle starts from a central depot and goes to a *supply point*, i.e, a 2nd level LP during its opening time, loads freight and goes to serve (some) customers associated to the supply point itself meeting customer's TW. Then, it goes to another supply point or back to the depot. TW associated with supply points or customers are hard, then no waiting time is allowed at their locations. This characterization of the 2nd level distribution system was introduced by Crainic et al. [56]. Differently, in Cattaruzza et al. [32] the MTVRPTW with Release Dates is introduced. Truck arrivals at intermediate depots is modeled introducing release dates on merchandise that represent the instant they become available for delivery. Vehicles that perform distribution must wait at the depot until all the products they have to deliver during the next route have arrived.

MTVRP in practice

Different studies facing practical cases consider the possibilities to perform several trips during the working day. For example, Brandão and Mercer [24] considered a MTVRPTW and vehicles with different capacities. Moreover, vehicles can be hired from the company in case of need and the access to some customers is restricted to particular vehicles. Drivers' schedule must respect the maximum legal driving time per day. Legal time breaks and unloading times are taken into account. Real instances from Burton's Biscuit Ltd. including 45 – 70 customers and the use of 11 vans and 11 tractors are considered. In their successive work Brandão and Mercer [25] adapted the algorithm in order to compare the results with those obtained by Taillard et al. [198]. A two-phase tabu search is performed. In the first phase a solution is allowed to become infeasible regarding travel time constraints, but in the second phase, only feasible solutions are accepted. Insert and swap moves are considered. Battarra et al. [17] consider the MTVRPTW and goods belonging to different commodities that cannot be transported in the same vehicle in the meantime. The objective is to minimize the number of used vehicles. The problem is divided in simpler subproblems, one for each commodity. A set of routes is then generated for each commodity and packed by means of a bin packing heuristic in order to obtain a solution. Alonso et al. [3] considered the periodic MTVRP. Then each customer could be served up to t times in a planning period of t time units. Moreover, not every vehicle can serve all the customers. The concept of

Authors	year
Fleischmann [78]	1990
Taillard, Laporte and Gendreau [198]	1996
Brandão and Mercer [24]	1997
Brandão and Mercer [25]	1998
Petch and Salhi [161]	2004
Gribkovskaia, Gullberg, Hovden and Wallace [95]	2006
Salhi and Petch [186]	2007
Olivera and Viera [156]	2007
Azi, Gendreau and Potvin [11]	2007
Alonso, Alvarez and Beasley [3]	2008
Battarra, Monaci, and Vigo [17]	2009
Cornillier, Laporte, Boctor and Renaud [47]	2009
Azi, Gendreau and Potvin [12]	2010
Hernandez, Feillet, Giroudeau and Naudi [106]	2011
Macedo, Alves, Valério de Carvalho, Clautiaux and Hanafi	2011
Crainic, Gajpal and Gendreau [51]	2012
Mingozzi, Roberti and Toth [140]	2012
Cattaruzza, Absi, Feillet and Vidal [33]	2013
Cattaruzza, Absi, Feillet, Guyon and Libeaut [32]	2013
Nguyen, Crainic and Toulouse [148]	2013

Table 2.8: Articles cited that concern MTVRP

multi-trip is also addressed by Cornillier et al. [47] and Gribkovskaia et al. [95]. The first paper concerns petrol distribution to gas stations, while the second proposes a model for the livestock collection.

2.5 Conclusion

Urban areas are growing faster and becoming the cornerstone of our society. Estimations forecast more than 80% of the population will live in cities by 2050 (European Commission [153]). Transportation of people and merchandise impacts the quality of inhabitants' life providing transfers but producing pollution and congestion. Understanding this phenomenon is crucial to manage the city reorganization or enlargement in such a way that both service quality and livability are guaranteed.

In order to contribute to the understanding of good transportation, this paper gives a picture of nowadays urban good movements (UGM). We classified the UGM, describing the main categories and providing statistics of each category with respect to the whole UGM (Section 2.2). It can be noticed that a large part of UGM perform several stops. Optimization of those movements can reduce, for instance, travelling times and pollution and/or increase quality of services.

We surveyed the papers that explicitly studied urban good transportation (Section 2.2.3), both from the authority perspectives (that look for an efficient city organization) and from private point of view (that usually are interested in meeting customer desire providing quality services). From those papers, we identified the problems that scholars, private companies and authorities face while studying urban deliveries - mostly congestion and unforeseen events - or aspects that are considered when solutions are proposed to simultaneously improve delivery efficiency and city livability, for example multi-level distribution systems and the usage of eco-friendly vehicles.

Congestion introduces a time-dependency of travel times. Unforeseen events introduce dynamism on the treated data that implies re-optimization of the planning. In multi-level distribution, concepts as consolidation and synchronization arise. Finally, structural roads limits and short autonomy or small size of eco-friendly vehicles impose short delivering trips with multiple returns to the depot and re-loads. The classical routing problems that arises in this context are the *Time-dependent*, the *Dynamic*, the *Multi-level* and the *Multi-trip* VRP. We surveyed all these families in Section 2.4 without focusing on the city logistics context. Peculiar difficulties arise: optimization of starting time in the time-dependent case, repositioning of the fleet in dynamic environment, synchronization in multi-level systems, assignment of routes to vehicle in the multi-trip situation, to cite some.

We observed that both scholars and enterprises look at multi-level distribution systems as an effective solution to efficiently deliver goods in cities while preserving urban livability. However, this line of research is still recent and focuses on simplified frameworks, that usually do not represent real life situation. Synchronization among levels, for example, starts to be studied nowadays and room is left for future research. Furthermore, the multi-trip aspect that is a natural consequence of the eco-friendly vehicle usage is not considered in several papers. Last, but not the least, time-dependency has not been addressed yet from an algorithmic point of view when studying multi-level distribution systems. We strongly believe that efficient solutions can be found only integrating this feature in future models, due to the relation between time and traveling speed variation that characterizes routing in cities. Moreover, the distribution scheme based on city distribution centers (CDCs) located on the outskirts of the city can lead to original multi-trip vehicle routing problems where a vehicle can start at a given CDC and finish at another one. This organization can be done in conjunction with TSPs that move goods among CDCs.

We outlined the numerous stakes that arise in the urban distribution context as reducing pollution or increasing mobility. There is a need of developing models and criteria that better consider and represent the different stakes in play. Objectives are not always shared by private and public sector. Then, multi-objective models are important avenues of research that can be followed to capture this discordance and obtain satisfactory solutions on both sides. Furthermore, an intelligent city development should be planned by authorities in collaboration with the private sector and considering dwellers' opinions. Future research should study original models where this collaboration is taken into account by proper criteria and objectives.

Researchers should recognize the continuous and dynamic development of our cities and capture those changes into their approaches. Examples are the consideration of access time windows (Muñuzuri et al. [144], Muñuzuri et al. [143]) or the possibility of the carrier to book parking spots in order to avoid illegal parking while delivering (Patier et al. [158]). Moreover, accordingly to Durand and Gonzalez-Feliu [64], e-commerce and distance purchasing flows represent nowadays about 5% of the total end consumer movements (ECM) in number of trips, but would increase up to 25% in the next five years. It can be anticipated that the development of home delivery transportation will also be accompanied with new organizations of deliveries and new types of VRPs. Among others, uncertainties on the presence of customers at home, possibility to deliver to collection points in case of absence, stronger interaction with customers through mobile devices might give raise to new challenges.

Finally, future research should as well explore solutions where merchandise is (partially) transported using the public network and consider consistency aspects as driver knowledge of the street network, association of clients with drivers, non-overlapping of routes and so on.

Chapter 3

The Multi Trip Vehicle Routing Problem: A Survey

Abstract

This paper presents a survey on the Multi-Trip Vehicle Routing Problem. This problem is an extension of the well-known Vehicle Routing Problem, where vehicles are allowed to be re-loaded and re-routed once they end a trip at the depot. It was introduced more than 25 years ago, and since then, researchers have been working on it, but no extensive survey has been proposed.

The contribution of this paper is mainly to fill this gap, proposing a full collection of the works that have been done on the subject. The wide practical applications that are highlighted should encourage academics and practitioners to put their efforts on this problem in further research.

3.1 Introduction

Nuisances related to congestion and pollution pushed scholars, communities and enterprises to study new delivery policies to increase city livability. Common approaches envisage the use of electrical vehicles and/or forbid heavy trucks to enter city centers. Moreover, physical city structure force final deliveries to be accomplished by small-sized vans that can go through narrow streets.

The usage of electrical vehicles or small-sized vans ends up in delivery trips shorter than the working day due to, respectively, limited autonomy and capacity. Unless multiple trips are allowed for vehicles, a consequence is the bad exploitation of the time horizon and the need of an oversized fleet to satisfy all the customers (Cattaruzza et al. [34]). Operations, then, assume the possibility to re-load vans when they are back at the depot and route them for another trip.

The routing problem that arises is the Multi-Trip Vehicle Routing Problem (MTVRP). To the best of our knowledge the MTVRP was first introduced by Fleischmann [78] in 1990 under the name *Vehicle Routing Problem with Multiple Use of Vehicles*. The *multi-trip* concept can however be found earlier in the literature. Salhi [185] allows vehicles to perform several trips in the context of fleet composition problems.

The scientific community has studied the problem and several of its variants proposing both heuristic and exact methods. Until now, no literature review has been proposed on the subject (except from Şen and Bülbül [57] that limit their research to the classical MTVRP). We think that a broad overview on the subject should be given in order to make scholars aware on the work done in more than 25 years. This motivates this survey.

The MTVRP appears in the literature under several names. In addition to the already mentioned *VRP with multiple use of vehicles* used by Fleischmann [78], it has been addressed as VRP *with multiple routes* (Azi et al. [11]), VRP *with multiple trips* (Olivera and Viera [156]) and VRP *with multiple depot returns* (Tsirimpas et al. [206]). Taniguchi and Van Der Heijden [201] allow vehicles to make multiple *traverses*, while the multiple usage of vehicles has been called *recycling of trucks* in Van Buer at al. [210].

This difference in the problem naming, made the research of papers difficult. Moreover, several works dealing with multiple trips do not mention this aspect in the title, nor in the abstract, making the hunt even harder. For this reason there could probably be some fishes that escaped our net.

All along the paper we will refer to a *trip* as a sequence of customer services preceded and followed by a visit to the depot and without intermediate depot returning. A sequence of trips performed by the same vehicle will be called *journey*. In the literature *trip* and *journey* can be respectively referred to *trip* and *tour*, *tour* and *multi-tour* (Aghezzaf et al. [1]) or to *voyage* and *route* (in maritime context, Section 3.4).

To further clarify, we say that a problem involves the multi-trip aspect, when vehicles have the possibility to accomplish journeys made by more than one trip. Roughly speaking, the multi-trip aspect is involved in a problem definition when a vehicle can leave the depot more than once during the horizon. In this paper we will survey works that deal with problems characterized by this feature.

We limit this survey to papers considering problems where the depot is unique, and then vehicles start and end all their trips at its location. Few exceptions consider several depots, but vehicles are associated with a specific depot that is the only one they visit along the journey. Moreover, we do not consider stochastic problems. In particular, in the stochastic VRP (SVRP), part of the data is stochastic. It can be the travel or the service times, the customers, or the the customer demands. One of the solving approaches for the SVRP is to consider *recourse actions*, i.e., decisions on how to adapt the original plan consequently to the realization of the uncertainty. When customer demands are stochastic, for example, a recourse action is to return at the depot to replenish the vehicle. This can be done when the quantity of product the vehicle is carrying is not enough to serve the remaining customers in the trip. This generates a multi-trip planning. However, we decided to exclude from the survey stochastic problems, since the multi-trip journey of a vehicle is a (possible) consequence of uncertain events and not a planned issue. Moreover, to the best of our knowledge, works on the stochastic MTVRP have not appeared yet. Hence, relevant papers for this survey are not excluded by this decision.

The paper is structured as follows. Sections 3.2 and 3.3 survey the MTVRP and the MTVRP with time windows (MTVRPTW) respectively. Due to their academic relevance, a model is proposed for each problem and a complete overview of results on benchmark instances is given, in addition to the references to all the papers published. Section 3.4 surveys multi-trip problems in maritime context. We decided to dedicate one section to this theme due to intrinsic characteristics of maritime routing that make the problem deeply different from the vehicle routing: costs of vessels are usually taken into account, time horizons are longer and trips visit few locations, which makes trip enumeration easier. Section 3.5 lists papers that make use of multi-trips in route planning coupled with production and/or inventory problems. Section 3.6 surveys papers inspired by real-applications and with a number of particular constraints. Finally, Section 3.7 concludes the paper.

3.2 Multi-Trip Vehicle Routing Problem

The Multi-Trip Vehicle Routing Problem (MTVRP) allows a vehicle to perform several trips during the working day. It can be defined on an directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N} = \{0, 1, \ldots, N\}$ is the set of nodes and $\mathcal{A} = \{(i,j)|i,j \in V, i < j\}$ is the set of arcs. It is possible to travel from node i to node j, incurring in a travel time T_{ij} and covering the distance D_{ij}. Node 0 represents the depot where a fleet of M identical vehicles with limited capacity Q is based. Nodes $1, \ldots, N$ represent the customers to be served, each one requiring a certain quantity Q_i of a product. Service at customer i takes S_i units of time. Service time at the depot is indicated with S_0. A time horizon T_H exists, which establishes the duration of the working day. Overtime is not allowed. It is assumed that Q, Q_i and T_H are nonnegative integers.

The MTVRPTW calls for the determination of a set of trips and an assignment of each trip to a vehicle, such that the travelled distance is minimized and the following conditions are satisfied:

(1) each trip starts and ends at the depot,

(2) each customer is visited exactly once,

(3) the sum of the demands of the customers in any trip does not exceed Q,

(4) the total duration of the trips assigned to the same vehicle does not exceed T_H.

The MTVRP is a $\mathcal{NP}$-hard problem since it can be reduced to the VRP, that is in turn $\mathcal{NP}$-hard (Lenstra and Rinnooy Kan [132]). A VRP instance $\mathcal{I}^{VRP}$ can be transformed into a MTVRP instance $\mathcal{I}^{MTVRP}$ as follows. Let $\mathcal{I}^{VRP}$ an instance for the VRP defined by N^{VRP} customers, M^{VRP} vehicles with capacity Q^{VRP}, travel times T_{ij}^{VRP} and travel distances D_{ij}^{VRP} for each $i, j = 0, \ldots, N^{VRP}$ and customer demands Q_i^{VRP}, $i = 0, \ldots, N^{VRP}$. Analogously, apex $MTVRP$ is used to indicate quantities that refer to an instance $\mathcal{I}^{MTVRP}$ of the MTVRP. Let us construct $\mathcal{I}^{MTVRP}$ as follows. $N^{MTVRP} = N^{VRP}$; $M^{MTVRP} = M^{VRP}$; $Q^{MTVRP} = Q^{VRP}$; $Q_i^{MTVRP} = Q_i^{VRP}$, $i = 0, \ldots, N^{VRP}$; $D_{ij}^{MTVRP} = D_{ij}^{VRP}$, $i, j = 0, \ldots, N^{VRP}$; $T_H = \sum_{(i,j)\in\mathcal{A}} T_{ij}^{VRP}$; $T_{i0}^{MTVRP} = T_{i0}^{VRP} + \frac{T_H}{2}$, $i = 1, \ldots, N^{VRP}$; $T_{ij}^{MTVRP} = T_{ij}^{VRP}$, $i = 0, \ldots, N^{VRP}$, $j = 1, \ldots, N^{VRP}$. Note that each vehicle performing more than one trip would violate the constraint on the time horizon.

The model for the MTVRP we propose is adapted from the model in Azi et al. [12] for a variant of the MTVRP with time windows. We introduce sets $\bar{\mathcal{N}} = \mathcal{N} \cup \{N+1\}$ and

$$\bar{\mathcal{A}} = \Big\{(i,j)|i,j \in \mathcal{N} \setminus \{0\}\Big\} \cup \Big\{(0,i)|i \in \mathcal{N} \setminus \{0\}\Big\} \cup \Big\{(i, N+1)|i \in \mathcal{N}\Big\}.$$

Node $N+1$ replicates the depot. Its demand and service time are null. Set $\bar{\mathcal{A}}$ contains all the arcs connecting each pair of customers, plus arcs that go from the node 0 to each customer, plus the arcs that go from each customer to node $N+1$, plus an arc that goes from node 0 to node $N+1$. We introduce the set $\mathcal{R}$ as well. It contains all the trips forming a solution, allowing some trips to be possibly empty. These trips traverse arc $(0, N+1)$. Without loss of generality, trips in $\mathcal{R}$ are indexed such that if trip r is performed before trip s by the same vehicle, then $r < s$.

The following variables are introduced:

- binary variables x_{ij}^r defined for each pair of nodes $i, j \in \bar{\mathcal{N}}$ such that $(i,j) \in \bar{\mathcal{A}}$, $r \in \mathcal{R}$ that indicate whether arc (i,j) is covered by trip r; Note that $x_{0,N+1}^r = 1$ when trip r is empty;
- binary variables y_i^r for each $i \in \mathcal{N} \setminus \{0\}$ and for each $r \in \mathcal{R}$ that indicate whether customer i is served by trip r;
- binary variables z_{rs}, with $r, s \in \mathcal{R}$ and $r < s$, that indicate whether trips r immediately precedes trip s in a journey;
- binary variables w_r for each $r \in \mathcal{R}$ that indicate whether trip r is the first in a journey;
- continuous variables t_i^r for each $i \in \bar{\mathcal{N}}$ and for each $r \in \mathcal{R}$ that indicate the instant service starts at node i when visited by trip r; each trip r starts at the depot at instant t_0^r and ends at t_{N+1}^r.

The model is as follows.

$$\text{(MTVRP)} \quad \min \sum_{r \in \mathcal{R}} \sum_{(i,j) \in \bar{\mathcal{A}}} D_{ij} x_{ij}^r \tag{3.1}$$

$$\text{s.t.} \quad \sum_{j \in \bar{\mathcal{N}} \setminus \{0\}} x_{ij}^r = y_i^r, \quad \forall i \in \mathcal{N} \setminus \{0\}, r \in \mathcal{R}, \tag{3.2}$$

$$\sum_{r \in \mathcal{R}} y_i^r = 1, \quad \forall i \in \mathcal{N} \setminus \{0\}, \tag{3.3}$$

$$\sum_{i \in \bar{\mathcal{N}} \setminus \{N+1\}} x_{ih}^r - \sum_{j \in \bar{\mathcal{N}} \setminus \{0\}} x_{hj}^r = 0, \quad \forall h \in \mathcal{N} \setminus \{0\}, \forall r \in \mathcal{R}, \tag{3.4}$$

$$\sum_{i \in \bar{\mathcal{N}} \setminus \{0\}} x_{0i}^r = 1, \quad \forall r \in \mathcal{R}, \tag{3.5}$$

$$\sum_{i \in \bar{\mathcal{N}} \setminus \{N+1\}} x_{N+1,i}^r = 1, \quad \forall r \in \mathcal{R}, \tag{3.6}$$

$$\sum_{i \in \mathcal{N} \setminus \{0\}} Q_i y_i^r \leq Q, \quad \forall r \in \mathcal{R}, \tag{3.7}$$

$$t_i^r + S_i + T_{ij} - \alpha \left(1 - x_{ij}^r\right) \leq t_j^r, \quad \forall i, j | (i,j) \in \bar{\mathcal{A}}, \forall r \in \mathcal{R}, \tag{3.8}$$

$$t_{N+1}^r + S_0 \leq t_0^s + \alpha(1 - z_{rs}), \quad \forall r, s \in \mathcal{R} | r < s, \tag{3.9}$$

$$t_{N+1}^r \leq T_H, \quad \forall r \in \mathcal{R}, \tag{3.10}$$

$$\sum_{r < s} z_{rs} = 1 - w_s, \quad \forall s \in \mathcal{R}, \tag{3.11}$$

$$\sum_{r \in \mathcal{R}} w_r \leq M, \tag{3.12}$$

$$x_{ij}^r \in \{0,1\}, \quad \forall i,j|(i,j) \in \bar{\mathcal{A}}, \forall r \in \mathcal{R}, \tag{3.13}$$
$$y_i^r \in \{0,1\}, \quad \forall i \in \mathcal{N} \setminus \{0\}, \forall r \in \mathcal{R}, \tag{3.14}$$
$$z_{rs} \in \{0,1\}, \quad \forall r,s \in \mathcal{R}|r < s \tag{3.15}$$
$$w_r \in \{0,1\}, \quad \forall r \in \mathcal{R}, \tag{3.16}$$
$$t_i^r \geq 0, \quad \forall i \in \bar{\mathcal{N}}, \forall r \in \mathcal{R}. \tag{3.17}$$

The objective function (3.1) minimizes the travelled distance. Contraints (3.2)–(3.3) guarantee that each customer is served exactly once. Constraints (3.4)–(3.6) are flow conservation constraints. Constraints (3.7) force vehicle capacities to be respected. Constraints (3.8) assure schedule components on each trip to be respected. Constraints (3.9) assure schedule components on each journey to be respected. Constraints (3.10) force the time horizon to be respected by each journey. Constraints (3.11) guarantee that each trip is preceded by exactly one other trip, except when it is the first assigned to a vehicle. Constraints (3.12) guarantee that not more than M vehicles are used. α is an arbitrary large value.

Allowing vehicles to perform multiple trips can be beneficial and solutions that cost less than the optimal VRP solution can be obtained. An example is given in Figure 3.1. There are two available vehicles each with capacity $Q = 30$, that need to serve four customers. Customers 1 and 2, and customers 3 and 4 have the same location. Demands are in brackets. The optimal VRP solution serves customers 1 and 3 with one vehicle and customers 2 and 4 with the other. The total travelled distance is $8d$. If vehicles can make several trips, a vehicle serves customers 3 and 4, while the other serves customers 1 and 2 within two round trips. The total travelled distance is $6d$.

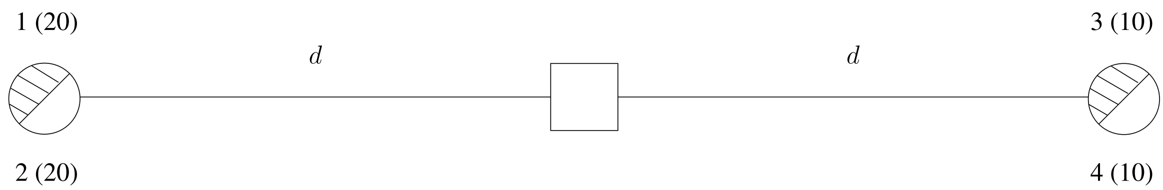


Figure 3.1: An example in which the optimal MTVRP solution costs less than the optimal VRP solution

The benchmark of instances for the MTVRP is introduced in Taillard et al. [198] and is constructed from the instances 1–5 and 11–12 proposed in Christofides et al. [41] (usually denoted CMT1–CMT5 and CMT11–CMT12) and instances 11–12 proposed in Fisher [77] (F11-F12) for the VRP. For each VRP instance, instances for the MTVRP are constructed with different values for the number of available vehicles M and two different values for the time horizons T_H, given by $T_H^1 = \left[\frac{1.05z^*}{M}\right]$ and $T_H^2 = \left[\frac{1.1z^*}{M}\right]$ where z^* is the solution cost of the original CVRP instance found by Rochat [178] and $[x]$ represents the closest integer to x (see Table 3.1). Traveled distances coincide with travel times.

Taillard et al. [198] do not report values on feasible solutions they found on the instances they propose. On the other side, when they cannot find feasible solutions, they penalize overtime by a factor $\theta = 2$ and provide the corresponding solution cost. Apart few exceptions, next researchers follow the same scheme. It is noteworthy that in this case infeasible solutions can cost less than feasible solution. Solution values on this benchmark set of in-

Instance	N	Q	z^*
CMT1	50	160	524.61
CMT2	75	140	835.26
CMT3	100	200	826.14
CMT4	150	200	1028.42
CMT5	199	200	1291.44
CMT11	120	200	1042.11
CMT12	100	200	819.56
F11	71	30000	241.97
F12	134	2210	1162.92

Table 3.1: Instances' details of Taillard et al. [198]

stances are reported in Table 3.2. The first column indicates the instance name and the number of customers N. Columns M, T_H^1 and T_H^2 are self-explanatory. Columns *Opt* report optimal values (when available). Columns *Best Known* report best known values, when the corresponding optimal values are not known. Column *Best Unfeas.* report the value of the best unfeasible solution, when no feasible solutions are known. This column is omitted for instances corresponding to values T_H^2 of the time horizon, since for all the instances a feasible solution have been found. The algorithms that permitted to obtain these solutions are described subsequently.

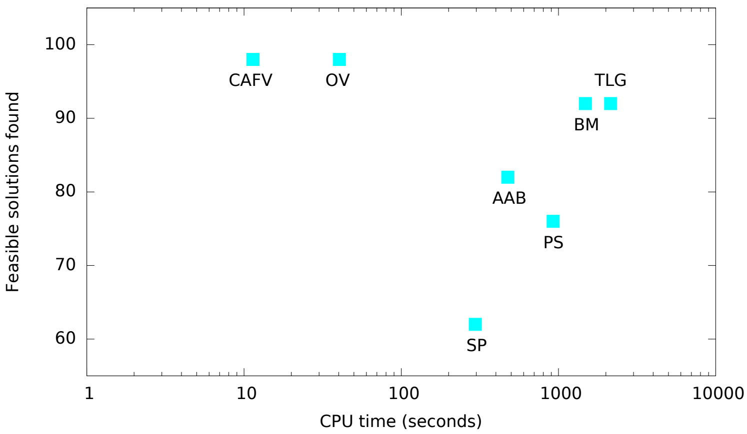
	M	T_H^1	Opt	Best Known	Best Unfeas.	T_H^2	Opt	Best Known
CMT1	1	551	524.61			577	524.61	
$N = 50$	2	275	533.00			289	529.85	
	3	184			569.54	192		552.68
	4	138			564.07	144	546.29	
CMT2	1	877	835.26			919	835.26	
$N = 75$	2	439	835.26			459	835.26	
	3	292	835.26			306	835.26	
	4	219	835.26			230	835.26	
	5	175	835.80			184	835.26	
	6	146		858.58		153	839.22	
	7	125			866.58	131		844.70
CMT3	1	867	826.14			909	826.14	
$N = 100$	2	434	826.14			454	826.14	
	3	289	826.14			303	826.14	
	4	217		829.54		227	826.14	
	5	173		832.89		182		832.34
	6	145		836.22		151		834.35
CMT4	1	1080		1031.00		1131		1031.07
$N = 150$	2	540		1031.07		566		1030.45
	3	360		1028.42		377		1031.59
	4	270		1031.10		283		1031.07
	5	216		1031.07		226		1030.86
	6	180		1034.61		189		1030.45
	7	154		1068.59		162		1036.08
	8	135		1056.54		141		1044.32
CMT5	1	1356		1302.43		1421		1299.86

	M	T_H^1	Opt	Best Known	Best Unfeas.	T_H^2	Opt	Best Known
$N = 199$	2	678		1302.15		710		1305.35
	3	452		1301.29		474		1301.03
	4	339		1304.78		355		1303.65
	5	271		1300.02		284		1300.62
	6	226		1303.37		237		1306.17
	7	194		1309.40		203		1031.54
	8	170		1303.91		178		1308.78
	9	151		1307.93		158		1307.25
	10	136		1323.01		142		1308.81
CMT11	1	1094	1042.11			1146	1042.11	
$N = 120$	2	547	1042.11			573	1042.11	
	3	365	1042.11			382	1042.11	
	4	274		1078.64		287	1042.11	
	5	219	1042.11			229	1042.11	
CMT12	1	861	819.56			902	819.56	
$N = 100$	2	430	819.56			451	819.56	
	3	287	819.56			301	819.56	
	4	215	819.56			225	819.56	
	5	172		845.56		180	824.78	
	6	143			845.48	150	823.14	
F11	1	254		241.97		266		241.97
$N = 71$	2	127		250.85		133		241.97
	3	85			256.93	89		254.07
F12	1	1221		1162.96		1279		1162.96
$N = 134$	2	611		1162.96		640		1162.96
	3	407		1162.96		426		1162.96

Table 3.2: Results on MTVRP instances

3.2.1 Heuristics

Most of the heuristics developed for the MTVRP are based on a two-phase paradigm. In a first step, trips are created and in a second step some of these trips are packed into journeys respecting T_H. Fleischmann [78] proposes a modification of the savings algorithm to obtain trips and uses a bin packing problem (BPP) heuristic to assign trips to vehicles. In Taillard et al. [198], VRP solutions are generated using a tabu search (TS) algorithm with adaptive memory (Taillard [197]). The trips forming the VRP solutions are stored in a list. From that list a subset of trips is selected and a MTVRP solution is constructed using a BPP heuristic. Petch and Salhi [161] propose a multi-phase algorithm with the minimization of the overtime as objective function. A pool of solutions is constructed by the parametrized Yellow's savings algorithm (Yellow [219]). For each solution in the pool, a MTVRP solution is constructed using a BPP heuristic. The MTVRP solutions are improved using 2-opt, 3-opt moves, combining trips and reallocating customers. In Salhi and Petch [186], as in Petch and Salhi [161], the maximum overtime is minimized. A genetic algorithm is proposed. In this method a chromosome is a sequence of strictly increasing angles, measured with respect to the depot, and dividing the plane into sectors. The customers are then clustered by assigning each one to the sector it occupies. In each cluster, the Clarke and Wright savings heuristic is used to solve a smaller VRP problem. The resulting trips are packed using a BPP heuristic. Olivera and Viera [156] use an adaptive memory approach to tackle the MTVRP. A memory $\mathcal{M}$ is constructed with different trips that form VRP solutions generated with the sweep algorithm. Each trip is labeled with its overtime value and its cost and trips are sorted using a lexicographic order. New VRP solutions are generated by probabilistically selecting trips in $\mathcal{M}$ and improved by a TS algorithm. New VRP solutions are used to update $\mathcal{M}$. From the best VRP solution a MTVRP solution is obtained using a BPP heuristic. Cattaruzza et al. [33] propose a population based algorithm for the MTVRP. A new tailored Local Search operator is introduced that takes advantage of the combination of classical moves and swap of trips among different vehicles. Algorithm comparison is given in Figure 3.2[1].



Symbol	Meaning
TLG	Taillard et al. [198]
BM	Brandão and Mercer [25]
PS	Petch and Salhi [161]
SP	Salhi and Petch [186]
OV	Olivera and Viera [156]
AAB	Alonso et al. [3]
CAFV	Cattaruzza et al. [33]

Figure 3.2: Algorithm comparison on benchmark instances

State-of-the-art algorithms are provided by Olivera and Viera [156] and Cattaruzza et al. [33].

The only problem-designed local search (LS) operator we are aware of is proposed in Cattaruzza et al. [33]. It is based on the observation that classical LS pejorative moves (i.e., moves that do not improve the objective function) designed for the VRP along with a better assignment of trips to vehicles can lead to a global improvement in the solution cost. An example is given in Figures 3.3–3.5. O_v indicates the overtime of vehicle v and $\theta = 2$ is the penalty factor for this overtime.

	r_1	r_2	r_3	T_v	θO_v
v_1	60	30		90	-
v_2	30	30	30	90	-
v_3	45	30	30	105	10
				cost:	295

Figure 3.3: Initial configuration

	r_1	r_2	r_3	T_v	θO_v
v_1	60	**25**		85	-
v_2	30	30	**40**	100	-
v_3	45	30	30	105	10
				cost:	300

Figure 3.4: Pejorative move. In bold trips involved in m

The example (Cattaruzza et al. [33]) involves three vehicles with up to three trips each and the time horizon $T_H = 100$ that is violated by the third vehicle (Figure 3.3). Let us consider a move m that involves trips r_2 and r_3 of vehicles v_1 and v_2 respectively (for example

[1]CPU times reported are the original and no scale factor is considered

	r_1	r_2	r_3	T_v	θO_v
v_1	60	**30**		90	-
v_2	30	30	40	100	-
v_3	45	**25**	30	100	-
				cost:	290

Figure 3.5: After re-packing the trips. In bold trips involved in re-packing phase

m relocates a customer from r_2 in r_3) and leads to the configuration shown in Figure 3.4, with an increase in the solution cost of 5 units (due to the increase in routing cost). In a classical LS procedure, move m would be discarded since it deteriorates the solution quality. However, with a different assignment of trips to vehicles, an improvement can be obtained. In the particular case, it consists in swapping r_2 in v_1 with r_2 in v_3 (Figure 3.5). This operator is called Combined Local Search and re-packing of trips is limited to swapping trips among vehicles.

3.2.2 Exact methods

Kov and Karaoglan [121] propose a branch and bound algorithm using VRP inequalities that are still valid for the multi-trip case. They face only instances with 50 customers, being able to solve 3 out of 8. They report an upper bound on instance CMT1, $M = 4$, $T_H = 144$ lower than the optimal solution reported by Mingozzi et al. [140].

The only other exact method we are aware of, is the one recently proposed by Mingozzi et al. [140]. It is based on two set partitioning-like formulations. 42 instances with up to 120 customers are solved. Optimal values are reported in column *Opt* in Table 3.2.

3.2.3 Variants of the MTVRP

This section introduces papers that deal with some variants of the MTVRP. Ronen [180] studies a problem of suppling retailers from a central depot by trucks allowed to visit one retailer per trip (trips are fully loaded). When returning at the depot, trucks possibly perform another round trip serving the same or another retailer. The fleet is heterogeneous and each truck is associated with a time availability and a maximal allowed overtime and undertime. The objective is to find an assignment of return trips to trucks in order to minimize routing cost and penalizations deriving from overtime and undertime utilization. A mixed binary formulation is provided and two simple heuristics are developed: the first assigns trips in trucks with overtime to trucks with undertime, the second swaps trips among trucks.

Prins [165] studies the heterogeneous fleet MTVRP where the main goal is to minimize total trip duration (equivalent to travelled distance) and secondly to minimize the number of routed vehicles. This objective function reflect the need to provide a high-quality service. A limitation on the trip duration $t_{max}(\leq T_H)$ is imposed. A two phase heuristic is used: trips are first obtained and then packed into journeys. A real case instance made by 71 trucks with capacities between 350 and 720 units, 775 shops, $t_{max} = 2700$ and $T_H = 3240$ minutes is tackled. Results show a saving of 7491 travelled minutes and 4 trucks compared with the solution obtained by the dispatcher of the company (which travel time is 61050 minutes and needs 23 trucks).

Alonso et al. [3] introduce the Site Dependent and Periodic MTVRP. Customers need to be delivered up to t times during a horizon of t days and cannot be served by all the vehicles (the fleet is heterogeneous). They propose a tabu search procedure. Due to the novelty of the problem, a set of 10 instances is generated. The smallest instance has 50 customers to serve, 2 days horizon, 3 delivery patterns and 2 different vehicles. The biggest instance has 1000 customer, 6 days, 12 patterns and 13 vehicles of 5 different types. Instances are available at `http://people.brunel.ac.uk/ mastjjb/jeb/orlib/sdmtpvrpinfo.html`. The procedure is run over classical instances for comparison purposes.

3.3 MTVRP with Time Windows

In the MTVRP with time windows (MTVRPTW) each customer i is associated with a time interval $[E_i, L_i]$ during which service should take place. Arriving at customer location earlier than E_i is usually allowed, but makes the driver wait until the opening of the time window (TW). On the opposite, late arrivals are forbidden. The time horizon is represented associating a TW equal to $[0, T_H] = [E_0, L_0]$ with the depot.

The MTVRPTW calls for the determination of a set of trips and an assignment of each trip to a vehicle, such that the travelled distance is minimized, conditions (1)–(4) are satisfied as well as

(5) trips do not start earlier than E_0 and do not finish later than L_0;

(6) service at customer i must start between E_i and L_i.

The MTVRPTW is obviously a $\mathcal{NP}$-hard problem: each MTVRP instance can be reduced to a MTVRPTW instance setting $E_i = 0$ and $L_i = T_H$ for all $i = 0, \dots, N$.

Differently than in the MTVRP, in the MTVRPTW the sequence of trips assigned to each vehicle is important for both solution cost and feasibility. The model we present for the MTVRPTW extends the model introduced in Section 3.2 and uses the same set of variables.

$$\text{(MTVRPTW)} \quad \min \sum_{r \in \mathcal{R}} \sum_{(i,j) \in \bar{\mathcal{A}}} D_{ij} x_{ij}^r \tag{3.18}$$

$$\text{s.t. } E_i y_i^r \le t_i^r \le L_i y_i^r, \quad \forall i \in \bar{\mathcal{N}}, \forall r \in \mathcal{R}, \tag{3.19}$$

$$(3.2)\text{–}(3.9),$$

$$(3.11)\text{–}(3.17).$$

The objective function (3.18) minimizes the travelled distance. Constraints (3.19) impose that the time windows are respected.

Exact methods

The only work we are aware of, is the one proposed by Hernandez et al. [105]. Solomon's instances of groups C2, R2, RC2 limited to the first 25 and 50 customers are used. Q is set to 100. M is set to 2 for instances with 25 customers, and to 4 for instances with 50. Distances are Euclidean and truncated to one decimal place. Loading time for trip r is equal to $\beta \sum_{i\in r} S_i$ where $i \in r$ indicates that customer i is visited in trip r. To take this into account, constraints (3.9) are replaced by

$$\beta \sum_{i\in r} S_i \le t_0^r, \quad \forall r \in \mathcal{R}, \tag{3.20}$$

$$t_{N+1}^r + \beta \sum_{i\in s} S_i \le t_0^s + \alpha(1 - z_{rs}), \quad \forall r, s \in \mathcal{R} | r < s, \tag{3.21}$$

in the model (MTVRPTW). β is usually fixed to the value 0.2.

Hernandez et al. [105] propose a Branch-and-Price algorithm. They are able to solve 25 out of 27 instances with 25 customers and 5 out of 27 instances with 50 customers. Results are reported in Table 3.3. Column *Opt* reports optimal values, Column *HRN* reports CPU times (in seconds) needed to obtain the optimal solution (a 30-hour limit is imposed). Column *Best Known* reports best known solution without optimality proof. Most of these values are provided by Cattaruzza et al. [31] (they run their procedure for the MTVRPTW with release dates, see Section 3.3.2, on this benchmark of instances). Best known solutions for instances RC208 with $N = 25$ and RC202 with $N = 50$ are provided by the exact method of Hernandez et al. [105] when the computation time limit is reached.

3.3.1 MTVRPTW with limited trip duration

The MTVRPTW with limited trip duration (MTVRPTW-LD) is a variant of the MTVRPTW, where trip duration is limited by a value $t_{\max}$. In particular, the service at the last customer of a trip cannot start more than $t_{\max}$ units of time after departure from the depot. This constraints are motivated by the fact that perishable goods must be delivered before a certain amount of time has passed from the moment they have been loaded. Azi et al. [11] introduced the problem for the single-vehicle version. In their successive work (Azi et al. [12]) the problem is extended to the multi-vehicle case. In particular, each trip r is also characterized by a loading time proportional to the total service time of the trip, i.e., $\beta \sum_{i\in r} S_i$. A particular characteristic of the problem is that serving all customers is not mandatory and a profit P_i is associated with each customer i and represents the profit of serving customer i. The objective function consists in minimize the unserved customers, breaking ties in favor of the minimum travelled distance.

The model for this problem is as follows.

$$\text{(MTVRPTW-LD)} \quad \min \sum_{r\in\mathcal{R}} \sum_{(i,j)\in\bar{\mathcal{A}}} D_{ij} x_{ij}^r - \bar{p} \sum_{r\in\mathcal{R}} \sum_{i\in V} P_i y_i^r \tag{3.22}$$

$$\text{s.t.} \quad \sum_{r\in\mathcal{R}} y_i^r \le 1, \quad \forall i \in \mathcal{N} \setminus \{0\}, \tag{3.23}$$

$$t_i^r \le t_{\max}, \quad \forall i \in \mathcal{N}, \forall r \in \mathcal{R}, \tag{3.24}$$

$$(3.2), (3.4)–(3.10),$$

$$(3.11)–(3.17),$$

$$(3.19)–(3.21),$$

		Opt	Best Known	HRN
RC	201	660.0		15.06
$N = 25$	202	596.8		55867.10
	203	530.1		28816.80
	204	-	518.0	-
	205	605.3		234.55
	206	575.1		1178.86
	207	528.2		23451.80
	208	-	506.4	-
R	201	554.6		31.73
$N = 25$	202	485.0		146.92
	203	444.2		472.17
	204	407.5		5084.34
	205	448.4		38.89
	206	413.9		107.27
	207	400.1		2218.48
	208	394.3		2988.72
	209	418.3		149.50
	210	448.3		234.63
	211	400.1		6628.09
C	201	380.8		17.39
$N = 25$	202	368.6		848.06
	203	361.7		85.72
	204	358.8		376.74
	205	377.2		166.64
	206	367.2		135.45
	207	359.1		403.84
	208	360.9		293.47
RC	201	1096.6		662.16
$N = 50$	202	1001.6		99346.30
	203	-	941.2	-
	204	-	915.9	-
	205	-	1058.7	-
	206	-	1027.4	-
	207	-	941.7	-
	208	-	916.8	-
R	201	909.8		237.34
$N = 50$	202	816.0		78879.90
	203	-	742.4	-
	204	-	702.3	-
	205	807.3		24061.70
	206	-	758.2	-
	207	-	715.7	-
	208	-	699.6	-
	209	-	746.0	-
	210	-	777.2	-
	211	-	717.4	-
C	201	-	714.2	-
$N = 50$	202	-	700.1	-
	203	-	688.0	-
	204	-	685.1	-
	205	-	700.0	-
	206	-	694.6	-
	207	-	689.7	-
	208	-	688.6	-

Table 3.3: Results on MTVRPTW instances by Hernandez et al. [105]

where $\bar{p}$ is a parameter that makes always beneficial to serve a customer. Constraints (3.23) guarantee that each customer is visited at most once. Constraints (3.24) force trips to respect the duration limit.

Exact methods

The presence of TW, the limited vehicle capacity that allows few customers to be served in each trip, and the limit imposed in trip duration, make the enumeration of all feasible trips achievable for small and medium size instances. The proposed exact methods are based on this observation.

Solomon's instances of groups C2, R2 and RC2 considering only the first 25 and 40 customers are adapted to the problem. Two values $t^1_{\max}$ and $t^2_{\max}$ of $t_{\max}$ are imposed for each group of instances: 75 and 100 for groups R2 and RC2 and 220 and 250 for C2 (service times in the latter group are equal to 90 for each customer, while in the other cases they are equal to 10). The number of vehicles is set to 2. Loading time is 0.2 times the sum of customers' service time served in the trip. These instances are introduced by Azi et al. [12].

Azi et al. [12] propose a branch-and-price algorithm. They are able to solve 22 instances with $t^1_{\max}$ and 18 with $t^2_{\max}$ allowing 30 hours of computation (corrected results are reported in Azi [10]). Distances among locations are truncated at the second decimal digit.

Macedo et al. [136] (and Macedo et al. [137]) propose a minimum flow model with a pseudo-polynomial number of variables (the number of variables is polynomial on the working time and on he number of trips, that is limited by a parameter proportional to $t_{\max}$; the number of constraints is polynomial on the working time). They are able to solve 41 instances with $t^1_{\max}$ and 26 with $t^2_{\max}$ allowing 2 hours of computation. Direct comparison with Azi et al. [12] cannot be done, since they do not truncate distances (Hernandez et al. [106]). This explains differences in optimal values on the same instances between Azi [10] and Macedo et al. [136].

Hernandez et al. [106] propose a Branch-and-Price algorithm. Differently than Azi et al. [12] and Macedo et al. [136] they consider mandatory to serve all customers. On some instances, they prove that a solution that serves all of them does not exist. Tests are run twice on instances proposed by Azi et al. [12]: once rounding distances at the second decimal digit and once without any rounding (for comparison purposes with Macedo et al. [136]).

Result comparison is reported in Tables 3.4–3.7. Columns *Opt* report the optimal value. Next columns report for each algorithm the CPU time (in seconds) needed to solve the instance, a "-" if the algorithm could not find an optimal solution or *NoSol* if Hernandez et al. [106] proved that all the customers cannot be served with the available vehicles. For these instances Azi et al. [10] or Macedo et al. [136] might have found the optimal solutions serving a subset of customers. In these specific cases, the percentage of unserved customers and the travelled distance are reported in the last two columns. Corresponding CPU times, then, refer to the time needed to find these solutions. In Tables 3.4–3.5 comparison is made between Macedo [136] (MCD) and Hernandez et al. [106] while in Tables 3.6–3.7 comparison is made between Azi [10] (AZI) and Hernandez et al. [106] (HRN).

One can notice that all previous works have a first phase where all the trips are generated. In the second phase, trips are put together in order to form a journey. This can be done

		Opt	MCD	HRN	MCD	
			CPU time	CPU time	Unserved (%)	Dist
RC	201	988.20	0.3	1.1		
$N = 25$	202	881.60	37.2	24.8		
$t_{max} = 75$	203	749.26	54.2	64.0		
	204	744.83	171.0	-		
	205	840.47	1.6	3.4		
	206	761.14	2.0	34.4		
	207	-	-	-		
	208	-	-	-		
R	201	762.53	0.5	0.1		
$N = 25$	202	645.86	3.1	0.6		
$t_{max} = 75$	203	622.04	10.6	2.2		
	204	579.75	106.2	5.0		
	205	634.17	1.5	0.8		
	206	596.81	4.7	0.9		
	207	585.81	19.4	4.7		
	208	579.75	66.0	7.4		
	209	602.47	4.9	1.6		
	210	636.24	11.8	8.0		
	211	575.97	64.5	25.9		
C	201	659.15	10.6	1.4		
$N = 25$	202	653.50	212.4	49.9		
$t_{max} = 220$	203	646.51	233.9	265.9		
	204	602.58	423.0	257.1		
	205	636.52	34.7	44.9		
	206	636.52	40.2	699.7		
	207	603.34	29.5	92.3		
	208	613.34	12.9	42.6		
RC	201	NoSol	29.4	0.4	22.5	1292.35
$N = 40$	202	NoSol	40.2	2.4	7.5	1458.09
$t_{max} = 75$	203	NoSol	-	5.6		
	204	-	-	-		
	205	NoSol	6992.6	0.9	15.0	1290.75
	206	NoSol	-	1.9		
	207	NoSol	-	4.5		
	208	-	-	-		
R	201	NoSol	2358.8	0.4	5.0	1130.73
$N = 40$	202	-	-	-		
$t_{max} = 75$	203	962.42	436.0	-		
	204	858.35	-	3811.2		
	205	1019.89	3263.7	2902.3		
	206	931.94	209.9	190.9		
	207	890.93	-	276.2		
	208	858.35	-	4328.0		
	209	935.95	771.3	227.2		
	210	963.45	1803.9	1297.1		
	211	869.88	-	4187.2		
C	201	1169.04	25.5	32.8		
$N = 40$	202	1111.34	79.4	70.3		
$t_{max} = 220$	203	1089.24	342.3	135.5		
	204	1039.35	-	112.9		
	205	1084.02	63.6	34.0		
	206	1081.57	109.3	173.5		
	207	1055.24	659.0	1700.3		
	208	1072.22	112.7	52.3		

Table 3.4: Comparison of Hernandez et al. [106] and Macedo et al. [136] on instances with short t_{max} and non-truncated distances

		Opt	MCD	HRN	MCD	
			CPU time	CPU time	Unserved (%)	Dist
RC	201	849.45	2.0	3.2		
$N = 25$	202	679.95	11.6	2.9		
$t_{max} = 100$	203	593.63	47.0	12.9		
	204	-	-	-		
	205	702.61	8.2	2.2		
	206	604.23	8.0	3.8		
	207	514.90	91.7	48.1		
	208	-	-	-		
R	201	698.26	1.3	0.8		
$N = 25$	202	617.60	32.6	4.5		
$t_{max} = 100$	203	577.80	64.1	10.5		
	204	483.37	-	29.9		
	205	559.21	9.4	3.3		
	206	523.70	40.0	6.4		
	207	512.04	-	425.6		
	208	483.37	-	82.3		
	209	517.74	47.7	12.3		
	210	547.29	58.9	2.4		
	211	474.54	-	59.1		
C	201	541.02	0.4	0.1		
$N = 25$	202	533.55	167.9	41.0		
$t_{max} = 250$	203	532.88	-	298.4		
	204	525.57	-	4711.6		
	205	530.05	3.7	0.9		
	206	527.95	20.7	129.3		
	207	525.57	44.2	25.8		
	208	525.57	63.2	114.1		
RC	201	NoSol	3.6	0.8	15.0	1157.65
$N = 40$	202	NoSol	1013.8	10.4	2.5	1322.08
$t_{max} = 100$	203	-	-	-		
	204	-	-	-		
	205	NoSol	35.7	3.9	5.0	1195.51
	206	-	-	-		
	207	-	-	-		
	208	-	-	-		
R	201	NoSol	-	13.2		
$N = 40$	202	-	-	-		
$t_{max} = 100$	203	816.65	-	2469.0		
	204	-	-	-		
	205	873.36	-	1200.4		
	206	812.42	-	1514.2		
	207	764.52	-	7029.8		
	208	-	-	-		
	209	768.99	-	454.0		
	210	-	-	-		
	211	-	-	-		
C	201	966.89	6.4	85.4		
$N = 40$	202	920.05	-	81.1		
$t_{max} = 250$	203	-	-	-		
	204	-	-	-		
	205	921.37	88.5	54.8		
	206	919.24	290.5	1522.2		
	207	-	-	-		
	208	915.61	491.5	2582.7		

Table 3.5: Comparison of Hernandez et al. [106] and Macedo et al. [136] on instances with long t_{max} and non-truncated distances

		Opt	AZI	HRN	AZI	
			CPU time	CPU time	Unserved (%)	Dist
RC	201	988.05	3.1	0.9		
$N = 25$	202	881.49	-	24.5		
$t_{max} = 75$	203	749.15	-	62.3		
	204	-	-	-		
	205	840.35	28.8	3.7		
	206	761.03	7156.8	35.7		
	207	-	-	-		
	208	-	-	-		
R	201	762.43	68.3	0.1		
$N = 25$	202	645.78	205.2	0.6		
$t_{max} = 75$	203	621.97	1333.2	2.0		
	204	579.68	30983.3	4.9		
	205	634.09	354.1	1.0		
	206	596.74	318.4	0.8		
	207	585.74	2853.5	3.5		
	208	579.68	9270.3	7.2		
	209	602.39	262.6	1.7		
	210	636.15	5094.1	8.5		
	211	575.91	5648.6	27.6		
C	201	659.02	40361.2	1.3		
$N = 25$	202	653.37	-	49.3		
$t_{max} = 220$	203	646.40	-	265.0		
	204	602.46	-	248.0		
	205	636.39	-	38.1		
	206	636.39	-	692.4		
	207	603.22	-	104.7		
	208	613.20	-	41.4		
RC	201	NoSol	14.6	0.3	22.5	1292.16
$N = 40$	202	NoSol	6823.2	2.4	7.5	1457.89
$t_{max} = 75$	203	NoSol	-	5.8	-	-
	204	-	-	-	-	-
	205	NoSol	1904.2	0.9	15.0	1272.12
	206	NoSol	-	1.7	-	-
	207	NoSol	-	4.1	-	-
	208	-	-	-	-	-
R	201	NoSol	2979.5	0.5	5.0	1130.59
$N = 40$	202	-	-	-		
$t_{max} = 75$	203	-	-	-		
	204	858.22	-	4049.2		
	205	1017.84	244494	1193.4		
	206	927.22	-	171.5		
	207	886.22	-	68.9		
	208	858.22	-	4954.8		
	209	935.81	-	198.2		
	210	952.92	-	246.5		
	211	869.75	-	5093.9		
C	201	1168.83	19978.9	31.3		
$N = 40$	202	1111.15	-	67.4		
$t_{max} = 220$	203	1088.55	-	186.5		
	204	1039.16	-	145.3		
	205	1083.81	-	34.1		
	206	1081.37	-	184.0		
	207	1055.04	-	1491.5		
	208	1071.99	-	52.3		

Table 3.6: Comparison of Hernandez et al. [106] and Azi et al. [10] on instances with short t_{max} and truncated distances

		Opt	AZI	HRN	AZI	
			CPU time	CPU time	Unserved (%)	Dist
RC	201	849.33	16.06	3.6		
$N = 25$	202	679.86	1096.3	3.5		
$t_{max} = 100$	203	593.56	-	13.1		
	204	-	-	-		
	205	702.49	262.8	2.6		
	206	604.12	222.7	2.9		
	207	514.81	-	45.5		
	208	-	-	-		
R	201	698.18	43.6	0.8		
$N = 25$	202	617.53	25249.9	4.1		
$t_{max} = 100$	203	577.74	75729.3	11.6		
	204	483.30	-	33.6		
	205	559.14	1202.3	3.7		
	206	523.64	28498.7	5.7		
	207	512.00	-	418.9		
	208	483.30	-	97.8		
	209	517.69	11173.9	14.4		
	210	547.23	26690	2.6		
	211	474.49	-	80.4		
C	201	540.90	1.3	0.1		
$N = 25$	202	533.43	-	51.4		
$t_{max} = 250$	203	532.77	-	335.7		
	204	525.46	-	4734.4		
	205	529.94	116.6	0.9		
	206	527.84	1987.2	123.9		
	207	525.46	-	31.1		
	208	525.46	-	4.7		
RC	201	NoSol	77.8	0.6	15.0	1157.48
$N = 40$	202	NoSol	-	10.3		
$t_{max} = 100$	203	-	-	-		
	204	-	-	-		
	205	NoSol	4733.3	4.2	5.0	1195.32
	206	-	-	-		
	207	-	-	-		
	208	-	-	-		
R	201	NoSol	127424	15.5	2.5	1075.05
$N = 40$	202	-	-	-		
$t_{max} = 100$	203	816.51	-	2429.0		
	204	-	-	-		
	205	872	-	926.4		
	206	812.31	-	1511.4		
	207	-	-	-		
	208	-	-	-		
	209	768.84	-	479.8		
	210	-	-	-		
	211	-	-	-		
C	201	966.7	659.2	90.4		
$N = 40$	202	919.85	-	84.2		
$t_{max} = 250$	203	-	-	66.3		
	204	-	-	-		
	205	921.19	-	1539.1		
	206	919.05	-	2673.7		
	207	-	-	-		
	208	915.41	-	-		

Table 3.7: Comparison of Hernandez et al. [106] and Azi et al. [10] on instances with long t_{max} and truncated distances

because the number of trips that need to be generated is limited by the short trip duration imposed by t_{max}. It is then not surprising that instances with shorter time limitation can be solved more easily.

Heuristics

Heuristic schemes allow to tackle bigger problem than those treated in the previous section. Then, instances used in this case are adapted from RC2, R2 and C2 instances proposed in Solomon [196] and the bigger instances proposed in Gehring and Homberg [82]. They are introduced in Azi et al. [14]. $t_{\max}$ is set to 100. Customer locations in Gehring and Homberg [82] instances are normalized in order to fit the $100\times$ 100 square as in Solomon instances. Service time is set to 10 for each customer. The number of vehicles is set to 3, 6, 12, 18, 24, 30 for, respectively, instances with 100, 200, 400, 600, 800 and 1000 customers. Loading time of trip r is equal to $\beta \sum_{i\in r} S_i$. Usually $\beta = 0.2$.

We are aware of only two algorithms designed specifically for this problem. The first is proposed by Azi et al. [14] and proposes an adaptive large neighborhood search for the problem. Destruction and insertion operators consider customers, trips and journeys.

Wang et al. [215] propose an algorithm based on the Adaptive Memory Procedure paradigm. Each solution that is constructed is inserted in a memory $\mathcal{M}$. When $\mathcal{M}$ reaches a certain size, a solution is constructed probabilistically selecting trips based on the quality of the solution they belong. Results obtained by Wang et al. [215] outperform those provided by Azi et al. [14] with respect to quality, but the procedure is on average three time more expensive in terms of CPU time. Results are reported in Table 3.8.

3.3.2 Variants of the MTVRPTW

In this section variants of the MTVRPTW are introduced. Cattaruzza et al. [31] introduced the MTVRPTW with release dates. A release date is associated with each merchandise and represents the time it becomes available at the depot for final delivery. In city distribution systems, trucks bring merchandise to city distribution centers (CDC), from where vans perform final deliveries to customers. Merchandise is supposed to arrive at the CDC continuously all along the working day, i.e., not all the goods are available for distribution at the beginning of the horizon. Release dates model this problem characteristics. The algorithm is run on instances for the MTVRPTW for comparison purposes and results are reported in column *Best known* of Table 3.3.

Crainic et al. [51] and Nguyen et al. [148] introduce the Multi-Zone MTVRP. This problem deals with synchronization problems that arise in two-echelon systems, where trucks that bring merchandise to CDC located around cities need to meet vans that perform final delivery to customers. In the particular Multi-Zone MTVRP, a vehicle starts from a central depot and goes to a CDC during its opening time, loads freight and goes to serve (some) customers associated to the CDC itself meeting customer's TW. Then, it goes to another CDC or back to the depot to terminate its trip. TW associated with CDC or customers are hard and no waiting time is allowed at their locations. In the Multi-Zone MTVRP, vehicles leave the depot once during the planning horizon, then they do not perform multi-trips in the sense

N	Instance class	Azi et al. [14]			Wang et al. [215]		
		unserved (%)	dist	CPU (s)	unserved (%)	dist	CPU (s)
100	RC	25.7	1899.2	28.1	22.9	1739.1	45
	R	10.9	1828.6	33.5	9.2	1588.5	49
	C	0	2232.9	43.2	0	1945.9	17.8
200	RC	12.4	9218.3	140.3	10.7	6576.3	163.9
	R	6.2	11103.7	126.9	3.2	7899	105.1
	C	0	9730.3	126.2	0	6646.7	67.9
400	RC	22.4	10128	460.4	19.5	7658.5	1155.2
	R	6.2	12657.7	427.7	3.9	9738	667.3
	C	0	10937.2	340	0	7727	416.3
600	RC	20.4	15577.9	1114.2	16.5	11165.1	4186
	R	6.4	19089.2	1009.1	4.2	14484.6	1950
	C	9.9	14626	1028.5	2.5	11616.7	2646.1
800	RC	24.3	20858.5	1842.8	19.5	14891.5	15677.1
	R	7.9	26136.9	1678.3	6	19763.1	6181.4
	C	24.9	14441.1	1747.4	13.4	11946.6	5756.9
1000	RC	26.3	25368.3	2855.2	25.2	19407.6	11874.4
	R	10.3	30732.2	2718.8	9.7	25032.1	7421.6
	C	36.5	14587.6	2602.4	30.6	12888.7	8449.3
	average	13.9	13953.0	1017.9	10.9	10706.4	3712.8

Table 3.8: Results on Azi et al. [14] instances

given in the Introduction of this paper. We decided to include the papers in the survey for completeness purposes, since the world *multi-trip* appears in the title.

Azi et al. [13] introduces a dynamic version of the MTVRPTW-LD. Requests are rejected when service is infeasible or not worth it. Different expected solutions are created based on possible occurrence of requests. Information on future request revealing can be obtained, for example, from historical information. New requests are tried to be inserted in expected solutions and the total profit summed up. If it is positive, the request is accepted and the plan re-optimized.

3.4 MTVRP in maritime

In maritime problems a set of *vessels* have to be sailed from a central *onshore supply depot* to several *offshore platforms*. A parallelism is evident with the classical VRP. However, important differences characterize maritime problems and we decided to treat them in a separate section. The reader interested in maritime transportation is referred to Christiansen et al. [40] and Christiansen et al. [39].

Remarkable differences peculiar to maritime problems are given here. First of all, the goal is to minimize sailing costs and *time charter* costs. i.e., costs due to the rent of vessels. Routing problems usually focus in minimizing only routing costs. Vehicle costs are considered in strategic problems where the fleet of vehicles needs to be sized. Second, vessels are usually chosen among different size vessels that go along with different renting costs and characteristics (capacity, sailing speed, accessibility to ports, etc.), while classical routing problems usually consider a homogeneous fleet. Other differences are the length of the time horizon T_H that goes from a working day in routing problems to weeks in the maritime case; routing times are shorter than sailing times; problem sizes are bigger in routing than in maritime.

Another important difference is that vessels are usually not going back to the original port. This makes most of the studies related to maritime falling out of the operating assumption that has been made in the introduction of this paper. There are however, some works that fit in our scheme.

Fagerholt [72] calls *liner shipping problem* (LSP) the problem of finding a set of vessels and an order of visits to the offshore platforms in order to minimize sailing and charter costs respecting capacity and time constraints. Moreover, vessels are allowed to be re-sailed once they are back at the onshore supply depot. We will use this terminology in the following of this paper. Due to the strategical nature of the objective function, other works classify this problem as an *optimal fleet composition* problem.

Fagerholt [72] propose a three-phase method for the *liner shipping problem* (LSP). It calls for the determination of a set of vessels and an order of visits to the offshore platforms in order to minimize sailing and charter costs respecting capacity and time constraints. Vessels are allowed to be re-sailed once they are back at the onshore supply depot.

In the first phase all the possible trips are generated for all the available vessels. In the second phase, trips are packed into the same vessels, always respecting feasibility. In the third phase, a solution is constructed solving a set partitioning problem. A real instance formed by 15 nodes and 5 different vessels available is solved. Moreover, three test instances with 20, 30 and 40 customers are created.

Fagerholt and Lindstad [73] introduce some temporal aspects in the LSP. Some offshore platforms close overnight, generating a time window for service. The algorithm proposed follows the same idea as the previous one. First all the feasible trips are generated, then a solution is constructed solving an integer programming model.

In Halvorsen-Weare et al. [100], phases 1 and 3 of Fagerholt's [72] approach are adapted to a more complicated problem where additional constraints are considered. Opening hours associated with onshore and offshore platforms need to be respected. Since each platform closes all days for a certain period, and the planning horizon counts several days, a multiple time-window is associated with each platform. Offshore platforms need to be visited more than once during T_H and departures from the onshore supply depot to the same offshore platform need to be *evenly spread*. This introduces the periodicity aspect. Finally trips need to be neither too short (for better capacity exploitation) nor too long (sailing times uncertainty increases in long time periods). Then, a minimum and a maximum trip duration and number of visits are introduced as constraints. Instances with up to 14 offshore platforms are considered with at most 3 of them with opening hours. Visits to platforms varies from 1 to 6 during T_H. Shyshou et al. [190] propose a large neighbourhood search for the problem.

Bendall and Stent [18] propose a model to determine cargo routes. Ships leave from the onshore supply depot and can visit once or more times a set of ports. The revenue needs to be maximized and is the difference between revenue earned transporting containers minus shipping costs. A set of possible trips is first determined and then a model is solved in order to determine how many times each trip is performed (possibly zero) in order to maximize profit. Then trips are heuristically packed into journeys. A simple case based on South East Asia is given as an example.

3.5 Production, inventory and multi-trip transportation problems

In the Production and Transportation Scheduling Problem (PTSP) one has to solve the production scheduling and the transportation routing problems simultaneously. The production and transportation phases are not considered as separate tasks, but the entire process is optimized. The interested reader is referred to the recent review on PTSP by Chen [37]. The problem is particularly important in make-to-order policies, where companies produce customized products and deliver them to clients within short time. No (or almost not) inventory is considered and then handling both production and transportation is crucial. Here we concentrate on cases where in the delivery phase, vehicles perform several trips.

Van Buer at al. [210] consider the production and distribution of newspapers. Overnight production of different newspapers is delivered to drop-off point by trucks. The objective is to minimize strategic costs (buying and maintaining the fleet of vehicles) and operational costs, while respecting delivering time windows associated with each drop-off point. When a truck gets back to the facility before production of a next trip is started, it is considered for multiple usage. An analysis is conducted on the benefit of re-using trucks on a real instance. The number of trucks needed to perform delivery decreases from 10 to 2 when re-usage is allowed. Due to the nature of the objective function, costs are then drastically decreased.

Chang and Lee [35] consider problems where one or two machines are available for production and one vehicle performs deliveries to one or two areas. Simple heuristics and worst case analysis is proposed for the different problems.

In Gesimar et al. [83] a plant produces a product with a short lifespan B, i.e., the product needs to arrive at customer location not later than B time units after production has terminated. Due to time and vehicle capacity constraints, the only vehicle in charge of distribution, needs to perform several trips in order to satisfy all the clients. Trips are first determined by means of the Split procedure (Prins [166]) and then re-ordered to minimize the total makespan. If production for trip i takes longer than trip $i-1$, production for trip i is scheduled right after production for trip $i-1$. On the other side, production is postponed to have product for trip i ready exactly at the time the vehicle is back at the depot. A final phase attempts to minimize the makespan anticipating production while respecting lifetime constraints. All these steps are embedded in a genetic algorithm.

In Ullrich [208] identical parallel machines produce goods that are delivered to customers by a fleet of vehicles that are allowed to perform several trips. Time windows associated with customers have hard lower bounds and soft upper bounds, and the objective is the minimization of the total tardiness at customer locations, i.e., $\sum_{i=1}^{N} \max\{t_i - L_i, 0\}$, where t_i indicates the instant vehicle reaches customer i. Benefits of simultaneous consideration of scheduling and routing problems are shown comparing optimal results with those obtained by two decomposition approaches. Moreover, a genetic algorithm is designed to solve the whole problem.

In inventory routing problems each customer consumes a product at a certain rate per unit of time. The supplier needs to determine, over a horizon consisting in T time periods, when to visit and the quantity to deliver to each customer at each visit, insuring that no stock-outs take place. Moreover, the supplier needs to organize customer visits into vehicle trips. A fleet of vehicles is available to the supplier, and a common assumption is that each vehicle can perform one trip per period of time. The interested reader is referred to the recent survey on Inventory Routing Problems by Coelho et al. [43] and to Bertazzi et al. [21].

Aghezzaf et al. [1] introduces the possibility of each vehicle to perform several trips during a time period. A column generation approach is proposed where columns represent journeys. Reduced cost columns are heuristically seek, turning the method not to be exact. Instances with 25 to up to 200 customers are created and the benefit of introducing the multi-trip aspect is confirmed by results showing savings between 12 and 16% of the total cost (vehicle usage, transportation, inventory and serving costs). A limitation in the driving time per period is considered in Raa and Aghezzaf [176]. In Raa and Aghezzaf [175] a long-term plan is made based on delivery patterns. The solution method can be summarized in four steps: first, customers are assigned to vehicles, then, for each vehicle a set of trips is created serving the different customers. Then, frequency of trips and finally trips are scheduled into journeys. In all these three papers, the size of the fleet need to be minimized, and then vehicle costs are included in the objective function.

3.6 The MTVRP in practice

This section lists the papers whose study is motivated by a real case study. Some of them could have been inserted in one of the previous section as a variant of the considered problem. However, we preferred to insert them in a particular section, to emphasize the deep practical motivation of the research and to show the practical interest of the *multi-trip* aspect.

In Brandão and Mercer [24] the delivery planning of British biscuits company need to be computed. Vehicles have different capacities, access to some customers is restricted to some vehicles (site dependent aspect), service must take place within time windows. Moreover, legal driver breaking times and unloading times are considered. A tabu search algorithm is proposed. This work is extended in Brandão and Mercer [25] for comparison with benchmark instances purposes (see Section 3.2).

Tung and Pinnoi [207] study a real waste collection problem in the city of Hanoi. Garbage is collected manually at house or industry locations by pushed handcarts and brought to collection points. Trucks leave from a depot and go to collection points where handcarts are unloaded into trucks. Then, handcarts start a new collection trip, while trucks go to the next collection point until they are (almost) loaded to capacity. They then go to the landfill where they are emptied. Then, another collecting trip starts, that has the landfill as origin and destination. After the last trip, trucks go back to the depot. Multi-trip aspect in the sense of this survey, arises at the handcart level. The authors focus only on the truck level. Additional characteristic of the problem is the presence of a minimum interval time between two consecutive visits to the same collecting point, due to operation time that occurs from collecting garbage by handcarts. Moreover, several time windows are associated with each collecting point. Handcarts arrive at the collection point at the beginning of a time window and wait until a truck arrives to collect garbage during the corresponding time window. First, the initial solution is constructed by means of a modified I1 heuristic (Solomon [196]) and improved by modified Or-opt and 2-opt*.

Gribkovskaia et al. [95] study the transport of live animals to slaughterhouses. Collection planning of animals at farms is made on a multi day base, prohibiting overnight trips and limiting the number of trips to three per day. Constraints taking into account animal welfare are considered as time limitations between the moment animals are loaded into the vehicle and the time the vehicle arrives at the slaughterhouse. A mathematical formulation of the problem is proposed, but no computational results are reported.

In Battarra et al. [17] different commodities have to be distributed to supermarkets located in a regional territory. Commodities are *incompatible*, i.e., they cannot be transported together by the same vehicle (even if each vehicle can carry all the products). Time windows are considered as well as duration limit on journey length: the difference between the time the last trip arrives at the depot and the start time of the first trip cannot exceed a certain value (representing the working day). Finally, the objective requires the minimization of the fleet size, breaking tails in favor of solutions with lower routing cost. They propose a guided search algorithm. Results are outperformed by the Iterated Local Search proposed by Cattaruzza et al. [34].

Lei et al. [131] seek to minimize production, inventory and distribution costs of a chemical company that produces road maintenance chemicals in different plants and ship products to customers in North-America. A fleet of vehicles is associated with each plant and trucks are allowed to perform several serving trips. In a first phase a mixed-integer program is solved where vehicles are allowed to serve customers only via direct shipment. In a second phase shipments are heuristically consolidated.

Lin and Kwok [133] study the case of a telecommunication company located in Hong-Kong that needs to deliver bills to its customers. Depots need to be chosen among a set of possible sites and serving trips need to be planned in order to serve all the customers. Total costs as well as loading and journey working times imbalance are minimized. The problem that arises is a location-routing problem with vehicles allowed to perform several trips. A tabu search and a simulated annealing heuristic are developed, both in two versions: one that creates journey made of one trip and one that allows journey to be composed by several trips.

In automated guided vehicle systems, vehicles deliver material to warehouses along fixed paths: the different points to serve are in a predefined sequence. The (only) decisions is needed to be taken regard vehicles depot returns for replenish purposes. The single vehicle case is studied in Tsirimpas et al. [206]. Three different problems are treated concerning diverse loading policies. Algorithms based on dynamic programming are developed.

In the Petrol Station Replenishment Problem (PSRP) introduced by Cornillier, et al. [46], a set of petrol stations need to be replenished by a fleet of heterogeneous vehicles. Each station requires a certain quantity of each product that lies between given minimum and maximum values. Each vehicle has different compartments that can be filled with any of products at once. Other side constraints are present as the obligation of emptying the front vehicle's compartment last and a maximum trip time limitation. A limitation on regular and overtime driving hours is given and the wage is different in the two cases. It is required to determine the quantities to deliver to each station, the loading of these products into compartments and to route the vehicles in order to maximize profits. The problem is extended to the multi-period case in Cornillier et al. [45]. Vehicles are allowed to perform several trips during each period. A multi-phase heuristics is proposed that for each period constructs trips and packs them into vehicles. Time windows and the multi-trip aspect are introduced in Cornillier et al. [47]. Two heuristics are proposed. The first heuristic is based on arc preselection that reduces the dimension of the network. The second heuristic divides the region in sectors and determines trips for each of these sectors. Finally, it recomposes them to form a solution. The problem is extended to the multi-depot case in Cornillier et al. [48] and is solved heuristically. Avella et al. [9] study a particular case of the problem where compartments must travel either completely full, or completely empty. They propose a heuristic procedure and a Branch-and-Price exact method. Instances from a company with 6 vehicles and around 25 customers per day are solved. The heuristic and exact methods produce solutions that cost respectively 12-15% and 22.25% less than the manual solutions provided by the company.

Taniguchi and Shimamoto [199] and Taniguchi and Van Der Heijden [201] allow vehicle to perform several trips in their simulation model to evaluate the benefits of traffic information systems, cooperation among companies and other city logistics initiatives.

3.7 Conclusions and perspectives

In this paper we presented the first complete survey on the *multi-trip* vehicle routing problem more than 25 years after that the concept was introduced by Salhi [185] and almost 25 years after the problem was formalized by Fleischmann [78].

The literature review has shown a clear lack in the development of exact methods for this problem. Only one method has been proposed for the MTVRP. It is characterized by the need of an initial feasible solution. Hence, it cannot provide an answer regarding the existence of such a feasible solution for the 5 instances in which all the developed heuristics have failed in providing a feasible planning.

The situation in the presence of time windows is similar. Three algorithm have been developed, two of which for the case with trip duration limitation. Moreover, the method for the MTVRPTW fails in solving instances with only 25 customers and 2 vehicles, instances with 50 customer and 4 vehicles being the biggest that can currently be closed. This underlines the need for further research on the subject, looking for more efficient methods.

The situation does not improve when the MTVRPTW with limited trip duration is considered. The limited trip duration limits the number of feasible trips, but some instances with 25 customers cannot be solved. The largest instance currently solved contains 40 customers. In maritime problems the case is different. The characteristics of the problem, such as a limited number of visits per trip, limit strongly the number of feasible trips. Moreover, instances are usually smaller than routing problems. This makes enumeration methods efficient.

Finally, this article has shown that the MTVRP and its variants are of practical interest. Moreover, some problems including production and inventory start now to be explored. These considerations should motivate researchers in pursuing further research in this area.

Chapter 4

A Memetic Algorithm for the Multi Trip Vehicle Routing Problem

Abstract

We consider the Multi-Trip Vehicle Routing Problem, in which a set of geographically scattered customers have to be served by a fleet of vehicles. Each vehicle can perform several trips during the working day. The objective is to minimize the total travel time while respecting temporal and capacity constraints.

The problem is particularly interesting in the city logistics context, where customers are located in city centers. Road and legal restrictions favor the use of small capacity vehicles to perform deliveries. This leads to trips much shorter than the working day. A vehicle can then go back to the depot and be re-loaded before starting another service trip.

We propose an hybrid genetic algorithm for the problem. Especially, we introduce a new local search operator based on the combination of standard VRP moves and swaps between trips. Our procedure is compared with those in the literature and it outperforms previous algorithms with respect to average solution quality. Moreover, a new feasible solution and many best known solutions are found.

4.1 Introduction

The well known Vehicle Routing Problem (VRP) is an $\mathcal{NP}$-hard combinatorial optimization problem where a set of geographically scattered customers has to be served by a fleet of vehicles. An implicit assumption of the VRP is that each vehicle can perform only one route in the planning horizon. This assumption is not realistic in several practical situations. For the distribution of goods in city centers, for example, small vehicles are generally preferred. Because of this capacity limitation, they daily perform several short tours. This problem is referred to as the Multi-Trip VRP (also VRP with multiple use of vehicles, Taillard et al. [198], VRP with multiple trips, Petch and Salhi [161] or VRP with multiple routes, Azi et al. [11]). In the rest of the paper it will be indicated as MTVRP.

The MTVRP is defined on an undirected graph $G = (V, E)$, where $V = \{0, 1, \dots, N\}$ is the set of vertices and $E = \{(i, j) | i, j \in V, i < j\}$ is the set of edges. It is possible to travel from i to j, incurring in a travel time T_{ij}. Vertex 0 represents the depot where a fleet of M identical vehicles with limited capacity Q is based. Vertices $1, \ldots, N$ represent the customers to be served, each one having a demand Q_i. A time horizon T_H exists, which establishes the duration of the working day. Overtime is not allowed. It is assumed that Q, Q_i and T_H are nonnegative integers.

The MTVRP calls for the determination of a set of routes and an assignment of each route to a vehicle, such that the total travel time is minimized and the following conditions are satisfied:

(1) each route starts and ends at the depot,

(2) each customer is visited by exactly one route,

(3) the sum of the demands of the customers in any route does not exceed Q,

(4) the total duration of the routes assigned to the same vehicle does not exceed T_H.

It is also supposed that each customer i could be served by a return trip, i.e, $T_{0i} + T_{i0} \leq T_H$ and $Q_i \leq Q$.

Few papers in the literature address the MTVRP and no efficient population-based algorithm were proposed. Our goal is to fill this gap proposing a memetic algorithm able to compete with previous works. Our interest in the MTVRP raises from the MODUM project[1], where mutualized distribution in city centers is explored. The contribution of this paper is threefold: 1) A high-performance memetic algorithm is proposed: the results found are the new state-of-the-art on classical instances for the MTVRP. Moreover, an instance has been solved for the first time, i.e., a feasible solution has been found; 2) An adaptation of the Split procedure (Prins [166]) to segment a chromosome into a MTVRP solution is developed; 3) A new local search (LS) operator, that combines standard VRP moves and re-assignment of trips to vehicles is introduced.

This paper is organized as follows. In Section 4.2 the literature on the MTVRP is reviewed. Section 4.3 describes the proposed algorithm. Section 4.4 details the *Combined* LS. Results are reported in Section 4.5. Conclusions and perspectives are discussed in Section 4.6.

4.2 Literature review

The well known VRP has been deeply studied in the last 50 years and many exact and heuristic methods have been proposed in the literature (see Toth and Vigo [204] and Golden et al. [90]). However, exact methods remain limited to problems with restricted size, i.e., less than 100 customers. Moreover, many different variants of the problem are introduced in order to face particular constraints that arise in everyday applications. Despite that, MTVRP has been investigated only in the last two decades and the literature is still scarce.

Fleischmann [78] was the first to address the problem in his working paper in 1990. He proposes a modification of the savings algorithm and uses a bin packing (BP) problem heuristic to assign routes to the vehicles. In Taillard et al. [198], VRP solutions are generated using

[1] http://www-lipn.univ-paris13.fr/modum

a tabu search (TS) algorithm with adaptive memory (Taillard [197]). The routes forming the VRP solutions are stored in a list. From that list a subset of routes is selected and a MTVRP solution is constructed using a BP heuristic. A benchmark of instances (constructed from VRP instances) is proposed. It will be used as efficiency comparison for all the authors that have developed a solution method for the MTVRP. Curiously, Taillard et al. [198] provide values only when the algorithm fails in finding a feasible solution, introducing an arbitrary penalization factor $\theta = 2$ for the overtime. Next papers follow the same scheme except Salhi and Petch [186] (Olivera and Viera [156] do not provide exact values, but just a GAP measure from a reference value as it will be explained in Section 4.5). Petch and Salhi [161] propose a multi-phase algorithm with the minimization of the overtime as objective function. A pool of solutions is constructed by the parametrized Yellow's savings algorithm (Yellow [219]). For each solution in the pool, a MTVRP solution is constructed using a BP heuristic. The MTVRP solutions are improved using 2-opt, 3-opt moves, combining routes and reallocating customers. In Salhi and Petch [186], as in Petch and Salhi [161], the maximum overtime is minimized. A genetic algorithm is proposed. In this method the plane is divided into circular sectors. Each sector is defined by an angle measured with respect to the depot and the x axis. A chromosome is the sequence of such angles in non-decreasing order. Clusters are created by assigning each customer to the sector it occupies. In each cluster, the Clarke and Wright savings heuristic is used to solve a smaller VRP problem. The resulting routes are packed using a BP heuristic. Olivera and Viera [156] use an adaptive memory approach to tackle the MTVRP. A memory $\mathcal{M}$ is constructed with different routes that form VRP solutions generated with the sweep algorithm. Each route is labeled with its overtime value and its cost and are sorted using a lexicographic order. New VRP solutions are generated by probabilistically selecting routes in $\mathcal{M}$ and improved by a TS algorithm. New VRP solutions are used to update $\mathcal{M}$. From the best VRP solution a MTVRP solution is obtained using a BP heuristic. Recently, Mingozzi et al. [140] propose an exact method for the MTVRP based on two set partitioning-like formulations. 52 instances with up to 120 customers and with a known feasible solution (without overtime) are tackled and in 42 cases the optimal solution is found.

Alonso et al. [3] consider the site-dependent periodic MTVRP. Each customer has to be served up to t times in a planning horizon of t periods. Moreover, not every vehicle can serve all the customers. To each customer is assigned a delivery pattern and it is assigned to a vehicle using GENIUS heuristic (Gendreau et al. [85]). If the insertion violates time or capacity constraints, a new route is initialized. Two moves are used to improve the solution: customers are moved from a route to another and different patterns are assigned to a customer.

The MTVRP with time windows (MTVRPTW) is addressed as well. Several exact methods are proposed (Azi et al. [11], Hernandez et al. [106]). Instances with 100 customers and 1 vehicle (Azi et al. [11]) and with 50 customers and 4 vehicles (Hernandez et al. [106]) can be solved to optimality.

Different studies facing practical cases envisage to perform several trips during the working day. For example, Brandão and Mercer [24] consider a MTVRPTW that arises from the biscuit distribution of a British company. Vehicles have different capacities, in case of need they can be hired from the company and the access to some customers is restricted to particular vehicles. Drivers' schedule must respect the maximum legal driving time per day. Legal time breaks and unloading times are taken into account. Real instances including 45 to 70 customers and the use of 11 vans and 11 tractors are considered. In their subsequent work, Brandão and Mercer [25] adapt the algorithm to compare the results with those obtained by Taillard et al. [198]. A two phases TS is performed. In the first phase, a solution is allowed to become infeasible regarding travel time constraints, but in the second phase, only feasible solutions are accepted. Insert and swap moves are considered. Battarra et al. [17] consider the MTVRPTW and different commodities that cannot be transported together. The objective is to minimize the number of used vehicles. The problem is decomposed in simpler subproblems, one for each commodity. A set of routes is then generated for each commodity and packed by means of a BP heuristic in order to obtain a solution. Data comes from real-world instances where goods are delivered to supermarkets placed in a regional territory. The concept of multi-trips is also addressed by Cornillier et al. [47] and Gribkovskaia et al. [95]. The former paper concerns the petrol distribution to gas stations, while the latter proposes a model for the livestock collection.

The idea of multi-trip is found in the context of city logistics as well. For example, Taniguchi and Shimamoto [199] propose a model to evaluate the impact of advanced information system in urban areas and they assume that vehicles are allowed to perform multiple trips per day. Browne et al. [29] present the case of supplies company operating in the City of London. From a micro-consolidation urban center, electrically assisted cargo tricycles and electric vans perform deliveries. Due to the small size of tricycles and electric vans, they perform several trips during each day.

4.3 A memetic algorithm for the MTVRP

Genetic algorithms (GA) are adaptive methods inspired from the natural evolution of biological organisms. An initial population of individuals (*chromosomes*) evolves through generations until satisfactory criteria of quality, a maximum number of iterations or time limits are reached. New individuals (*children*) are generated from individuals forming the current generation (*parents*) by means of genetic operators (*crossover* and *mutation*). The principles of genetic procedure were firstly formalized by Holland [109] and have been successfully used in different contexts (Neri and Cotta [147]). The papers of Prins [166] and Vidal et al. [213] are two examples of efficient GA (the former for the VRP and the latter for the multi depot VRP and the periodic VRP) in the VRP field. In particular, GAs allow for a diversified exploration over the search space due to the management of several solutions at the same time. When Local Search (LS) algorithms are part of the procedure, the GA is commonly called memetic algorithm (MA). For an overview of GAs and MAs the reader is respectively referred to Reeves [177] and Moscato and Cotta [142].

In this section the proposed MA for the MTVRP is described. It makes use of an adaptation of the Split procedure (Prins [166]) to obtain a MTVRP solution from *giant tours* (Section 4.3.2). The population diversity management is inspired by the work of Vidal et al. [213]: for survival, individuals are selected according to their quality and their contribution to the diversification of the population (Section 4.3.6). A sketch of the method is given in Algorithm 1.

A new advanced feature is embedded in the LS: when a pejorative move is detected, it is tested in combination with a re-assignment of trips. In case of improvement, both the move and the re-assignment are performed (Section 4.4).

Algorithm 1 Memetic Algorithm outline

```
Initialize population (Section 4.3.5)
while Termination criterion is not met do
    Select parent chromosomes Ψ_{P_1} and Ψ_{P_2} (Section 4.3.3)
    Generate a child Ψ_C (Section 4.3.3)
    Educate Ψ_C (Section 4.3.4)
    if Ψ_C is infeasible then
        Repair Ψ_C (Section 4.3.4)
    end if
    Insert Ψ_C in the population
    if Dimension of the population exceeds a given size then
        Select survivors (Section 4.3.6)
    end if
end while
```

4.3.1 Solution representation and search space

A chromosome is a sequence (permutation) $\Psi = (\Psi_1, \ldots, \Psi_N)$ of N client nodes, without trip delimiters. Ψ can be viewed as a TSP solution that has to be turned in a feasible MTVRP solution by splitting the chromosome (inserting trip delimiters and assigning trips to vehicles). From that point of view, Ψ is usually called a *giant tour*. From a giant tour Ψ, different MTVRP solutions can be constructed depending on the way Ψ is split.

During the search phase, overtime and overload are allowed and penalized in the fitness function with factors θ and λ respectively, even though a feasible solution is required.

A procedure *AdSplit* (explained in Section 4.3.2) is used to get a MTVRP solution ξ from Ψ. The following notation is introduced: $T_v(\xi)$ and $O_v(\xi) = \max\{0, T_v(\xi) - T_H\}$ are respectively the travel time and the overtime of vehicle v in solution ξ. $L_r(\xi)$ is the load of route r and $r \in v$ indicates that route r is assigned to vehicle v. The fitness $F(\Psi)$ of the chromosome Ψ is the cost of the best solution ξ found by *AdSplit* and it is defined as

$$F(\Psi) = c(\xi) = \sum_{v=1}^{M} T_v(\xi) + \theta \sum_{v=1}^{M} O_v(\xi) + \lambda \sum_{v=1}^{M} \sum_{r \in v} \max\{0, L_r(\xi) - Q\} \tag{4.1}$$

When confusion cannot arise, solution ξ will be omitted in the notation. The chromosome Ψ is called *feasible* (*infeasible*) if *AdSplit* obtains, from Ψ, a feasible (infeasible) solution ξ.

4.3.2 A Split algorithm for the multi-trip problems

The splitting procedure proposed here, called *AdSplit*, is an adaptation of the procedure proposed by Prins in [166]. It is used to turn a chromosome into MTVRP solutions. The cost of the solution obtained by *AdSplit* is associated with the chromosome itself as fitness value in order to evaluate its quality. The procedure is used each time a new individual is generated, either randomly (at the beginning for initialization purposes, Section 4.3.5), or after mating parents to generate children by means of crossover operators, Section 4.3.3.

Auxiliary graph construction

The splitting procedure works on an auxiliary graph $H = (V', A')$. V' contains $N+1$ nodes indexed from 0 to N. Arc (i, j), $i < j$, represents a trip serving customers from Ψ_{i+1} to Ψ_j in the order they are in Ψ. With each arc (i, j), is associated a cost c_{ij} defined as

$$c_{ij} = \tau_{ij} + \theta \max\{0, \tau_{ij} - T_H\} + \lambda \max\{0, L_{ij} - Q\} \tag{4.2}$$

where τ_{ij} and L_{ij} represent respectively the trip travelling time and the sum of customer's requests served during the trip.

A simple example with five customers is given in Figures 4.1–4.2. $\Psi = (1, 2, 3, 4, 5)$, $T_H = 45$, $Q = 50$, $\theta = \lambda = 2$ and the demand of each customer is given between brackets. For example, arc $(1, 5)$ in Figure 4.2 represents the trip serving customers from 2 to 5. $\tau_{15} = 116$, $L_{15} = 76$. The arc cost is then $c_{15} = 116 + 2 \cdot (116 - 45) + 2 \cdot (76 - 50) = 310$.

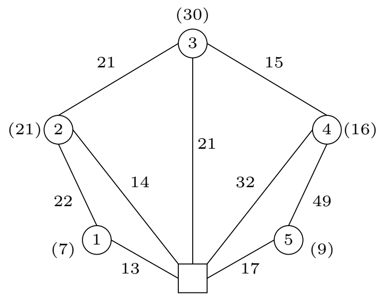


Figure 4.1: Example with 5 customers: demands in brackets, $T_H = 45$, $Q = 50$, $\theta = \lambda = 2$

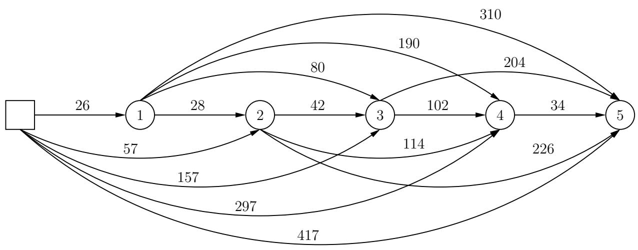


Figure 4.2: Auxiliary graph (to each arc (i, j), cost c_{ij} is assigned as defined in Equation 4.2)

Once H is computed, paths basically represent set of trips that can be assigned to vehicles. In the VRP context, an optimal splitting is equivalent to a shortest path (SP) in H each arc representing a route which is assigned to a vehicle. Since H is acyclic, Bellman's algorithm can be used to find the SP in $O(N^2)$. In the MTVRP context, more than one trip can be assigned to the same vehicle. The procedure proposed in Prins [166] cannot be directly used and is modified as explained in Section 4.3.2.

Assignment procedure

The assignment procedure both selects and assigns trips to vehicles. It consists of two phases. In the first phase, the SP in H is computed. In the second phase, trips of the SP are assigned to vehicles by means of a labelling algorithm. The labelling algorithm works as follows.

Starting from node 0, labels are progressively extended along the graph defined by SP. Each label $\mathcal{L}$ has $M+3$ fields: the first M fields store vehicle travel times in decreasing order (enhancing the strength of the dominance rule), the $(M+1)^{\text{th}}$ field memorizes the total load infeasibility, the $(M+2)^{\text{th}}$ the predecessor node, and the last field keeps the cost of the partial solution evaluated using Equation 4.1 and equivalent to the cost $c(\mathcal{L})$ of label $\mathcal{L}$. When extending a label, M new labels are constructed, one for each possible allocation of the new trip to a vehicle. When node N is reached, the label $\mathcal{L}$ with minimum cost $c(\mathcal{L})$ associated with node N is selected and the related solution is constructed.

Dominated labels, accordingly with the following dominance rule, are discarded: let $\mathcal{L}^1$ and $\mathcal{L}^2$ be two labels associated with the same node i. $\mathcal{L}^1$ dominates $\mathcal{L}^2$ if and only if

$$c(\mathcal{L}^1) + \theta \sum_{j=1}^{M} \delta_j(\mathcal{L}^1, \mathcal{L}^2) \leq c(\mathcal{L}^2) \tag{4.3}$$

where $c(\mathcal{L})$ is the cost associated with label $\mathcal{L}$,

$$\delta_j(\mathcal{L}^1, \mathcal{L}^2) = \max \left\{ 0, \min \left\{ T_H, T_j(\mathcal{L}^1) \right\} - \min \left\{ T_H, T_j(\mathcal{L}^2) \right\} \right\}$$

and $T_j(\mathcal{L})$ is the (partial) travel time of vehicle j associated with label $\mathcal{L}$. Roughly speaking, given two labels $\mathcal{L}^1$ and $\mathcal{L}^2$, extending $\mathcal{L}^1$ is penalized as much as possible while it is not extending $\mathcal{L}^2$ in the same way. If Inequality 4.3 holds, $\mathcal{L}^2$ cannot be extended in a better way than $\mathcal{L}^1$, and it is eliminated.

The procedure is illustrated for the simple example in Figure 4.1. First, SP is calculated on the corresponding auxiliary graph H. The solution is then constructed with the assignment procedure just explained. The SP and the solution obtained are depicted in Figure 4.3.

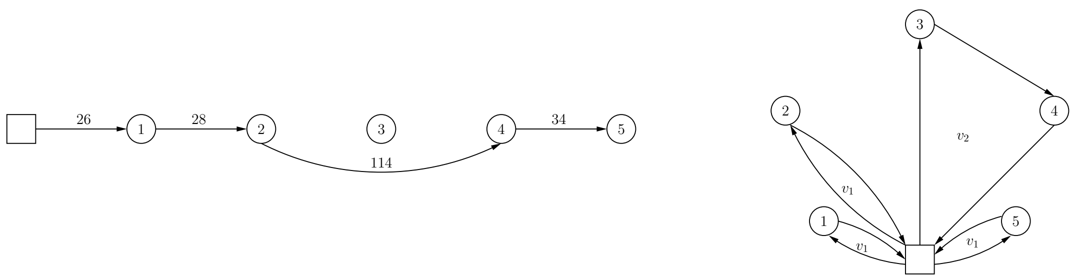


Figure 4.3: MTVRP solution obtained from arcs forming the shortest path ($F(\Psi) = 288$)

Label's extension is reported in Table 4.1. The first line reports the node. Due to space limitations, labels associated with nodes 0 and 3 are not reported: only the null label is associated with node 0, while node 3 is not connected. Moreover, the predecessor of each label is straightforward and load infeasibility is zero for each label considered. Thus, they are omitted. Column *dom* indicates whether the corresponding label is dominated by another label associated with the same node.

Note that this approach provides the optimal assignment of trips in SP, but is suboptimal with regard to the decomposition of Ψ, as illustrated in Figure 4.4.

1				2				4				5			
v_1	v_2	c	dom	v_1	v_2	c	dom	v_1	v_2	c	dom	v_1	v_2	c	dom
26	0	26	no	28	26	54	no	96	26	224	yes	102	54	328	yes
				54	0	72	no	94	28	220	yes	88	68	288	no
								122	0	276	yes				
								68	54	184	no				

Table 4.1: Labels associated with nodes of H during the assignment procedure

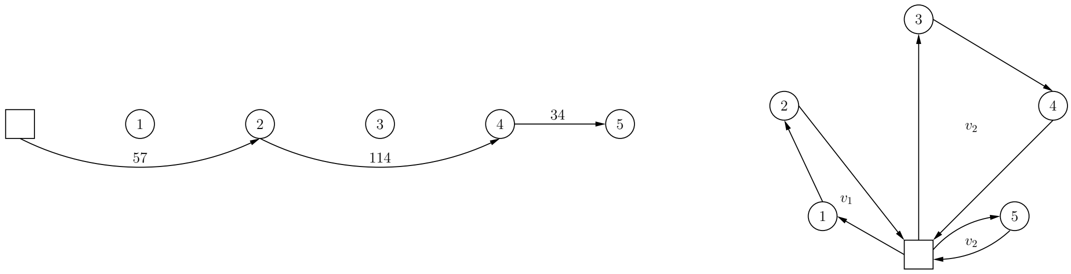


Figure 4.4: Best MTVRP solution for Ψ ($F(\Psi) = 273$)

Applying the procedure on the complete graph H, the label that minimizes Equation 4.1 at node N would correspond to the best decomposition of Ψ in the MTVRP context. One could however expect that a huge number of labels would need to be treated, which do not appear to be viable in the MA context.

Improving the Split procedure

Arc (i, j) in the auxiliary graph H represents the trip serving customers $(\Psi_{i+1}, \Psi_{i+2}, \ldots, \Psi_j)$ in the order they appear in the giant tour Ψ. Visiting customers in a different order can lead to a trip with a smaller cost. Let suppose the distance between customers 1 and 4 in the example shown in Figure 4.1 is equal to 20. The trip $(0, 2, 3, 4, 1, 0)$ would cost 207 that together with trip $(0, 5, 0)$ leads to a solution of cost 241 (Figure 4.5). As proposed by Prins

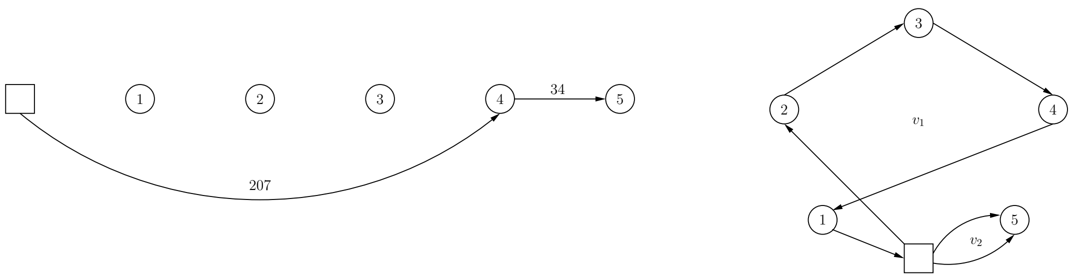


Figure 4.5: MTVRP solution obtained considering rotations of customers in the same trip ($F(\Psi) = 241$)

et al. [168], each rotation (circular left shit) can be considered and evaluated in constant time. For example, a one-position rotation corresponds to the trip $(0, \Psi_{i+2}, \ldots, \Psi_j, \Psi_{i+1}, 0)$. Then, given an arc, all the possible rotations are considered looking for the best starting point of the trip without introducing any computational burden (see Prins et al. [168] for a detailed explanation).

A pseudo-code sketch of the *AdSplit* procedure is proposed in Algorithm 2. Procedure *SP_best_in()* computes the shortest path on graph H, taking into consideration the best rotation for each arc. With each node i, it associates its successor $succ_i$, the travelling time and load of the trip represented by $(i, succ_i)$. These values are obtained when needed by procedures *get_successor(i)*, *get_best_in_time(i)* and *get_load(i)*. $\mathcal{L}_k$ indicates the k^{th} field of label $\mathcal{L}$, while $\mathcal{L}_{lInf}$, $\mathcal{L}_{pred}$ and $\mathcal{L}_c$ refer respectively to the $(M+1)^{\text{th}}$, $(m+2)^{\text{th}}$ and $(m+3)^{\text{th}}$ label fields. $sort(\mathcal{L})$ sorts the first m fields in decreasing order. If $\mathcal{L}$ is dominated by a label in $ListLabel_i$, $is\_dominated(ListLabel_i, \mathcal{L})$ returns `true`, otherwise it returns `false`. Labels in $ListLabel_i$ dominated by the new inserted label $\mathcal{L}$ are eliminated from the list by $eliminate\_dominated\_labels(ListLabel_i, \mathcal{L})$.

Algorithm 2 *AdSplit*

$SP\_best\_in()$
for $i = 0$ to N **do**
    $LabelList_i = \emptyset$
end for
$LabelList_0 \leftarrow (\overbrace{0, \ldots, 0}^{M}, 0, 0, 0)$
$current = 0$
while $current < N$ **do**
    $succ = get\_successor(current)$
    $load = get\_load(current)$;
    $time = get\_best\_in\_time(current)$
    for all $\mathcal{L} \in LabelList_{current}$ **do**
        for $k = 1 \rightarrow M$ **do**
            $\mathcal{L}^* = \mathcal{L}$
            $\mathcal{L}^*_k = \mathcal{L}_k + time$
            $sort(\mathcal{L})$
            $\mathcal{L}^*_{lInf} = \mathcal{L}_{lInf} + \lambda \cdot \max\{load - Q, 0\}$
            $\mathcal{L}^*_c = \mathcal{L}_c + time + \theta \cdot \max\{\mathcal{L}^*_k - T_H, 0\} - \theta \cdot \max\{\mathcal{L}_k - T_H, 0\} + \lambda \cdot \max\{load - Q, 0\}$
            $\mathcal{L}^*_{pred} = current$
            if not $is\_dominated(ListLabel_{succ}, \mathcal{L}^*)$ **then**
                $ListLabel_{succ} \leftarrow \mathcal{L}^*$
                $eliminate\_dominated\_labels(ListLabel_{succ}, \mathcal{L}^*)$
            end if
        end for
    end for
    $current = succ$
end while

4.3.3 Crossover

The classic OX operator is used. Figure 4.6 shows how the OX works. First, two cutting points have to be chosen in the parents Ψ_{P_1} and Ψ_{P_2}. In the example they are $i = 4$ and $j = 7$. Indicating with $\Psi_{C_1} = OX(\Psi_{P_1}, \Psi_{P_2})$ the first child, $\Psi_{C_1}(k) = \Psi_{P_1}(k)$ for $k = i, \ldots, j$. Then, Ψ_{P_2} is circularly swept from $\Psi_{P_2}(j+1)$ onward inserting in Ψ_{C_1} the missing nodes. By inverting the roles between Ψ_{P_1} and Ψ_{P_2}, we obtain the second child $\Psi_{C_2} = OX(\Psi_{P_2}, \Psi_{P_1})$.

$$
\begin{array}{lccc|cccc|ccc}
 & & & & \overset{i=4}{\downarrow} & & & \overset{j=7}{\downarrow} & & & \\
\Psi_{P_1}: & 2 & 6 & 4 & 7 & 8 & 5 & 10 & 9 & 3 & 1 \\
\Psi_{P_2}: & 3 & 7 & 9 & 2 & 4 & 10 & 1 & 6 & 5 & 8 \\
 & & & & & & & & & & \\
\Psi_{C_1}: & 2 & 4 & 1 & 7 & 8 & 5 & 10 & 6 & 3 & 9 \\
\Psi_{C_2}: & 7 & 8 & 5 & 2 & 4 & 10 & 1 & 9 & 3 & 6
\end{array}
$$

Figure 4.6: OX operator

Parents Ψ_{P_1} and Ψ_{P_2} are selected with the classic *binary tournament* method: two chromosomes are randomly drawn from the population and the one with the lower fitness is selected. The procedure is repeated twice, once for the selection of each parent. The child that has to be inserted in the population is randomly selected between children $\Psi_{C_1} = OX(\Psi_{P_1}, \Psi_{P_2})$ and $\Psi_{C_2} = OX(\Psi_{P_2}, \Psi_{P_1})$.

4.3.4 Local search - education and repair procedures

After crossover, the obtained child is evaluated by means of *AdSplit* procedure, and *educated* applying LS procedure with a probability p_{LS} trying to improve its quality. LS is usually used in literature as mutation operator in order to obtain a high-performance hybrid GA.

The operators listed in the following are used. Let u and z be two nodes and t and x be their respective successors (that could be the depot as well). $R(u)$ indicates the route visiting customer u. The following simple types of moves are tested

- m1 If u is a client node, remove u and insert it after z;
- m2 If u and t are clients, remove them and insert u and t after z;
- m3 If u and t are clients, remove them and insert t and u after z;
- m4 If u and z are clients, swap u and z;
- m5 If u, t and z are clients, swap u and t with z;
- m6 If u, t, z and x are clients, swap u and t with z and x;
- m7 If $R(u) = R(z)$, replace (u, t) and (z, x) by (u, z) and (t, x);
- m8 If $R(u) \neq R(z)$, replace (u, t) and (z, x) by (u, z) and (t, x);
- m9 If $R(u) \neq R(z)$, replace (u, t) and (z, x) by (u, x) and (t, z);
- m10 If $R(u) = R(z)$, create another route with all customers from u to z (or from z to u if z comes before u) and put it in a randomly drawn vehicle.

The nodes can belong to the same route or to different routes. Routes can either belong to the same vehicle or to different vehicles. Moves m1–m3 correspond to *insertion* moves,

moves m4–m6 to *swaps*, move m7 is the well known *2-opt* and moves m8, m9 are usually called *2-opt**.

Moves m1–m9 are those used in Prins [166]. If $u = z$ in m10, a new route with only customer u is created.

At the beginning of the LS with each type of move mi, $i = 1, \dots, 10$ is associated a weight $w_i = w$ and the status *active*. At each iteration the LS procedure probabilistically selects a move among the active moves. The probability of move mi to be chosen is w_i/W where $W = \sum_{i=1}^{10} w_i$. The selected move Mi is evaluated and the first improvement criterion is adopted. If the move fails, i.e., the current solution is a local optimum in the neighbourhood defined by Mi, Mi becomes *inactive* and cannot be selected anymore until another move succeeds. The LS terminates when all the moves are inactive, i.e., a local optimum in the neighbourhood defined by m1–m10 is reached.

After a fixed number of iterations ω (arbitrarily fixed to 100), the weights are updated accordingly to the number of successes. Precisely, $w_i = w_i + \frac{success_i}{attempts_i}$, where $success_i$ and $attempts_i$ indicate respectively the number of times move Mi succeeded and was performed ($attempts_i$ is usually not the same for all moves due to probabilistic selection). W is updated accordingly. Weights w_i can be viewed as a short-term memory, i.e., a move that historically succeeds more will have a higher probability to be chosen.

To speed up LS, granular search is implemented as proposed by Toth and Vigo [205]: a move is considered only when z is one of the n_{closest} closest customers of u (*filtering rule*).

Each time a solution ξ is obtained from chromosome Ψ by means of *AdSplit*, it is stored in four different N-size vectors that memorize in i^{th} position the predecessor, the successor, the vehicle and the route of customer i. The travel time of each vehicle and the load of each route are stored as well. In this way, moves m1–m9 are evaluated in constant time, while m10 is in $O(N)$. Then, given a solution ξ and defining its neighbourhood $N(\xi)$ by the set of moves m1–m10, it can be completely explored in $O(N^3)$ time (more precisely, in $O(N^2 \cdot N_{closest})$ with the usage of the filtering rule), although the neighbourhood defined by m1–m9 requires $O(N^2)$ operations to be explored.

After LS is applied, the educated chromosome can be either feasible or infeasible. In the latter case the *repair* procedure is applied with a probability p_{rep}. It consists in applying again LS with λ (load infeasibility penalization parameter) and/or θ temporarily multiplied by 10, regarding the nature of the infeasibility. If a feasible chromosome is obtained, it is inserted in the population, otherwise λ and/or θ are (temporarily) multiplied again by 10 and LS reapplied. The original chromosome is not discarded even if the repaired chromosome is feasible (Vidal et al. [213]). All the chromosomes obtained during LS and repair procedure are as well inserted in the population.

4.3.5 Population structure, initialization and stopping criteria

An ordered population Π of chromosomes is kept. A key value k_Ψ is associated with each chromosome Ψ and the population is sorted regarding the key value. k_Ψ corresponds to the fitness $F(\Psi)$ of Ψ multiplied by a penalization factor P. $P = 1$ if Ψ is feasible, $P = 1.5$ if Ψ is time-infeasible, $P = 2$ if Ψ is load-infeasible, $P = 3$ if Ψ is both load and time-infeasible. This is done in order to ensure the best feasible solution found so far corresponds to the chromosome in the first position of the population (infeasible individuals can cost less than the best feasible one) and in general to keep good quality individuals at the top of Π. Moreover, it allows to manage both feasible and infeasible chromosomes in the same population, differently from what is done, for example, in Vidal et al. [213], where the population is divided in two subpopulations, one for feasible and the other for infeasible chromosomes.

The initial population is formed of π random generated chromosomes evaluated with the *AdSplit* procedure and improved applying LS.

The procedure terminates after a maximum number of iterations has been performed. An iteration consists of generating a child Ψ_C crossing parents that undergoes evaluation (by means of *AdSplit*), education and eventually reparation procedures. Ψ_C is then inserted in the population. It can be noticed that the number of iterations correspond to the number of crossovers performed.

4.3.6 Survivor strategy

When the population reaches a maximum dimension, i.e., $\pi + \mu$, a survivor selection is performed as proposed by Vidal et al. [214], [213]. Survivor chromosomes are selected based on quality, i.e., on fitness $F(\Psi)$, *and* their diversity contribution $f(\Psi)$ defined as the average distance between Ψ and its n_c closest neighbours in Π (forming set N_c) as follows:

$$f(\Psi) = \frac{1}{n_c} \sum_{\Psi_1 \in N_c} D(\Psi, \Psi_1), \tag{4.4}$$

where $D(\cdot, \cdot)$ is the *broken pair* distance, that is the number of pairs of adjacent customers in Ψ that are broken in Ψ_1 (Prins [167]). $D(\cdot, \cdot)$ gives a measure on the amount of common arcs between Ψ and Ψ_1. A biased fitness $bF(\cdot)$ is calculated for each chromosome as follows:

$$bF(\Psi) = r_F(\Psi) + (1 - \frac{n_e}{|\Pi|}) r_f(\Psi) \tag{4.5}$$

where $r_F(\Psi)$ and $r_f(\Psi)$ are the ranks of chromosome Ψ calculated based on fitness F and function f defined in Equation 4.4 respectively, and n_e is a parameter that ensures elitism properties during selection (see Vidal et al. [213] for a formal proof).

4.4 Combined Local Search

To optimize the packing of routes into vehicles, we introduce the possibility of a re-pack of trips along with a *pejorative* move m among m1–m10 introduced in Section 4.3.4. By pejorative move, we mean a move that does not decrease the solution cost. The swap between trips (*Swp*) in different vehicles is used as re-assignment procedure.

To understand the idea of the Combined LS (CLS) consider Figures 4.7–4.9. The example involves three vehicles with up to three routes each and duration constraint $T_H = 100$ that is violated by the third vehicle (Figure 4.7). Move m involves routes r_2 and r_3 of vehicles v_1

and v_2 respectively and it leads to the configuration shown in Figure 4.8 with an increase in the solution cost of 5 units (due to the increase in routing cost). Since m is pejorative, it would be discarded by the LS procedure. However, with a different assignment of trips to vehicles, an improvement can be obtained. In the particular case, it consists of swapping r_2 in v_1 with r_2 in v_3 (Figure 4.9).

	r_1	r_2	r_3	T_v	θO_v
v_1	60	30		90	-
v_2	30	30	30	90	-
v_3	45	30	30	105	10
				cost:	295

Figure 4.7: Initial configuration

	r_1	r_2	r_3	T_v	θO_v
v_1	60	**25**		85	-
v_2	30	30	**40**	100	-
v_3	45	30	30	105	10
				cost:	300

Figure 4.8: Pejorative move. In bold trips involved in M

	r_1	r_2	r_3	T_v	θO_v
v_1	60	**30**		90	-
v_2	30	30	40	100	-
v_3	45	**25**	30	100	-
				cost:	290

Figure 4.9: After *Swp*. In bold trips involved in *Swp*

The goal of the CLS is to detect when the combination of moves m1–m10 along with a swap of two trips leads to a better solution and, in that case, to perform both the move *and* the swap.

For the sake of computing time, the main issue here is to avoid evaluating every possible combination of moves with swaps (indicated with m+*Swp*). We propose to limit the evaluations of m+*Swp* according to the following rule $\mathcal{R}_1$:

Rule 1 ($\mathcal{R}_1$)**.** *The evaluations of* m+Swp *is limited to those that would improve the solution even if the assignment of routes to vehicles is optimal before* m+Swp *is applied.*

Using the subsequent propositions, it is then possible to limit heavily the size of the neighborhood explored with $m + Swp$.

In the following, we will note respectively ξ, ξ^m, ξ^{m+Swp} the current solution, the solution after applying move m and the solution after performing *Swp* as shown in Figure 4.10. A vehicle without (with) overtime will be called *feasible* (*infeasible*).

$$\xi \xrightarrow{\quad m \quad} \xi^m \xrightarrow{\quad Swp \quad} \xi^{m+Swp}$$

Figure 4.10: Notation

It is noteworthy that *Swp* can modify overtime, but does not affect the traveling time and the load infeasibility of the solution. We state two propositions in order to justify the restriction of pejorative moves that are tested along with swaps. The first proposition identifies which *Swp* can be worth to be tested once m has been performed. On the other hand, the second proposition states which moves m can lead to a global improvement when combined with a *Swp*. We start discussing the choice of swaps.

Proposition 1. *Under rule $\mathcal{R}_1$, we can restrict the choice of the* Swp *as follows:*

1. Swp *involves (trips in) two different vehicles v_1 and v_2,*
2. *exactly one vehicle between v_1 and v_2 is feasible,*
3. *at least one vehicle between v_1 and v_2 has to have been involved in m.*

Proof. The following notation is introduced. ΔO refers to a difference in the overtime of the solution induced by move m or *Swp*. In particular $\Delta O(\xi^m) = O(\xi^m) - O(\xi)$ and $\Delta O(\xi^{m+Swp}) = O(\xi^{m+Swp}) - O(\xi^m)$.

We can notice that swapping trips belonging to the same vehicle v cannot lead to any improvement: T_v is not reduced, then O_v is not reduced neither (that proves 1). Let consider two trips belonging to two different vehicles v_1 and v_2. For ease of notation, we note r_i the trip that belongs to v_i ($r_i \in v_i$) and τ_i, instead of τ_{r_i}, the travel time of trip r_i. Let T_1 (resp., O_1) and T_2 (resp., O_2) be the respective travel times (resp., overtimes) of the two vehicles. If point 2 does not hold, we will prove that swapping r_1 with r_2 cannot improve the solution.

Let suppose both vehicles are feasible or both are infeasible, i.e., $T_i(\xi^m) \leq T_H$ or $T_i(\xi^m) > T_H$, $i = 1, 2$. We consider the two cases separately.

a. $T_1(\xi^m) \leq T_H$ and $T_2(\xi^m) \leq T_H$. $O_1(\xi^m) = O_2(\xi^m) = 0$. No improvement can be carried out with *Swp*.

b. $T_1(\xi^m) > T_H$ and $T_2(\xi^m) > T_H$. We consider without loss of generality $\tau_2(\xi^m) \leq \tau_1(\xi^m)$. With *Swp*, overtime of vehicle v_1 decreases; $\Delta O_1(\xi^{m+Swp}) = \max\{\tau_2(\xi^m) - \tau_1(\xi^m), T_H - T_1(\xi^m)\}$. Overtime of vehicle v_2 increases; $\Delta O_2(\xi^{m+Swp}) = \tau_1(\xi^m) - \tau_2(\xi^m)$. Then $\Delta O(\xi^{m+Swp}) = \Delta O_1(\xi^{m+Swp}) + \Delta O_2(\xi^{m+Swp}) \geq \tau_2(\xi^m) - \tau_1(\xi^m) + \tau_1(\xi^m) - \tau_2(\xi^m)$, that is, $\Delta O(\xi^{m+Swp}) \geq 0$.

This proves point 2. Point 3 directly follows from the rule $\mathcal{R}_1$. Let us suppose v_1 and v_2 are not involved in M. Then, $\tau_k(\xi) = \tau_k(\xi^m)$ for all $r_k \in v_i$, $i = 1, 2$. If an improvement is obtained by *Swp*, the same improvement could have been obtained applying *Swp* before m (that does not modify trips involved in *Swp*). This means that when the initial assignment of trips to vehicles is optimal, no improvement can be carried out.

□

Let us now move the discussion to the choice of the move m to be tested along with a *Swp*. We introduce the following proposition.

Proposition 2. *Under Proposition 1 we can restrict the choice of moves involved along with a swap to those such that*

$$\tau_r(\xi^m) < \tau_r(\xi) \quad \text{for at least one route } r \text{ involved in } m. \qquad (\mathcal{C}_1)$$

Proof. We suppose the assignment of trips to vehicles is optimal before m is applied. We will show that when $\mathcal{C}_1$ does not hold, $m + Swp$ cannot improve the solution cost. Applying $\mathcal{R}_1$, such moves can be discarded. We indicate respectively with R and R^m the set of trips that form ξ and ξ^m. Without loss of generality we can suppose $|R| = |R_m|$ (if m creates a new route, an empty route could be added in R). Let us indicate with r a trip in R and with r_m the corresponding trip in R_m after m have been applied. The following considerations are valid.

① The cost of the solution ξ^{m+Swp} is greater than or equal to the cost of the solution obtained by optimally assigning trips in R^m to vehicles. We indicate such solution with $\xi^*_{R^m}$ and its cost with $c^*_{R^m}$;

② Let $\tilde{\xi}$ be the solution constructed by assigning trips in R as follows: r is assigned to vehicle v if and only if the corresponding r_m is assigned to vehicle v in $\xi^*_{R^m}$. We note $\tilde{c}$ the cost of such solution. Since $\tau_{r_m} \geq \tau_r$ for all $r \in R$ ($\mathcal{C}_1$ does not hold), $c^*_{R^m} \geq \tilde{c}$ is verified.

③ We note c^*_R the cost of the solution obtained by optimally assigning trips in R to vehicles. Then, it holds $\tilde{c} \geq c^*_R$.

④ We have assumed the initial assignment of trips to vehicles to be optimal. Then, $c^*_R = c(\xi)$.

Concluding, the following holds

$$c(\xi^{m+Swp}) \overset{①}{\geq} c^*_{R^m} \overset{②}{\geq} \tilde{c} \overset{③}{\geq} c^*_R \overset{④}{=} c(\xi),$$

namely, $m + Swp$ cannot improve the solution ξ. □

An algorithm sketch of the procedure is given in Algorithm 3. $Detect\_Trips\_To\_Swap(v_1, v_2)$ is a function that tests swaps between trips in vehicles v_1 and v_2. If it finds a pair of trips r_1, r_2 that improves the solution if swapped, it returns them and sets *tripDetected* to `TRUE`. Otherwise *tripDetected* is set to `FALSE`. Function $perform(m)$ performs move M while $swap(v_1, v_2, r_1, r_2)$ swaps trips r_1, r_2.

4.5 Computational results

This section reports the computational results obtained with the proposed method. The algorithm is coded in C++, compiled with Visual Studio 2008 and run on a Intel Xeon 2.80 Ghz processor. It is tested on classical instances in the MTVRP literature. These instances were introduced by Taillard et al. [198] and are constructed from the instances 1–5 and 11–12 proposed in Christofides et al. [41] (that will be denoted CMT1–CMT5 and CMT11–CMT12 in the following) and instances 11–12 proposed in Fisher [77] (F11-F12) for the VRP. For each VRP instance, instances for MTVRP are constructed with different values of m and two values of T_H, given by $T^1_H = \left[\frac{1.05z^*}{M}\right]$ and $T^2_H = \left[\frac{1.1z^*}{M}\right]$ where z^* is the solution cost of the original CVRP instances found by Rochat [178] and $[x]$ represents the closest integer to x (see Table 4.2). There are, in total, 104 different instances. For 42 of them, the optimal value

Algorithm 3 Combined LS

```
evaluate move m
if m improves the solution then
   accept m
else
   if C_1 then
      for all v_1 involved in m do
         for all v_2 ≠ v_1 do
            if (T_1(ξ^m) < T_H ∧ T_2(ξ^m) > T_H) ∨ (T_1(ξ^m) > T_H ∧ T_2(ξ^m) < T_H) then
               (r_1, r_2, tripDetected) = Detect_Trips_To_Swap(v_1, v_2)
               if tripDetected then
                  perform(m)
                  swap(v_1, v_2, r_1, r_2)
               end if
            end if
         end for
      end for
   end if
end if
```

Instance	N	Q	z^*
CMT1	50	160	524.61
CMT2	75	140	835.26
CMT3	100	200	826.14
CMT4	150	200	1028.42
CMT5	199	200	1291.44
CMT11	120	200	1042.11
CMT12	100	200	819.56
F11	71	30000	241.97
F12	134	2210	1162.92

Table 4.2: Instances' details

is known and is provided by Mingozzi et al. [140]. We classify them in a first group denoted G1. For the remaining 62 instances, 56 have a known feasible solution (they will form a second group G2). The six remaining instances form the third group G3. These instances are not solved yet. These groups of the instances set will be used during the presentation of the computational results. When it is necessary to indicate a specific instance, the notation $\mathcal{N}\_T_H^i\_m$, will be used, where $\mathcal{N}$ stands for the original VRP instance name and $i = 1, 2$ for the horizon length.

4.5.1 Parameter settings

Overtime and overload penalization parameters

The overtime penalization parameter θ is set to 2 and it is kept fixed during all the search. That is done because the value $\theta = 2$ is used in literature to penalize overtime when a feasible solution is not found.

The overload penalization parameter λ is set to $\bar{D}/\bar{Q}$, where $\bar{D}$ represents the average distance among customers and $\bar{Q}$ the average demand of customers. The value of λ is kept fixed during the search. Different dynamic adaptation schemes were tested, but no visible improvements were obtained.

Parameter tuning

The procedure requires the setting of a number n_p of parameters among values that have to be chosen in sensible ranges. To set the parameters involved in our algorithm, a *tuning method* is used. Roughly speaking, a tuning method is a procedure whose search space is $P^1 \times \cdots \times P^{n_p}$, where P^i is the domain of parameter i and looks for the solution with the best *utility*, that is a measure of the algorithm's efficiency on a given parameter vector (Smit and Eiben [194]). In particular, the Evolutionary Strategy with Covariance Matrix Adaptation proposed by Hansen and Ostermeier [101] is used. The tuning algorithm is run on a limited set of instances formed by CMT1_T_H^2_4, CMT2_T_H^1_6, CMT3_T_H^1_6, CMT4_T_H^1_8, CMT5_T_H^1_9, CMT11_T_H^1_4 to determine the values of parameters listed in Table 4.3. Instances with a large number of vehicles were selected since they are more difficult to solve. Other parameters are fixed a priori: the probability of educate a new

	Parameter	Range	Final value
Π	Dimension of population	$[1, 100]$	9
μ	Children generated at each generation	$[1, 100]$	32
n^e	Proportion of elite individuals $n_e = n^e \times \Pi$ (Eq. 4.4)	$[0.1, 1]$	0.2
n^c	Proportion of close individuals $n_c = n^c \times \Pi$ (Eq. 4.5)	$[0.1, 1]$	0.35
h	Granularity threshold in LS $n_{closest} = h \times n$	$[0.2, 1]$	0.45

Table 4.3: Parameter Tuning

chromosome is $p_{LS} = 1$ and the probability to repair an infeasible chromosome is $p_{rep} = 0.5$ as in Vidal et al. [214]. The adopted survivor strategy (Section 4.3.6) allows for the use of LS to educate each chromosome without premature convergence of the population. That is in particular due to the fact that survivor chromosomes are selected based on their contribution to the diversification of the population as well as their fitness value.

4.5.2 Discussion

A fair and comprehensive comparison with previous works is quite difficult to carry out since (as already mentioned) complete and precise values are reported only by Salhi and Petch [186].

Olivera and Veira [156], report detailed results as well, but with some imprecision. Indeed, these authors provide gaps to values z^* (see Table 4.2), which cannot be precisely converted into solution costs due to truncation.

Notation reported in Table 4.4 will be used in the following. In all tables, the first three columns indicate respectively the name, the number of vehicles and the time horizon of the instances.

Symbol	Meaning
TLG	results from Taillard, Laporte and Gendreau [198]
BM	results from Brandão and Mercer [25]
SP	results from Salhi and Petch [186]
OV	results from Olivera and Viera [156]
AAB	results from Alonso, Alvarez and Beasley [3]
MRT	results from Mingozzi, Roberti and Toth [140]
MA-F	results from our MA stopping at the first feasible found solution
MA	results from MA without the usage of CLS
MA+CLS	results from our MA with the usage of CLS
Best	Best value over five runs
Av	Average value over five runs
Worst	Worst value over five runs
StDv	Standard deviation over five runs
#fs	Number of runs ended with a feasible solution
#opt	Number of runs ended with an optimal solution
✓	a feasible solution is found
×	a feasible solution is not found
✘	the instance is not considered

Table 4.4: Notation for computational results

All procedures stop once 2000 individuals have been generated. Preliminary computational experiments shown that it is a good compromise between solution quality and computational efficiency.

The results are reported as follows. In Section 4.5.2, the ability of the algorithm to find feasible solutions is tested. The algorithm terminates whenever a feasible solution is found, or after 2000 chromosome constructions. In Section 4.5.2 two variants of the algorithm, with or without CLS, are evaluated and complete and detailed results are reported. Both versions stop after a fixed number of iterations. Separate comparison with the results obtained by Olivera and Viera [156] is discussed in Section 4.5.2. Finally, computational times comparison is discussed in Section 4.5.2.

Feasibility check algorithm

The procedure is first run five times over all instances to measure its capability to obtain feasible solutions: it stops as soon as a feasible solution is found or after 2000 iterations. It is indicated as MA-F. The efficiency of the algorithm is measured on the time needed to find a feasible solution without considering its value, following the implicit idea of the paper by Taillard et al. [198]. Results are reported on Tables 4.5 and 4.6.

Instance			Algorithm						
Name	M	T_H	TLG	BM	SP	OV	AAB	MA-F	#fs
CMT1	1	551	✓	✓	✓	✓	✓	✓	5
	2	275	✓	✓	×	✓	✓	✓	5
	1	577	✓	✓	✓	✓	✓	✓	5
	2	289	✓	✓	✓	✓	✓	✓	5
	4	144	✓	✓	✓	✓	✓	✓	5
CMT2	1	877	✓	✓	✓	✓	✓	✓	5
	2	439	✓	✓	✓	✓	✓	✓	5
	3	292	✓	✓	×	✓	✓	✓	5
	4	219	✓	✓	✓	✓	✓	✓	5
	5	175	✓	✓	×	✓	✓	✓	5
	1	919	✓	✓	✓	✓	✓	✓	5
	2	459	✓	✓	✓	✓	✓	✓	5
	3	306	✓	✓	✓	✓	✓	✓	5
	4	230	✓	✓	✓	✓	✓	✓	5
	5	184	✓	✓	✓	✓	✓	✓	5
	6	153	✓	✓	×	✓	✓	✓	5
CMT3	1	867	✓	✓	✓	✓	✓	✓	5
	2	434	✓	✓	✓	✓	✓	✓	5
	3	289	✓	✓	×	✓	✓	✓	5
	1	909	✓	✓	✓	✓	✓	✓	5
	2	454	✓	✓	✓	✓	✓	✓	5
	3	303	✓	✓	✓	✓	✓	✓	5
	4	227	✓	✓	✓	✓	✓	✓	5
CMT11	1	1094	✓	✓	✓	✓	✓	✓	5
	2	547	✓	✓	×	✓	✓	✓	5
	3	365	✓	✓	×	✓	✓	✓	5
	5	219	✓	✓	×	✓	✓	✓	5
	1	1146	✓	✓	✓	✓	✓	✓	5
	2	573	✓	✓	✓	✓	✓	✓	5
	3	382	✓	✓	✓	✓	✓	✓	5
	4	287	✓	✓	×	✓	✓	✓	5
	5	229	✓	✓	✓	✓	✓	✓	5
CMT12	1	861	✓	✓	✓	✓	✓	✓	5
	2	430	✓	✓	✓	✓	✓	✓	5
	3	287	✓	✓	✓	✓	✓	✓	5
	4	215	✓	✓	✓	✓	✓	✓	5
	1	902	✓	✓	✓	✓	✓	✓	5
	2	451	✓	✓	✓	✓	✓	✓	5
	3	301	✓	✓	✓	✓	✓	✓	5
	4	225	✓	✓	✓	✓	✓	✓	5
	5	180	✓	✓	✓	✓	✓	✓	5
	6	150	✓	✓	✓	✓	✓	✓	5
# instances solved			42	42	33	42	42	42	

Table 4.5: Feasibility check on the 42 instances in G1

The algorithm is able to find a feasible solution in at least one run on all instances from groups G1 and G2. Better, feasible solutions are always found on G1 and for 50 instances

Instance			Algorithm						
Name	M	T_H	TLG	BM	SP	OV	AAB	MA-F	#fs
CMT1	3	192	✓	✓	✓	✓	✓	✓	5
CMT2	6	146	×	×	×	✓	×	✓	1
	7	131	✓	✓	×	✓	✓	✓	5
CMT3	4	217	✓	✓	×	✓	✓	✓	5
	5	173	×	✓	×	✓	✓	✓	5
	6	145	✓	✓	×	✓	✓	✓	5
	5	182	✓	✓	✓	✓	✓	✓	5
	6	151	✓	✓	✓	✓	✓	✓	5
CMT4	1	1080	✓	✓	✓	✓	✓	✓	5
	2	540	✓	✓	✓	✓	✓	✓	5
	3	360	✓	✓	×	✓	✓	✓	5
	4	270	✓	✓	×	✓	✓	✓	5
	5	216	✓	✓	×	✓	✓	✓	5
	6	180	✓	✓	×	✓	✓	✓	5
	8	135	×	×	×	✓	×	✓	2
	1	1131	✓	✓	✓	✓	✓	✓	5
	2	566	✓	✓	✓	✓	✓	✓	5
	3	377	✓	✓	✓	✓	✓	✓	5
	4	283	✓	✓	✓	✓	✓	✓	5
	5	226	✓	✓	✓	✓	✓	✓	5
	6	189	✓	✓	✓	✓	✓	✓	5
	7	162	✓	✓	×	✓	✓	✓	5
	8	141	✓	✓	×	✓	✓	✓	5
CMT5	1	1356	✓	✓	✓	✓	✓	✓	5
	2	678	✓	✓	✓	✓	✓	✓	5
	3	452	✓	✓	×	✓	✓	✓	5
	4	339	✓	✓	×	✓	✓	✓	5
	5	271	✓	✓	×	✓	✓	✓	5
	6	226	✓	✓	×	✓	✓	✓	5
	7	194	✓	✓	×	✓	✓	✓	5
	8	170	✓	✓	×	✓	✓	✓	5
	9	151	✓	×	×	✓	×	✓	4
	10	136	×	×	×	✓	×	✓	2
	1	1421	✓	✓	✓	✓	✓	✓	5
	2	710	✓	✓	✓	✓	✓	✓	5
	3	474	✓	✓	✓	✓	✓	✓	5
	4	355	✓	✓	✓	✓	✓	✓	5
	5	284	✓	✓	✓	✓	✓	✓	5
	6	237	✓	✓	✓	✓	✓	✓	5
	7	203	✓	✓	✓	✓	✓	✓	5
	8	178	✓	✓	×	✓	✓	✓	5
	9	158	✓	✓	×	✓	✓	✓	5
	10	142	✓	✓	×	✓	✓	✓	5
CMT11	4	274	×	×	×	✓	×	✓	1
CMT12	5	172	✓	✓	×	✓	✓	✓	1
F11	1	254	✓	✓	×	✓	✘	✓	5
	2	127	×	×	×	✓	✘	✓	5
	1	266	✓	✓	✓	✓	✘	✓	5
	2	133	✓	✓	✓	✓	✘	✓	5
	3	89	✓	✓	✓	✓	✘	✓	5
F12	1	1221	✓	✓	✓	✓	✘	✓	5
	2	611	✓	✓	✓	✓	✘	✓	5
	3	407	✓	✓	✓	✓	✘	✓	5
	1	1279	✓	✓	✓	✓	✘	✓	5
	2	640	✓	✓	✓	✓	✘	✓	5
	3	426	✓	✓	✓	✓	✘	✓	5
# instances solved			50	50	29	56	40	56	

Table 4.6: Feasibility check on the 56 instances in G2

out of 56 on G2. In general, on all the 490 runs, feasible solutions are obtained 471 times, denoting high efficiency of the algorithm. Comparatively, only Olivera and Viera [156] exhibit similar results. No feasible solutions are found on G3 instances.

Detailed results

The algorithm is run again five times over all the instances. Complete and detailed results are reported in Tables 4.7–4.10. Results from the MA without CLS are reported in columns indicated with MA while those from the MA with CLS are given in columns indicated with MA+CLS.

Table 4.7 reports results obtained on the 42 instances of G1. Optimal values are indicated in bold. MA and MA+CLS find optimal solutions on all the five runs in 23 cases, but the former finds the optimal value at least once in 32 cases while the latter in 37 cases. In general, MA+CLS is more efficient in finding optimal solutions: they are obtained 137 times over 210 runs while MA finds optimal solutions 129 times. Both procedures always find feasible solutions. Note that Salhi and Petch [186] do not find any optimal solution and it is outperformed by both methods on all instances.

Results on instances of G2 are detailed in Table 4.8. Here, bold numbers are used to indicate best known values. MA finds a feasible solution at least once over all instances and the procedure finds a feasible solution on all the five runs in 50 cases (out of 56) for a total of 261 feasible solutions out of 280 runs. Introducing the CLS improves the results. Feasible solutions are always found in 52 cases and at least 2 feasible solutions are found over the five runs for a total of 271 feasible solutions. Again, solutions found by the procedures are always better than those reported in Salhi and Petch [186].

Tables 4.9 and 4.10 report results on instances of G3. First of all, it can be noticed from Table 4.9 that MA+CLS finds a new feasible solution for instance CMT4_T_H^1_7 (details can be found in A). On the other five instances (Table 4.10), direct comparison with other methods on values of infeasible solutions found is possible. MA+CLS finds two new best known values for instances CMT2_T_H^1_7 and F11_T_H^1_3. For the latter, the new best known value is as well reached by MA. On average, both methods outperform the others.

Averagely, MA+CLS performs better than MA as can be seen in the last columns of Tables 4.7, 4.8 and 4.10. This, together with the new feasible solution found for instance CMT4_T_H^1_7 by MA+CLS, validates the usefulness and efficiency of the CLS.

Detailed comparison with Olivera and Viera [156]

A full comparison following the scheme proposed by Olivera and Viera [156] is proposed in Tables 4.11 and 4.12. Given a solution ξ, the value $GAP(\xi)$ is calculated as

$$GAP(\xi) = 100 \cdot \left(\frac{c(\xi)}{z^*} - 1 \right), \tag{4.6}$$

and results are reported accordingly. The number of runs ended with a feasible solution is reported for instances in G1 as in Olivera and Viera [156]. Focusing on the gap values, as it can be noticed, results obtained by MA+CLS outperform those by Olivera and Viera [156].

			Algorithm							
Instance			MRT	SP	MA			MA+CLS		
Name	M	T_H	Optimal	Best	Best	Average	#opt	Best	Av	#opt
CMT1	1	551	524.61	546.28	**524.61**	524.61	5	**524.61**	524.61	5
	2	275	533.00	×	**533.00**	533.67	4	**533.00**	533.00	5
	1	577	524.61	547.14	**524.61**	524.61	5	**524.61**	524.61	5
	2	289	529.85	549.42	**529.85**	529.85	5	**529.85**	530.67	3
	4	144	546.29	566.86	**546.29**	546.29	5	**546.29**	546.29	5
CMT2	1	877	835.26	869.06	**835.26**	838.40	2	**835.26**	838.40	2
	2	439	835.26	865.48	835.77	840.04	0	**835.26**	838.59	1
	3	292	835.26	×	**835.26**	836.32	1	**835.26**	838.58	2
	4	219	835.26	856.77	835.77	839.41	0	835.77	839.77	0
	5	175	835.8	×	836.18	841.97	0	836.18	836.52	0
	1	919	835.26	869.73	**835.26**	835.48	2	**835.26**	835.48	2
	2	459	835.26	881.50	**835.26**	839.20	1	**835.26**	836.46	1
	3	306	835.26	869.11	835.77	840.07	0	**835.26**	837.40	2
	4	230	835.26	880.90	838.17	840.41	0	**835.26**	837.73	2
	5	184	835.26	883.29	835.77	837.71	0	835.77	837.99	0
	6	153	839.22	×	843.09	848.06	0	**839.22**	846.02	1
CMT3	1	867	826.14	845.33	**826.14**	827.96	1	**826.14**	827.96	1
	2	434	826.14	850.65	**826.14**	827.96	0	**826.14**	827.75	2
	3	289	826.14	×	828.08	829.63	0	**826.14**	828.53	1
	1	909	826.14	845.33	829.45	829.53	0	829.45	829.53	0
	2	454	826.14	872.10	**826.14**	828.80	1	**826.14**	827.96	1
	3	303	826.14	869.48	**826.14**	828.94	1	827.39	829.09	0
	4	227	826.14	878.00	**826.14**	828.01	1	**826.14**	827.55	1
CMT11	1	1094	1042.11	1088.26	**1042.11**	1042.11	5	**1042.11**	1042.11	5
	2	547	1042.11	×	**1042.11**	1042.11	5	**1042.11**	1042.11	5
	3	365	1042.11	×	**1042.11**	1042.11	5	**1042.11**	1042.11	5
	5	219	1042.11	×	**1042.11**	1042.11	5	**1042.11**	1042.11	5
	1	1146	1042.11	1088.26	**1042.11**	1042.11	5	**1042.11**	1042.11	5
	2	573	1042.11	1110.10	**1042.11**	1042.11	5	**1042.11**	1042.11	5
	3	382	1042.11	1088.56	**1042.11**	1042.11	5	**1042.11**	1042.11	5
	4	287	1042.11	×	**1042.11**	1042.11	5	**1042.11**	1042.11	5
	5	229	1042.11	1092.95	**1042.11**	1042.11	5	**1042.11**	1042.11	5
CMT12	1	861	819.56	819.97	**819.56**	819.56	5	**819.56**	819.56	5
	2	430	819.56	821.33	**819.56**	819.56	5	**819.56**	819.56	5
	3	287	819.56	826.98	**819.56**	819.56	5	**819.56**	819.56	5
	4	215	819.56	824.57	**819.56**	819.56	5	**819.56**	819.56	5
	1	902	819.56	819.97	**819.56**	819.56	5	**819.56**	819.56	5
	2	451	819.56	829.54	**819.56**	819.56	5	**819.56**	819.56	5
	3	301	819.56	851.16	**819.56**	819.56	5	**819.56**	819.56	5
	4	225	819.56	821.53	**819.56**	819.56	5	**819.56**	819.56	5
	5	180	824.78	833.85	**824.78**	824.78	5	**824.78**	824.78	5
	6	150	823.14	855.36	**823.14**	823.14	5	**823.14**	823.15	5
total optimal solutions found							129			137
average					838.85	840.01		838.64	839.62	
average GAP from optimal					0.05	0.19		0.03	0.15	

Table 4.7: Feasible solutions on the 42 instances in G1

				Algorithm						
Instance				SP	MA			MA+CLS		
Name	M	T_H	Best known	Best	Best	Av	#fs	Best	Av	#fs
CMT1	3	192	552.68	560.26	**552.68**	552.68	5	**552.68**	552.68	5
CMT2	6	146	858.58	×	**858.58**	858.58	1	859.16	859.42	3
	7	131	844.70	×	853.88	861.64	5	**844.70**	854.70	5
CMT3	4	217	829.45	×	**829.45**	829.65	5	**829.45**	829.45	5
	5	173	832.89	×	**832.89**	835.98	5	**832.89**	843.72	5
	6	145	836.22	×	**836.22**	837.06	5	**836.22**	836.22	5
	5	182	832.34	901.30	833.02	833.83	5	**832.34**	832.88	5
	6	151	834.35	861.76	**834.35**	834.83	5	**834.35**	834.35	5
CMT4	1	1080	1031.00	1064.06	**1031.00**	1034.22	5	**1031.00**	1034.22	5
	2	540	1031.07	1065.86	1032.65	1038.33	5	**1031.07**	1037.89	5
	3	360	1028.42	×	1029.56	1036.73	5	**1028.42**	1032.79	5
	4	270	1031.10	×	1036.25	1040.75	5	**1031.10**	1037.09	5
	5	216	1031.07	×	1032.69	1040.42	5	**1031.07**	1037.41	5
	6	180	1034.61	×	1043.42	1046.71	5	**1034.61**	1041.82	5
	8	135	1056.54	×	1056.93	1059.58	2	**1056.54**	1059.68	3
	1	1131	1031.07	1088.93	**1031.07**	1038.77	5	**1031.07**	1038.77	5
	2	566	1030.45	1070.50	**1030.45**	1037.29	5	1034.08	1040.39	5
	3	377	1031.59	1077.24	1031.63	1040.75	5	**1031.59**	1032.92	5
	4	283	1031.07	1119.05	**1031.07**	1034.69	5	1031.96	1036.33	5
	5	226	1030.86	1085.38	1033.05	1039.52	5	**1030.86**	1035.52	5
	6	189	1030.45	1112.03	1032.16	1038.62	5	**1030.45**	1037.10	5
	7	162	1036.08	×	1043.92	1047.87	5	**1036.08**	1043.60	5
	8	141	1044.32	×	1044.71	1050.28	5	**1044.32**	1048.08	5
CMT5	1	1356	1302.43	1347.34	**1302.43**	1308.27	5	**1302.43**	1308.27	5
	2	678	1302.15	1346.63	**1302.15**	1309.04	5	1306.26	1309.66	5
	3	452	1301.29	×	1301.41	1309.33	5	**1301.29**	1307.85	5
	4	339	1304.78	×	1308.93	1312.76	5	**1304.78**	1308.07	5
	5	271	1300.02	×	1307.78	1314.66	5	**1300.02**	1307.10	5
	6	226	1303.37	×	**1303.37**	1314.29	5	1308.40	1311.16	5
	7	194	1309.40	×	1315.41	1319.86	5	**1309.40**	1313.06	5
	8	170	1303.91	×	1310.48	1316.53	5	**1303.91**	1308.98	5
	9	151	1307.93	×	1329.86	1331.61	4	**1307.93**	1317.03	5
	10	136	1323.01	×	1326.54	1331.59	2	**1323.01**	1329.00	5
	1	1421	1299.86	1340.44	**1299.86**	1310.43	5	**1299.86**	1310.43	5
	2	710	1305.35	1399.65	**1305.35**	1310.98	5	1307.70	1314.05	5
	3	474	1301.03	1409.37	**1301.03**	1312.15	5	1308.76	1310.93	5
	4	355	1303.65	1397.60	**1303.65**	1311.19	5	1310.97	1312.40	5
	5	284	1300.62	1411.19	1308.04	1311.87	5	**1300.62**	1308.75	5
	6	237	1306.17	1377.07	**1306.17**	1308.49	5	1306.25	1311.40	5
	7	203	1301.54	1394.73	1311.35	1314.18	5	**1301.54**	1313.66	5
	8	178	1308.78	×	1311.93	1313.86	5	**1308.78**	1310.61	5
	9	158	1307.25	×	1312.28	1318.26	5	**1307.25**	1311.32	5
	10	142	1308.81	×	1312.04	1321.27	5	**1308.81**	1316.80	5
CMT11	4	274	1078.64	×	1080.12	1080.12	1	**1078.64**	1080.38	3
CMT12	5	172	845.56	×	849.89	849.89	1	**845.564**	847.727	2
F11	1	254	241.97	×	**241.97**	241.97	5	**241.97**	241.97	5
	2	127	250.85	×	**250.85**	250.85	5	**250.85**	250.85	5
	1	266	241.97	254.07	**241.97**	241.97	5	**241.97**	241.97	5
	2	133	241.97	254.07	**241.97**	241.97	5	**241.97**	241.97	5
	3	89	254.07	256.53	**254.07**	254.07	5	**254.07**	254.07	5
F12	1	1221	1162.96	1190.21	**1162.96**	1162.96	5	**1162.96**	1162.96	5
	2	611	1162.96	1194.24	**1162.96**	1162.96	5	**1162.96**	1162.96	5
	3	407	1162.96	1199.86	**1162.96**	1163.05	5	**1162.96**	1162.96	5
	1	1279	1162.96	1183.00	**1162.96**	1162.96	5	**1162.96**	1162.96	5
	2	640	1162.96	1199.64	**1162.96**	1162.96	5	**1162.96**	1162.96	5
	3	426	1162.96	1215.43	**1162.96**	1162.96	5	**1162.96**	1162.96	5
total feasible solutions found							261			271
average					1040.90	1044.70		1039.23	1043.11	

Table 4.8: Feasible solutions on the 56 instances in G2

				Algorithm										
				TLG	BM	SP	OV	AAB	MA			MA+CLS		
Name	M	T_H	Best known	Best	Best	Best	Best	Best	Best	Av	#fs	Best	Av	#fs
CMT1	3	184	-	×	×	×	×	×	×	-	-	×	-	-
CMT1	4	138	-	×	×	×	×	×	×	-	-	×	-	-
CMT2	7	125	-	×	×	×	×	×	×	-	-	×	-	-
CMT4	7	154	1068.59	×	×	×	×	×	×	-	-	**1068.59**	1068.59	1
CMT12	6	143	-	×	×	×	×	×	×	-	-	×	-	-
F11	3	85	-	×	×	×	×	✘	×	-	-	×	-	-

Table 4.9: Feasible solutions on the 6 instances in G3

				Algorithm								
				TLG	BM	SP	OV	AAB	MA		MA+CLS	
Name	M	T_H	Best known	best	best	best	best	best	Best	Av	Best	Av
CMT1	3	184	569.54	579.48	575.73	586.32	573.4	**569.54**	**569.54**	569.54	**569.54**	569.54
CMT1	4	138	564.07	565.27	**564.07**	632.54	**564.07**	564.1	**564.07**	564.07	**564.07**	564.07
CMT2	7	125	866.58	878.29	896.57	1056.34	877.12	878.05	876.77	880.06	**866.58**	873.14
CMT12	6	143	845.48	**845.48**	847.85	898.88	860.61	866.54	**845.48**	845.48	**845.48**	845.48
F11	3	85	256.93	257.31	257.47	266.85	260.55	✘	**256.93**	256.93	**256.93**	256.93
			average	625.17	628.34	688.19	627.15	-	622.56	623.21	620.52	621.83

Table 4.10: Non-feasible solutions on the 5 unsolved instances in G3

On the other side, the algorithm proposed by Olivera and Viera [156] performs better than MA-F. A probable reason is that MA-F terminates the procedure as soon as a feasible solution is found, while Olivera and Viera [156] check for feasibility each 100 iterations of their procedure. Regarding computing times (see Section 4.5.2), note however that Olivera and Viera [156] and MA-F are much quicker than MA+CLS.

Computational times

A fair computational time comparison could not be performed as the machine relative speeds were not found for all the computers used by previous papers. Machines used in previous works are listed in Table 4.13. Original computational times, as well as those of our method, are reported in Table 4.14 (times are expressed in seconds). Furthermore, algorithm differences and inharmonious computational time reporting complicate comparison. In particular, Taillard et al. [198] perform their algorithm five times on each instance. If no feasible solution is found, it is run another time. Average time on all runs is reported. Brandão and Mercer [25] stop their procedure once a feasible solution is found and they report computational times over the runs where a feasible solution is found. Salhi and Petch [186] and Alonso et al. [3] stop their algorithm when a maximum number of iteration is reached, but while the former reports average computational time over five runs the latter runs the algorithm just once. Olivera and Viera [156] check for feasibility each 100 iterations and terminate the computation in case of success. They report computational times only for the best run. We report the average computational time, on five runs, for each class of instances. Finally, our goal is to find high quality solution and not to just satisfy feasibility as it was done in previous works. Keeping that in mind, it can be noticed from Table 4.14 that MA-F is able to find feasible solutions very quickly (almost instantaneously for instances of families CMT3 and F12). We can also notice that the use of the CLS increases the time spent by the procedure. The time increase is, however, rewarded by more efficiency in finding optimal and feasible solutions as already outlined in Section 4.5.2.

4.6 Conclusion and future work

In this paper we proposed a genetic algorithm for the Multi-Trip Vehicle Routing Problem. It is the first evolutionary procedure that efficiently faces the benchmark of instances proposed in the literature.

We use an adaptation of the *Split* procedure proposed by Prins [166] to evaluate the chromosomes.

We introduce a new LS operator that performs pejorative moves along with re-assignment of trips to vehicles and is called Combined LS (CLS). The efficiency of the CLS is validated by the quality of the results obtained. This opens a new promising research direction related to the management of moves combined with re-packing procedures.

We report detailed results over all instances (and not only for unsolved instances) and we give precise values of the found solutions (differently than what is done in Olivera and Viera [156]).

Instance			MA-F					OV					MA+CLS				
Name	M	T_H	Best	Av	Worst	StDv	#fs	Best	Av	Worst	StDv	#fs	Best	Av	Worst	StDv	#fs
CMT1	1	551	0.0	2.7	4.8	2.0	5	0.0	0.0	0.0	0.0	5	0.0	0.0	0.0	0.0	5
	2	275	1.6	3.2	4.2	1.2	5	1.6	1.7	2.2	0.3	5	1.6	1.6	1.6	0.0	5
	1	577	2.3	4.8	7.3	2.3	5	0.0	0.0	0.0	0.0	5	0.0	0.0	0.0	0.0	5
	2	289	4.6	6.9	9.3	1.7	5	1.0	1.1	1.6	0.3	5	1.0	1.2	1.6	0.3	5
	4	144	4.6	5.7	7.6	1.3	5	4.1	4.2	4.3	0.1	5	4.1	4.1	4.1	0.0	5
CMT2	1	877	2.8	3.4	3.8	0.5	5	0.0	0.2	0.3	0.1	5	0.0	0.4	0.7	0.4	5
	2	439	2.6	3.6	4.4	0.7	5	0.2	0.5	1.0	0.4	5	0.0	0.4	0.7	0.3	5
	3	292	3.3	3.7	4.4	0.5	5	0.2	0.5	1.3	0.5	5	0.0	0.4	0.8	0.4	5
	4	219	1.8	2.4	3.3	0.7	5	0.1	0.7	1.7	0.7	5	0.1	0.5	0.8	0.3	5
	5	175	2.6	2.9	3.6	0.4	5	1.2	1.6	2.1	0.4	5	0.1	0.2	0.2	0.0	5
	1	919	4.3	4.9	5.5	0.5	5	0.1	0.7	1.3	0.5	5	0.0	0.0	0.1	0.0	5
	2	459	3.7	4.2	4.6	0.4	5	0.1	0.5	1.0	0.4	5	0.0	0.1	0.5	0.2	5
	3	306	3.3	5.3	8.0	1.9	5	0.1	0.4	0.8	0.3	5	0.0	0.3	0.7	0.4	5
	4	230	4.2	6.0	9.6	2.2	5	0.1	0.2	0.6	0.2	5	0.0	0.3	0.5	0.3	5
	5	184	3.7	5.4	8.1	2.0	5	0.1	0.9	1.3	0.5	5	0.1	0.3	0.7	0.3	5
	6	153	3.3	4.9	6.0	1.0	5	2.5	3.4	4.8	0.9	5	0.5	1.3	2.4	0.7	5
CMT3	1	867	1.6	2.8	4.0	0.9	5	0.0	0.4	0.6	0.2	5	0.0	0.2	0.4	0.2	5
	2	434	1.2	3.0	4.4	1.5	5	0.4	0.5	1.0	0.2	5	0.0	0.2	0.4	0.2	5
	3	289	2.3	3.4	4.6	0.8	5	0.5	0.7	0.9	0.2	5	0.0	0.3	0.4	0.2	5
	1	909	2.1	3.3	4.7	1.1	5	0.4	0.5	0.7	0.2	5	0.4	0.4	0.4	0.0	5
	2	454	2.2	4.8	6.7	1.6	5	0.3	0.4	0.7	0.2	5	0.0	0.2	0.4	0.2	5
	3	303	2.5	5.3	8.9	2.6	5	0.2	0.4	0.4	0.1	5	0.2	0.4	0.4	0.1	5
	4	227	2.3	5.4	7.9	2.1	5	0.0	0.3	0.5	0.2	5	0.0	0.2	0.4	0.1	5
CMT11	1	1094	0.8	3.6	5.0	1.7	5	0.1	2.5	3.2	1.3	5	0.0	0.0	0.0	0.0	5
	2	547	0.8	2.8	4.4	1.5	5	3.0	3.1	3.2	0.1	5	0.0	0.0	0.0	0.0	5
	3	365	0.2	0.7	1.1	0.3	5	0.2	2.7	4.5	2.1	5	0.0	0.0	0.0	0.0	5
	5	219	0.4	1.5	2.3	0.8	5	0.1	0.8	1.3	0.4	5	0.0	0.0	0.0	0.0	5
	1	1146	3.8	7.1	10.0	3.0	5	0.2	2.5	3.1	1.3	5	0.0	0.0	0.0	0.0	5
	2	573	0.8	3.8	7.1	2.3	5	1.2	2.6	3.1	0.8	5	0.0	0.0	0.0	0.0	5
	3	382	1.0	4.1	5.0	1.7	5	0.1	0.8	3.3	1.4	5	0.0	0.0	0.0	0.0	5
	4	287	0.6	1.6	2.6	0.8	5	0.1	0.2	0.4	0.1	5	0.0	0.0	0.0	0.0	5
	5	229	1.3	2.5	4.4	1.3	5	0.1	0.3	0.6	0.2	5	0.0	0.0	0.0	0.0	5
CMT12	1	861	0.2	1.5	5.0	2.0	5	0.0	0.1	0.2	0.1	5	0.0	0.0	0.0	0.0	5
	2	430	0.3	1.8	4.0	1.7	5	0.0	0.1	0.2	0.1	5	0.0	0.0	0.0	0.0	5
	3	287	0.5	2.1	3.3	1.3	5	0.0	0.1	0.1	0.1	5	0.0	0.0	0.0	0.0	5
	4	215	0.3	0.6	1.1	0.3	5	0.0	0.0	0.0	0.0	5	0.0	0.0	0.0	0.0	5
	1	902	0.2	4.2	8.1	3.7	5	0.0	0.1	0.1	0.1	5	0.0	0.0	0.0	0.0	5
	2	451	0.8	4.0	6.0	2.5	5	0.0	0.1	0.1	0.1	5	0.0	0.0	0.0	0.0	5
	3	301	0.7	2.6	4.2	1.7	5	0.0	0.1	0.1	0.1	5	0.0	0.0	0.0	0.0	5
	4	225	0.9	2.0	5.2	1.8	5	0.0	0.1	0.1	0.1	5	0.0	0.0	0.0	0.0	5
	5	180	1.2	3.7	6.8	2.9	5	0.6	0.6	0.6	0.0	5	0.6	0.6	0.6	0.0	5
	6	150	0.4	3.8	6.9	2.3	5	0.4	0.5	0.7	0.1	5	0.4	0.4	0.4	0.0	5
total feasible solutions found							210					210					210
		average	1.9	3.6	5.4	1.5		0.5	0.9	1.3	0.4		0.2	0.3	0.5	0.1	

Table 4.11: Comparison with OV on the 42 instances in G1

Instance			MA-F					OV					MA+CLS				
Name	M	T_H	Best	Av	Worst	StDv	#fs	Best	Av	Worst	StDv	#fs	Best	Av	Worst	StDv	#fs
CMT1	3	192	6.7	7.5	8.8	1.0	5	5.4	5.5	6.0	0.3	5	5.4	5.4	5.4	0.0	5
CMT2	6	146	2.9	2.9	2.9	-	1	2.6	2.6	2.6	-	1	2.9	2.9	2.9	0.0	3
	7	131	3.7	5.4	7.2	1.5	5	3.6	4.2	4.7	0.4	5	1.1	2.3	2.9	0.7	5
CMT3	4	217	1.4	3.4	4.6	1.3	5	0.4	0.8	1.2	0.3	5	0.4	0.4	0.4	0.0	5
	5	173	2.2	2.9	3.4	0.6	5	2.6	3.2	4.1	0.6	5	0.8	2.1	2.5	0.7	5
	6	145	1.2	1.8	2.3	0.4	5	1.2	2.0	3.0	0.7	5	1.2	1.2	1.2	0.0	5
	5	182	2.6	4.1	5.2	1.1	5	0.8	1.0	1.9	0.5	5	0.7	0.8	0.8	0.0	5
	6	151	2.6	5.2	7.8	1.9	5	1.1	1.7	1.9	0.4	5	1.0	1.0	1.0	0.0	5
CMT4	1	1080	3.1	3.8	4.6	0.7	5	0.5	0.9	1.3	0.4	5	0.3	0.6	1.4	0.5	5
	2	540	2.6	3.7	4.4	0.7	5	0.8	0.7	3.5	1.1	5	0.3	0.9	1.4	0.6	5
	3	360	2.3	3.5	4.2	0.8	5	0.7	1.3	1.8	0.4	5	0.0	0.4	1.4	0.6	5
	4	270	2.6	3.7	4.2	0.7	5	0.8	1.6	2.5	0.9	5	0.3	0.8	1.5	0.6	5
	5	216	2.9	3.6	4.6	0.7	5	0.4	1.6	2.8	0.9	5	0.3	0.9	1.8	0.7	5
	6	180	2.7	3.6	4.2	0.7	5	2.0	3.2	4.5	0.9	5	0.6	1.3	2.1	0.6	5
	8	135	2.8	3.0	3.3	0.4	2	3.0	3.6	4.3	0.7	3	2.7	3.0	3.4	0.3	3
	1	1131	4.5	6.5	8.3	1.8	5	1.0	1.3	1.6	0.2	5	0.3	1.0	1.6	0.7	5
	2	566	3.5	5.2	5.8	1.0	5	0.9	1.5	2.1	0.5	5	0.6	1.2	1.5	0.4	5
	3	377	3.9	5.6	6.7	1.1	5	0.5	1.1	1.6	0.4	5	0.3	0.4	0.5	0.1	5
	4	283	4.1	5.4	6.5	0.8	5	1.0	1.7	2.5	0.5	5	0.3	0.8	1.2	0.4	5
	5	226	4.7	5.8	6.8	0.9	5	0.8	1.5	2.3	0.6	5	0.2	0.7	1.8	0.6	5
	6	189	5.6	6.6	7.7	0.9	5	0.4	1.4	2.0	0.7	5	0.2	0.8	1.7	0.7	5
	7	162	4.7	6.8	8.5	1.5	5	2.6	3.3	3.9	0.6	5	0.7	1.5	2.4	0.7	5
	8	141	3.1	5.9	8.6	2.2	5	3.5	4.4	5.6	0.8	5	1.5	1.9	2.2	0.3	5
CMT5	1	1356	4.1	4.6	4.9	0.3	5	1.9	2.8	3.4	0.6	5	0.9	1.3	1.5	0.3	5
	2	678	4.3	4.6	4.9	0.2	5	2.0	3.1	3.9	0.8	5	1.1	1.4	1.8	0.2	5
	3	452	4.2	4.4	4.7	0.2	5	1.6	2.1	2.8	0.4	5	0.8	1.3	1.5	0.3	5
	4	339	4.2	4.3	4.7	0.2	5	2.5	3.0	3.2	0.3	5	1.0	1.3	1.5	0.2	5
	5	271	4.0	4.2	4.6	0.3	5	2.5	3.1	4.1	0.2	5	0.7	1.2	1.5	0.4	5
	6	226	3.5	3.8	4.5	0.4	5	3.1	3.5	4.0	0.4	5	1.3	1.5	1.9	0.2	5
	7	194	2.9	3.7	4.3	0.6	5	3.6	3.8	4.1	0.2	5	1.4	1.7	2.1	0.3	5
	8	170	3.3	3.7	4.4	0.5	5	2.8	3.5	4.2	0.6	5	1.0	1.4	1.8	0.3	5
	9	151	3.6	4.0	4.5	0.5	4	3.4	3.7	4.0	0.3	4	1.3	2.0	2.8	0.6	5
	10	136	3.5	3.8	4.1	0.4	2	4.1	4.1	4.1	-	1	2.4	2.9	3.6	0.5	5
	1	1421	6.5	7.6	8.5	0.8	5	2.1	2.7	3.5	0.7	5	0.7	1.5	2.0	0.6	5
	2	710	5.3	6.1	7.1	0.8	5	1.8	2.4	2.9	0.5	5	1.3	1.8	2.0	0.3	5
	3	474	4.6	6.7	8.6	1.4	5	1.6	2.3	2.6	0.4	5	1.3	1.5	1.6	0.1	5
	4	355	4.9	5.9	8.2	1.4	5	1.3	2.4	3.6	1.0	5	1.5	1.6	1.9	0.2	5
	5	284	7.0	7.5	8.1	0.5	5	2.0	2.7	3.3	0.5	5	0.7	1.3	1.9	0.5	5
	6	237	4.4	6.9	8.1	1.4	5	1.5	2.7	4.1	1.0	5	1.1	1.5	2.0	0.3	5
	7	203	5.8	6.8	7.9	0.9	5	2.3	3.2	3.7	0.6	5	0.8	1.7	2.2	0.5	5
	8	178	5.7	7.1	8.2	1.3	5	1.8	2.0	2.3	0.2	5	1.3	1.5	1.6	0.1	5
	9	158	5.2	6.4	7.8	0.9	5	3.0	3.4	4.2	0.5	5	1.2	1.5	1.9	0.3	5
	10	142	4.5	5.9	6.7	0.9	5	4.1	4.6	5.0	0.4	5	1.3	2.0	2.3	0.4	5
CMT11	4	274	4.4	4.4	4.4	-	1	4.2	4.3	4.4	0.1	2	3.5	3.7	4.0	0.3	3
CMT12	5	172	3.7	3.7	3.7	-	1	3.1	3.3	3.1	0.1	2	3.2	3.4	3.7	0.4	2
F11	1	254	1.7	3.1	4.2	1.0	5	0.0	0.3	0.7	0.3	5	0.0	0.0	0.0	0.0	5
	2	127	3.9	4.2	4.5	0.2	5	4.2	4.3	4.3	0.1	4	3.7	3.7	3.7	0.0	5
	1	266	1.9	4.8	6.8	1.8	5	0.0	0.4	0.7	0.3	5	0.0	0.0	0.0	0.0	5
	2	133	5.2	6.9	9.3	1.7	5	0.0	0.8	1.8	0.8	5	0.0	0.0	0.0	0.0	5
	3	89	5.0	5.5	7.2	1.0	5	5.4	7.0	8.2	1.3	5	5.0	5.0	5.0	0.0	5
F12	1	1221	1.3	3.8	4.9	1.4	5	0.5	0.6	0.9	0.2	5	0.0	0.0	0.0	0.0	5
	2	611	2.7	3.9	4.6	0.9	5	0.7	0.9	1.1	0.2	5	0.0	0.0	0.0	0.0	5
	3	407	1.1	2.3	3.9	1.2	5	0.3	0.5	0.7	0.2	5	0.0	0.0	0.0	0.0	5
	1	1279	1.0	4.5	7.8	2.7	5	0.8	1.0	1.3	0.2	5	0.0	0.0	0.0	0.0	5
	2	640	1.0	4.7	8.9	3.0	5	0.7	0.9	1.0	0.1	5	0.0	0.0	0.0	0.0	5
	3	426	0.4	5.6	8.8	3.2	5	0.4	0.8	1.0	0.3	5	0.0	0.0	0.0	0.0	5
total feasible solutions found							261					262					271
		average	3.6	4.9	6.0	1.0		1.9	2.4	3.0	0.5		1.1	1.4	1.8	0.3	

Table 4.12: Comparison with OV on the 56 instances in G2

Paper	Machine	RAM
TLG	100 Mhz Silicon Graphics Idingo	-
BM	HP Vectra XU Pentium Pro 200 Mhz	-
SP	Ultra Enterprise 450 dual processor 300 Mhz	-
OV	1.8 Ghz AMD Athlon XP 2200+	480 Mb
AAB	DELL Dimensio 8200 1.6 Ghz	256 Mb

Table 4.13: Machines used in previous papers

Instance		Algorithm							
Name	#	TLG	BM	SP	OV	AAB	MA-F	MA	MA+CLS
CMT1	8	300	150	16	16	161	3	10	30
CMT2	14	420	300	30	29	221	4	25	118
CMT3	12	1440	600	70	27	459	1	52	173
CMT4	16	3060	1500	206	68	681	31	169	493
CMT5	20	3960	3750	484	125	870	37	354	1284
CMT11	10	2700	1500	1132	28	527	12	99	302
CMT12	12	1380	600	45	27	414	10	37	138
F11	6	1560	150	93	13	✘	5	21	40
F12	6	4500	4800	584	31	✘	0	87	160

Table 4.14: CPU times comparison. Times expressed in seconds

The method finds a feasible solution over 99 instances, one more than all the previous works (that have failed in finding a feasible solution for instance CMT4_T_H^1_7). Solutions found are always better than those reported by Salhi and Petch [186] (the only paper which gives detailed results). *GAP* values are on average better than those reported by Olivera and Viera [156].

The proposed algorithm could be extended to the MTVRP with time windows introducing slight modifications into the *AdSplit* procedure explained in Section 4.3.2, in moves M1–M10 and in the CLS. This will be the subject of future research.

Chapter 5

The Multi Trip Vehicle Routing Problem With Time Windows and Release Dates

Abstract

The Multi-Trip Vehicle Routing Problem with Time Windows and Release Dates is a variant of the Multi-Trip Vehicle Routing Problem where a time windows is associated with each customer and a *release date* is associated with each merchandise to be delivered at a certain client. The release date represents the moment the merchandise becomes available at the depot for final delivery.

The problem is relevant in city logistics context, where delivery systems based on city distribution centers (CDC) are studied. Trucks arrive at the CDC during the whole working day to deliver goods that are transferred to eco-friendly vehicles in charge of accomplish final deliveries to customers.

We propose a population-based algorithm for the problem based on giant tour representation of the chromosomes as well as a *split* procedure to obtain solutions from individuals.

5.1 Introduction

The well-known Vehicle Routing Problem (VRP) is an $\mathcal{NP}$-hard combinatorial optimization problem where a set of geographically scattered customers has to be served by a fleet of vehicles minimizing routing costs and respecting capacity constraints on vehicles. The VRP represents a simplified problem that is usually far from the reality of freight distribution. To better represent real world problems, different aspects need to be taken into account leading to more complex problems, usually called *rich* (or *multi-attribute*) problems.

In this paper, we introduce a new variant of the VRP, the Multi Trip Vehicle Routing Problem with Time Windows and Release Dates. Our interest for this problem originates from mutualized distribution in cities, where external goods continuously arrive in a City Distribution Center (CDC) from where the last-mile delivery is operated

In this context, vehicle distribution from the CDC should be optimized on a daily basis. As vehicles have preferably small capacities and the fleet size should be minimized, vehicles will typically perform several trips along the day. This introduces the *multi trip* aspect.

Customers usually ask to be served within a certain time interval. Meeting these intervals is vital for the carrier: delays mean losing reliability and trustworthiness and often means paying a penalty. Then, *time windows* should be considered and associated with each customer.

Finally, merchandise can be delivered to the CDC all day long. This means that they are not necessarily available at the CDC at the beginning of the planning horizon. Vehicle routes must then be designed such that no vehicle leaves the CDC before the goods it has to transport in its trip have arrived. The concept of *release date* is associated with each merchandise, indicating the time at which the merchandise is available at the CDC.

These attributes together lead to the Multi Trip Vehicle Routing Problem with Time Windows and Release Dates (MTVRPTWR). It is noteworthy that the problem is *static* (or *off-line*) even if the merchandise continuously arrives to the CDC during the day. In fact, the release dates are supposed to be known before the working day starts.

Online variants can consider requests that become known at time intervals differently long before the respective release date. This aspect raises the issue of the value of the information. These variants, as well as stochastic variants, where the actual availability of the merchandise might differ from the release date, are left for future research. To the best of our knowledge, this is the first time release dates are considered in a routing problem. Discussions with researchers from several software companies revealed that this issue, coupled with multiple trips and time windows, is actually very relevant in practice and models a problem that companies have to deal with (Grunert [96] and Kleff [120]).

The MTVRPTWR implicitly models the dependence between different levels in multi-level distribution systems. The particular case of two-level distribution systems (called as well two-echelon or two-tier distribution systems, Hemmelmayr et al. [104], Crainic [49]) has recently been investigated by scholars. In these systems, final delivery trips depend on the time and on the CDC where vehicles operating in the first level unload goods. Information sharing and synchronization between the two levels of distribution are, however, assumed. On the contrary, the MTVRPTWR models a situation where the first level of distribution cannot be controlled. The purpose of this paper is to contribute filling this gap introducing a new routing problem that takes into account this aspect as well as proposing an efficient procedure to solve it.

The paper is organized as follows. In Section 5.2 the problem is formally defined and characterized. Section 5.3 reviews related research. Section 5.4 describes a heuristic solution. Instance sets and results are presented in Section 5.5, while Section 5.6 concludes the paper.

5.2 Problem definition and notation

This section defines the problem (Section 5.2.1) and introduces the notation used in the rest of the paper (Section 5.2.2). Moreover, relations with the VRP with pickup and delivery are presented in Section 5.2.3. Finally, a characterization of problems with release dates is proposed in Section 5.2.4.

5.2.1 Problem definition

The MTVRPTWR can be defined on a complete undirected graph $G = (V, E)$, where $V = \{0, \dots, N\}$ is the set of vertices and $E = \{(i, j)|i, j \in V, i < j\}$ the set of edges. Vertex 0 represents the depot, where a fleet of M identical vehicles with capacity Q is based. Vertices $1, \dots, N$ represent the customers. With each customer is associated a demand Q_i that needs to be delivered during a time window (TW) indicated by $[E_i, L_i]$. Service at customer i takes S_i and must start during the TW. Service time S_0 at the depot is usually called loading time. Arriving at customer location before E_i is allowed. Since the service cannot start earlier than E_i, the driver must wait. On the other side, late arrival at customer location is forbidden. Moreover, the quantity Q_i of product requested by customer i is available at the depot not earlier than R_i. R_i is called the *release date*. For brevity, we will say *R_i is associated with customer i*, instead of *R_i is associated with the quantity Q_i requested by customer i*. It is possible to travel from i to j incurring a travel time T_{ij} and covering a distance D_{ij}, $i, j = 0, \dots, N$.

It is noteworthy that the release dates introduced here do not have the same implication as those considered in scheduling problems (called as well release or ready times, Błażewicz [22]). In that case, the release date is the earliest time at which job can be processed on a machine (Pinedo [163]). A vehicle needs to wait until all goods it has to carry are available at the depot, i.e., it cannot start the trip before the maximal release date associated with the customers it has to serve. On the other hand, a machine can start processing available jobs without waiting for all to be ready.

A time horizon T_H is given and establishes the duration of the working day. It can be viewed as a TW associated with the depot. Thus, it is assumed that $[E_0, L_0] = [0, T_H]$, $Q_0 = 0$. Operations cannot start before E_0 and all vehicles must be back to the depot at time L_0.

The MTVRPTWR calls for the determination of a set of trips and an assignment of each trip to a vehicle, such that the total travel distance is minimized and the following conditions are satisfied:

(1) each trip starts and ends at the depot;

(2) trips do not start earlier than $E_0 = 0$ and finish later than $L_0 = T_H$;

(3) no trips assigned to the same vehicle overlap;

(4) operations of each trip start not earlier than the greater R_i associated with customers assigned to the trip itself;

(5) each customer is visited by exactly one trip;

(6) service at customer i starts in the associated range $[E_i, L_i]$;

(7) the sum of the demands of the customers served by any trip does not exceed Q.

It is supposed that each customer i can be served by a return trip, i.e, $R_i + T_{0i} \leq L_i$, $\max\{E_i, R_i + T_{0i}\} + S_i + T_{i0} \leq T_H$ and $Q_i \leq Q$ (otherwise no feasible solution would exist).

5.2.2 Notation

The notation used in the rest of the paper is introduced here. The symbol σ will always indicate a trip. The capital Σ will indicate the set of trips assigned to a vehicle. It will be called *journey* in the following. A journey is formed by different trips. The symbol $\oplus$ is used to indicate the concatenation of paths (partial trips) or trips. For example $(v_1, \ldots, v_n) \oplus (w_1, \ldots, w_m)$ is the concatenation of two paths (that results in a trip if v_1 and w_m are the depot). $\sigma_1 \oplus \sigma_2$ means that trip σ_1 is performed right before σ_2 by the same vehicle.

The time a vehicle is available at the depot to perform a given trip σ, is indicated as T^σ. It is noteworthy that the vehicle cannot start trip σ before $\max\{T^\sigma, \max_{v\in\sigma} R_v\}$. Here and in the next sections, the symbol "$\in$" will be used to describe "belonging". For example $\sigma \in \Sigma$ means trip σ belongs to journey Σ, $v \in \sigma$ means customer v belongs to (i.e., it is served by) trip σ, and so on.

The service of a customer i will be called *feasible* if the vehicle arrives at its location before L_i, *infeasible* otherwise. A trip σ is called feasible if service at each customer in σ is feasible, infeasible otherwise. A journey is called feasible if it is composed by feasible trips, infeasible otherwise.

5.2.3 Relationship with the VRPPDTW

In the VRP with pickup and delivery (VRPPD), requests $i = 1, \ldots, N$ need to be picked up at defined locations $PICK_i$ before delivery takes place at location DEL_i. Precedence constraints impose pickups to be performed before deliveries. The VRPPD with time windows (VRPPDTW) asks pickups and deliveries to take place during a TW. The MTVRPTWR can be reduced to an extension of the VRPPDTW where multiple trips are allowed. Given an instance of the MTVRPTWR, N requests are defined such that $PICK_i$ is the depot for every request and DEL_i is the location of customer i. The pickup TW associated with request i is set to $[R_i, T_H]$, the delivery TW is set to $[E_i, L_i]$. Travelling from the depot to a pickup location and between pickup locations does not require time, i.e., $T_{0PICK_i} = 0$, $T_{PICK_i0} = 0$ and $T_{PICK_iPICK_j} = 0$. On the other hand $T_{PICK_iDEL_j} = T_{0j}$ and $T_{DEL_iDEL_j} = T_{ij}$. Distances are managed similarly.

5.2.4 Problem characterization

Larsen [128] and Larsen et al. [129] introduce some parameters to characterize problems in the context of the Dynamic VRP (DVRP). Even if there is no exact correspondence between the DVRP and the MTVRPTWR, these problems share similarities. In the DVRP, a request becomes known at a certain time called *disclosure time* (Pillac et al. [162]), while in the MTVRPTWR, merchandise becomes available at a certain moment that we call *release date*. The main difference is that all the release dates are known at the beginning of the working day, that makes the MTVRPTWR a static problem. On the other side, requests in the DVRP context become known during operations.

We introduce the *rigidity* of a system (which corresponds to the degree of dynamism for the DVRP) as

$$r = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{L_i - R_i}{T_H}\right). \tag{5.1}$$

Since $R_i \leq L_i$, $r \in [0, 1]$. It represents how close are the R_i to the corresponding L_i on average. In particular, $r = 0$ when $R_i = 0$ and $L_i = T_H$ for all customers i. On the other side $r = 1$ when $R_i = L_i$ for all customers i. A considerable amount of release dates R_i close to the corresponding L_i thus makes the plan more rigid than having the release dates R_i far in time from L_i.

It can be noticed that, when $R_i = 0$ for all customers i, the rigidity r can be strictly positive since it depends on the relationship between L_i and T_H. We introduce another parameter that takes into account only the relationship between R_i and L_i. We define the *instance tightness* as

$$tight = \frac{1}{N}\sum_{i=1}^{N}\frac{R_i}{L_i}. \tag{5.2}$$

The parameter *tight* equals zero when $R_i = 0$ for all customers i and equals one when $R_i = L_i$ for all customers i. Practically, r and *tight* are strictly less than 1: all $R_i = L_i$ would make the problem infeasible as long as travel times between the depot and the customers are not null.

5.3 Literature Review

To the best of our knowledge, there is no routing problem considering release dates. On the other side, the Multi Trip VRP with Time Windows (MTVRPTW) is an extension of the well-known Vehicle Routing Problem with Time Windows (VRPTW) where customers must be served during a time interval called *time window* (Bräysy and Gendreau [27, 28]) and the Multi Trip VRP, that allows vehicles to perform several trips during the working day (Cattaruzza et al. [33], Olivera and Vera [156], Taillard et al. [198]).

The MTVRP is a $\mathcal{NP}$-hard problem (Olivera and Viera [156]). This makes the MTVRPTW a $\mathcal{NP}$-hard problem (the MTVRP can be reduced to the MTVRPTW associating a TW equal to $[0, T_H]$ with each customer). Finally, the MTVRPTWR is $\mathcal{NP}$-hard since the MTVRPTW trivially reduces to the MTVRPTWR setting all the release dates to zero.

Despite its practical interest, the literature on the MTVRPTW is pretty scarce and most of the existing papers propose exact methods. Azi et al. [11] propose an exact algorithm for solving the single vehicle MTVRPTW. The solution approach exploits an elementary shortest path algorithm with resource constraints. In the first phase all non-dominated paths are calculated. Then the shortest path algorithm is applied to a modified graph. Each node represents a non-dominated trip and two nodes are connected by an arc when it is possible to serve the two trips consecutively and they do not serve common customers. Solomon's instances are used with different values of time horizon. 16 instances out of 54 with 100 customers are solved to optimality.

Azi et al. [12] address the MTVRPTW where trips are constrained by a limited duration and serve all customers is not mandatory. A column generation approach embedded within a branch-and-price algorithm is proposed. A set packing formulation is given for the master problem and each column represents a working day. Since each pricing problem is an elementary shortest path with resource constraints, a similar approach to the one proposed in Azi et al. [11] is applied. As in Azi et al. [11], Solomon's instances are considered. Due to the limitations of the algorithm, the authors focus on instances formed by the first 25 or 40 customers of each Solomon's instance. The algorithm can also solve few instances of size 50. Hernandez et al. [106] use a similar approach. A set covering formulation is given for the problem and each column represents a trip instead of a working day. With this method applied on the same instances proposed in Azi et al. [12] optimal solutions are found for a majority of instances with size up to 50. Macedo et al. [136] propose a minimum flow model where variable represent feasible trips. Optimal solutions are found for a majority of instances with size up to 50 in 2 hours of computation time. In all the previous works, the full set of feasible trips is generated. This is practicable due to the presence of time windows and trip duration constraint that limit the cardinality of the set.

Hernandez et al. [105] propose the only exact method on the MTVRPTW. A branch-and-price algorithm is proposed and instances up to 50 customers and 4 vehicles can be solved.

Battarra et al. [17] study an extension of the MTVRPTW where products are clustered in different commodities that cannot be transported in the same vehicle. They first generate a set of feasible trips considering each commodity independently. Then, these trips are assigned to vehicles in order to obtain a solution.

5.4 Method

This section describes the hybrid genetic algorithm $\mathcal{A}^{CAF}$ that we developed for the MTVRPTWR. In genetic algorithms (GA), a set of *chromosomes* forms a population that evolves across generations until termination criteria are met. New individuals (*children*) are generated from those in the current population, called *parents*, by *crossover* and *mutation* operators. GA turned out to be highly efficient heuristic methods to face different problems, mainly because children generation allows to explore new zones (children differ from parents) that are promising (children keep good parents' characteristics). The papers of Prins [166], Vidal et al. [213] and Cattaruzza et al. [33] are three examples of efficient GA (respectively for the VRP; the multi depot VRP and the periodic VRP; and for the multi trip VRP) in the VRP field.

The principles of genetic procedure were first formalized by Holland [109]. The interested reader is referred to Reeves [177], Moscato and Cotta [142] and Neri and Cotta [147] for an overview of population based procedures.

5.4.1 Algorithm outline

This section outlines our $\mathcal{A}^{CAF}$ for the MTVRPTWR. Initially, a population Π of π chromosomes is generated. Chromosomes are sequences of client nodes without trip delimiters. Each chromosome is evaluated with an adaptation of the Split procedure developed by

Prins [166], that we call *AdSplit* (Section 5.4.4). *AdSplit* obtains a MTVRPTWR solution ξ from a chromosome Ψ. The fitness of chromosome Ψ is then defined as the cost of the solution ξ.

All individuals are improved by means of a local search (LS) procedure called as well *education* phase (Section 5.4.3). At each iteration two chromosomes Ψ_{P_1} and Ψ_{P_2} are selected using the classical binary tournament procedure. Two children Ψ_{C_1} and Ψ_{C_2} are obtained crossing parents with the order crossover procedure and one (Ψ_C) is randomly selected between them. Ψ_C undergoes *AdSplit* for evaluation and LS for education. It is then inserted in Π.

After μ children are inserted in Π, π chromosomes are selected based on their fitness (representing their quality) and diversification contribution to the population itself, while the other μ are eliminated (survivor procedure, Section 5.4.5). A sketch of the method is given in Algorithm 4.

Algorithm 4 $\mathcal{A}^{CAF}$ outline

```
Initialize population
while Termination criteria are not met do
    Select parent chromosomes Ψ_P1 and Ψ_P2
    Generate a child Ψ_C
    Educate Ψ_C
    Insert Ψ_C in the population
    if Dimension of the population exceeds a given size then
        Select survivors
    end if
end while
```

5.4.2 Solution representation and search space

A chromosome is a sequence (permutation) $\Psi = (\Psi_1, \ldots, \Psi_N)$ of the N client nodes, without trip delimiters. Ψ can be viewed as a TSP solution that has to be turned into a feasible MTVRPTWR solution by splitting the chromosome (inserting trip delimiters and assigning trips to vehicles). Ψ is usually called a *giant tour*. From a giant tour Ψ, different MTVRPTWR solutions can be constructed depending on the way Ψ is split.

During the search phase, overload and TW violations are allowed and penalized in the fitness function. Two penalization factors are needed: θ for TW violation and λ for load infeasibility.

A procedure *AdSplit* (explained in Section 5.4.4) is used to get a MTVRPTWR solution ξ from Ψ. The following notation is introduced: $D_\Sigma(\xi)$ and $TW_\Sigma(\xi)$ are respectively the traveled distance and the TW violation of journey Σ in solution ξ. $L_\sigma(\xi)$ is the load of trip σ. The fitness $F(\Psi)$ of the chromosome Ψ is the cost $c(\xi)$ of the solution ξ found by *AdSplit* and it is defined as

$$F(\Psi) = c(\xi) = \sum_{\Sigma=1}^{M} D_\Sigma(\xi) + \theta \sum_{\Sigma=1}^{M} TW_\Sigma(\xi) + \lambda \sum_{\Sigma=1}^{M} \sum_{\sigma \in \Sigma} \max\{0, L_\sigma(\xi) - Q\}. \tag{5.3}$$

When confusion cannot arise, solution ξ will be omitted in the notation. The chromosome Ψ is called *feasible* (*infeasible*) if *AdSplit* obtains, from Ψ, a feasible (infeasible) solution ξ.

In practice a chromosome is split in order to obtain a solution and then it possibly undergoes LS for improvement. It would then be natural to continue describing the solution method with the *AdSplit* procedure. However, *AdSplit* takes advantages of aspects in the LS. Thus, we will start presenting the latter.

5.4.3 Local search

This section presents the local search (LS) procedure embedded in $\mathcal{A}^{CAF}$. First, an efficient scheme to manage TW violations is presented (Section 5.4.3) based on Vidal et al. [214] work. In Section 5.4.3 peculiar characteristics of our problem are introduced, and Section 5.4.3 describes the general LS procedure.

Local Search for VRP with TW

LS in presence of TW becomes more complicated than in the classic VRP. Feasibility checks and routing cost variations cannot be straightforwardly calculated in constant time. Savelsbergh [187] proposes a scheme to check feasibility and to calculate cost variation for series of k-opt moves. Nagata et al. [145] propose a new scheme to evaluate TW violations. Roughly speaking, when a vehicle arrives late at the customer location, it is allowed to drive back in time in order to meet the TW and a penalization proportional to the TW violation is introduced in the objective function. They also propose formulas to evaluate in constant time inter-route moves as relocation, exchange and 2-opt (inter-route 2-opt is usually called 2-opt*)[1].

Vidal et al. [214] generalize this scheme for a large class of routing problems with TW. Each move is seen as a concatenation of paths. In particular, given a path ρ, the quantities $T(\rho)$, $TW(\rho)$, $E(\rho)$, $L(\rho)$, $D(\rho)$ and $Q(\rho)$ are introduced. $T(\rho)$ and $TW(\rho)$ are respectively the minimum duration and the minimum penalization (called as well *time warp*) of ρ. $E(\rho)$ and $L(\rho)$ are the earliest and the latest date service can start at the first customer of ρ (that can be the depot) allowing minimum duration and TW violation. $D(\rho)$ is the travelled distance while $Q(\rho)$ is the cumulative demand of served customer. We will call these quantities *features*. For a path made by a single customer i, features are initialized as follows: $T(i) = S_i$, $TW(i) = 0$, $E(i) = E_i$, $L(i) = L_i$, $D(i) = 0$, $Q(i) = Q_i$. Given two paths ρ_1 and ρ_2, the following relations hold (see Vidal et al. [214] for formal proofs):

$$T(\rho_1 \oplus \rho_2) = T(\rho_1) + T(\rho_2) + T_{v^1_{n_1}, v^2_1} + \Delta_{WT}; \tag{5.4}$$

$$TW(\rho_1 \oplus \rho_2) = TW(\rho_1) + TW(\rho_2) + \Delta_{TW}; \tag{5.5}$$

$$E(\rho_1 \oplus \rho_2) = \max\{E(\rho_2) - \Delta, E(\rho_1)\} - \Delta_{WT}; \tag{5.6}$$

$$L(\rho_1 \oplus \rho_2) = \min\{L(\rho_2) - \Delta, L(\rho_1)\} + \Delta_{TW}; \tag{5.7}$$

[1] The formula proposed for inter-route relocations provides incorrect results in particular cases and it has been corrected by Schneider et al. [188].

$$D(\rho_1 \oplus \rho_2) = D(\rho_1) + D(\rho_2) + D_{v^1_{n_1}, v^2_1}; \tag{5.8}$$
$$Q(\rho_1 \oplus \rho_2) = Q(\rho_1) + Q(\rho_2); \tag{5.9}$$

where

$$\Delta = T(\rho_1) - TW(\rho_1) + T_{v^1_{n_1}, v^2_1};$$
$$\Delta_{WT} = \max\{E(\rho_2) - \Delta - L(\rho_1), 0\};$$
$$\Delta_{TW} = \max\{E(\rho_1) + \Delta - L(\rho_2), 0\}.$$

Equations (5.4)–(5.9) allow evaluating classical LS moves in constant time. For example, relocate customer v_i from trip $\sigma_1 = (0, v_1, \ldots, v_{i-1}, v_i, v_{i+1}, \ldots, v_{n_1}, 0)$ to trip $\sigma_2 = (0, w_1, \ldots, w_j, w_{j+1}, \ldots, w_{n_2}, 0)$ between customers w_j and w_{j+1} can be evaluated applying Equations (5.4)–(5.9) to $(0, v_1, \ldots, v_{i-1}) \oplus (v_{i+1}, \ldots, v_{n_1}, 0)$ and $(0, w_1, \ldots, w_j) \oplus (v_i) \oplus (w_{j+1}, \ldots, w_{n_2}, 0)$.

Features defined in Equations (5.4)–(5.9) need to be available for all the paths and their reverse present in the current solution. Reverse paths come into play, for example, in the evaluation of 2-opt moves. Moreover, these quantities need to be updated each time a move is implemented. Straightforward update takes $O(N^2)$, but it can be done in $O(N^{8/7})$ using speed up techniques proposed by Irnich [115].

We use this approach (except for the little modification introduced by the release dates, Section 5.4.3) to evaluate *local* variations due to a move. By *local* we mean a cost variation that occurs in the trips affected by the move itself. It is noteworthy that this variation can affect successive trips in the same journey (Section 5.4.3).

Local search: introduction of release dates and application to the multi trip case

The MTVRPTWR has two more attributes to consider compared to the VRPTW: vehicles can perform several trips, and merchandise can be non-available at the depot at the beginning of the horizon.

We introduce a new feature $R(\rho)$ as the greatest release date of customers served in ρ. We define $R(i) = R_i$ and Equation (5.10) holds

$$R(\rho_1 \oplus \rho_2) = \max\{R(\rho_1), R(\rho_2)\}. \tag{5.10}$$

Then, given two paths ρ_1 and ρ_2, the quantity $R(\rho_1 \oplus \rho_2)$ can be calculated in constant time from values $R(\rho_1)$ and $R(\rho_2)$.

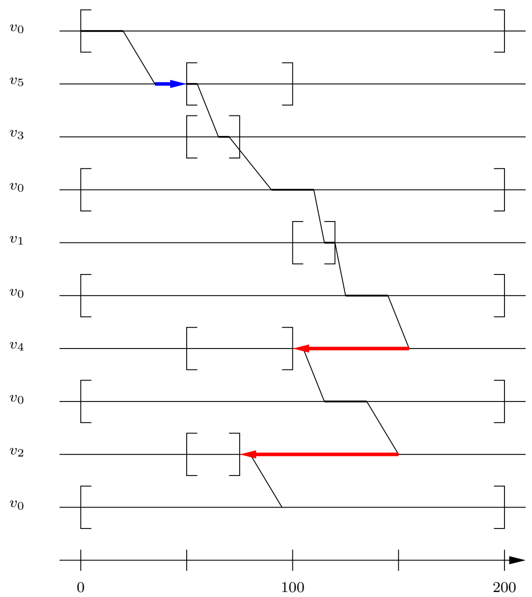
It is observed that in the VRP it is always possible to start a new trip σ at time t, with $E(\sigma) \leq t \leq L(\sigma)$, because all the vehicles are available at the depot at $t = 0$ and are assigned to only one trip. Then, the minimum travelling duration time $T(\sigma)$ and the minimum TW violation $TW(\sigma)$ can always be obtained. On the other side, this cannot be possible when vehicles are allowed to perform more than one trip or in the presence of release dates. In the first case it can be that $L(\sigma) < T^\sigma$ because of previous trips, in the second case, it can be that $L(\sigma) < R(\sigma)$ (that means trip σ is infeasible). We, thus, need to calculate $T(\sigma)$ and $TW(\sigma)$ based on the effective time t a vehicle leaves the depot to perform σ. To do that, we can use the following relations proved by Vidal et al. [214] that provide the value of these quantities as functions of the starting time t:

$$T(\rho)(t) = T(\rho) + \max\{0, E(\rho) - t\}; \tag{5.11}$$
$$TW(\rho)(t) = TW(\rho) + \max\{0, t - L(\rho)\}. \tag{5.12}$$

Finally, the time T^{σ_i} a vehicle is available at the depot to perform trip σ_i can be recursively calculated as follows

$$T^{\sigma_1} = 0; \tag{5.13}$$
$$T^{\sigma_{i+1}} = T(\sigma_i)(\max\{R(\sigma_i), T^{\sigma_i}\}) - TW(\sigma_i)(\max\{R(\sigma_i), T^{\sigma_i}\}). \tag{5.14}$$

The Figure 5.1 introduces an example (data are given in tables, travel times and distances coincide) that illustrates a consequence of using Nagata et al. [145] penalization scheme for TW violations in the multi-trip context. In particular Figure 5.1 depicts a journey formed by four trips: $\sigma_1 = (v_0, v_5, v_3, v_0)$, $\sigma_1 = (v_0, v_1, v_0)$, $\sigma_3 = (v_0, v_4, v_0)$ and $\sigma_4 = (v_0, v_2, v_0)$ where v_0 represents the depot. It can be noticed that $115 = T^{\sigma_4} < T^{\sigma_3} = 125$ and $95 = T^{\sigma_5} < T^{\sigma_4}$. This means that the vehicle can be available at the depot to perform trip σ_{i+1} before the time it was available to perform trip σ_i. Indeed, it travelled back in time and it is the consequence of a deep use of the time warp, i.e., a deep time window violation.



	S_i	Q_i	E_i	L_i	R_i
v_0	20	0	0	200	0
v_1	5	20	100	120	60
v_2	5	20	50	75	0
v_3	5	20	50	75	0
v_4	5	20	50	100	60
v_5	5	20	50	100	0

Distance matrix						
	v_0	v_1	v_2	v_3	v_4	v_5
v_0	0	5	15	20	10	15
v_1	5	0	20	20	15	15
v_2	15	20	0	40	20	30
v_3	20	20	40	0	30	10
v_4	10	15	20	30	0	20
v_5	15	15	30	10	20	0

Figure 5.1: The last two trips arrive at the depot before they have left it

One needs also to observe that the effect of a move on trip $\sigma_i \in \Sigma$ can modify $T^{\sigma_{i+1}}$, that in turn can modify $T(\sigma_{i+1})$ and $TW(\sigma_{i+1})$. Hence, the effect of the move on trip σ_i can be propagated to the following trips.

When loading times at the depot are constant and not trip-dependent, evaluation of the TW violation can be done in constant time, considering a journey as a unique segment with multiple visits at the depot. Conversely, the calculation requires $O(\Sigma_{\max})$, where $\Sigma_{\max}$ indicates the maximum number of trips among all journeys.

Local search: general scheme and penalty adaptation

We consider the move 2-opt and the move ex_ρ that exchanges two segments ρ_1 and ρ_2 of successive customers, where ρ_1 and ρ_2 are respectively subsequences of trips σ_1 and σ_2. Let v_1 and v_2 be two customers. In particular, it can be observed that if:

- $\rho_1 = v_1$ and $\rho_2 = \emptyset$, ex_ρ is the classical relocate;
- $\rho_1 = v_1$ and $\rho_2 = v_2$, ex_ρ is the classical exchange (or swap);
- $\rho_1 = (v_1, \ldots, 0)$ and $\rho_2 = (v_2, \ldots, 0)$, or $\rho_1 = (0, \ldots, v_1)$ and $\rho_2 = (0, \ldots, v_2)$, and $\sigma_1 \neq \sigma_2$, ex_ρ is the classical 2-opt*;
- $\rho_1 = \sigma_1$ and $\rho_2 = \emptyset$, ex_ρ relocates a full trip;
- $\rho_1 = \sigma_1$ and $\rho_2 = \sigma_2$, ex_ρ exchanges trips.

The last two moves, take into account the multi-trip aspect. Relocations and swaps of trips are inter-vehicle (resp. intra-vehicle) moves if σ_1 and σ_2 belong to a different (resp. the same) journey. No limitation on the segment size is considered. Different neighbour structures considering a maximum segment size could speed up the search against a payback due to the possible lower quality of the obtained solutions.

Initially, a neighbourhood defined by one of the listed move is randomly chosen and deeply explored. The best improving move is then executed, if it exists. In this case, another neighbourhood is randomly chosen. The LS terminates when all the neighborhoods have been fully explored without finding any improving move.

When the population reaches the dimension of $\pi + \mu$, the values of each penalty factor $P = \theta, \lambda$ is adjusted as follows:

$$P = \begin{cases} 1.25 \times P \quad \text{if } \zeta_P > \zeta^+_{ref}; \\ \max\{1, 0.85 \times P\} \quad \text{if } \zeta_P < \zeta^-_{ref}, \end{cases}$$

where ζ_P is the percentage of infeasible chromosomes over the last μ individuals generated with respect to time window violation ($P = \theta$) or capacity violation ($P = \lambda$), while ζ^+_{ref} and ζ^-_{ref} are reference values.

5.4.4 A Split algorithm for multi-trip problems with time windows and release dates

The split procedure, indicated with *AdSplit*, is an adaptation of the procedure proposed by Prins [166]. It turns a permutation of the N customers into a solution for the MTVRPTWR. It works on an acyclic graph H whose nodes represent customers, and arcs represent trips. The graph construction is illustrated in Section 5.4.4, while Section 5.4.4 presents the *AdSplit* procedure.

Auxiliary graph construction

The auxiliary graph $H = (V', A')$ is defined with $N+1$ nodes indexed from 0 to N. Node i represents customer Ψ_i in the chromosome $\Psi = (\Psi_1, \ldots, \Psi_N)$. Arc (i,j), $i < j$, represents trip σ_{i+1}^j serving customers from Ψ_{i+1} to Ψ_j in this order, i.e., $\sigma_{i+1}^j = (0, \Psi_{i+1}, \ldots, \Psi_j, 0)$. Cost c_{ij} of arc (i,j) is given by the following equation

$$c_{ij} = D(\sigma_{i+1}^j) + \theta TW(\sigma_{i+1}^j) + \lambda \max\{0, Q(\sigma_{i+1}^j) - Q\},$$

that is the sum of the travelled distance in trip σ_{i+1}^j plus the penalized TW and capacity violations. Figure 5.2 depicts graph H for the example in Figure 5.1, where $\Psi = (1,2,3,4,5)$. Cost of arc $(0,1)$ is twice the travelling distance needed to go from the depot to customer

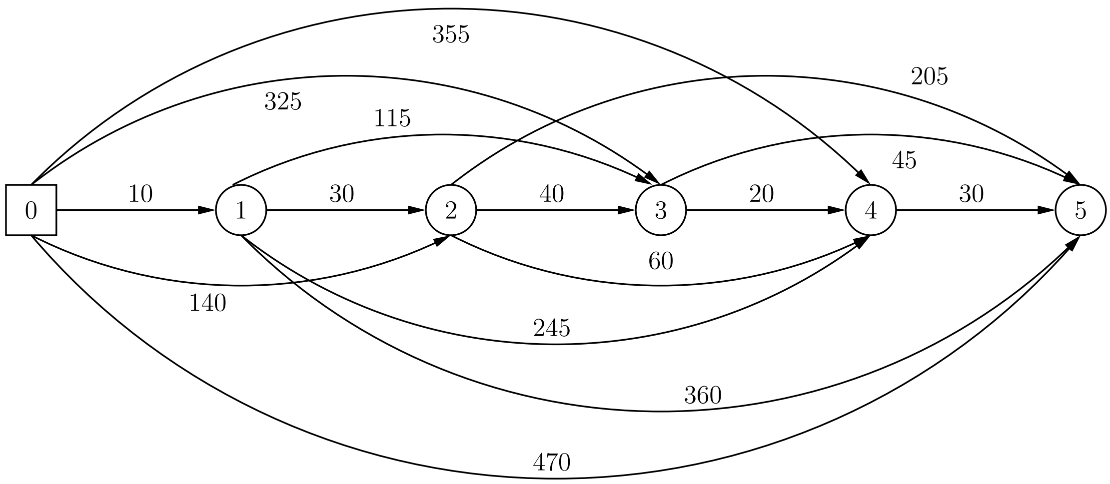


Figure 5.2: The auxiliary graph for chromosome $\Psi = (1,2,3,4,5)$, data as in Figure 5.1, $Q = 60$ and $\theta = \lambda = 2$

1, i.e., 10. Cost of arc (0,2) is the travelling distance needed to reach customer 1 from the depot, then visit customer 2 and to conclude the trip at the depot, i.e., 40. The penalization $50 \cdot \theta$ is added since the vehicle arrives at customer 2 at 125 and its time window closes at 75. Then, for $\theta = 2$, arc $(0,2)$ costs 140. Other costs are computed similarly.

It is noteworthy that the arc cost does not take into account the position of the trip in the journey, but it is the (penalized) cost of the trip when it is performed at $t = 0$. Therefore, the (contingent) TW violation due to later departure is not taken into account.

Assignment of trips to vehicles

In the MTVRPTWR context in particular and in the MTVRP context in general, more than one trip can be assigned to the same vehicle. TW penalization deeply depends on the time trips leave the depot and this aspect cannot be considered by the constant costs associated with arcs on H.

Due to the correspondence between arcs and trips in H and to simplify the presentation, in this section we will indifferently refer to arc or trip. Consequently, the assignment of arcs to a vehicle, will mean that the trips represented by these arcs are assigned to this vehicle.

Each path that goes from node 0 to node N, represents a set of trips, that need to be assigned to vehicles to form a solution. Arcs are assigned to a vehicle in the order they appear in the path. For each arc, M assignments (one for each vehicle) are possible. In each case, the trip is positioned after the trips already assigned to the vehicle. Hence, the order of trips in vehicles meets the order in Ψ.

The aim of the *AdSplit* procedure is to find the path whose optimal assignment of trips to vehicles results in the best solution. It can be shown that the problem solved with the *AdSplit* procedure in $\mathcal{NP}$-hard.

Let us consider the problem $SPLIT$ that consists in finding the optimal solution for an auxiliary graph H (constructed as explained in Section 4.4.1) defined by an instance I of the MTVRPTWR and a permutation Ψ of customers in I. The objective of $SPLIT$ is to find the optimal solution ξ_Ψ associated with Ψ on graph H. In what follows we analyze the complexity of the $SPLIT$ problem.

Proposition 3. *The problem $SPLIT$ is $\mathcal{NP}$-Hard.*

Proof. We perform a reduction from the Bin Packing Problem (BPP). The decision version of the BPP is defined given M^{BPP} bins of size Q^{BPP} and N^{BPP} objects of size $Q_i^{BPP}, i = 1, \dots, N^{BPP}$. We need to answer the question: can the N^{BPP} objects be inserted in the M^{BPP} available bins respecting capacity constraints?

The key idea of the reduction is to define an instance of the $SPLIT$ problem such that each arc (i, j) corresponds to the selection of objects $\{i + 1, \dots, j\}$ of the BPP.

Given an instance I^{BPP} of the BPP, an instance of the MTVRPTWR problem is defined with the following parameters: $N = N^{BPP}$, $T_{ij} = Q_j^{BPP}$ for all $i, j = 1, \dots, N$, and $T_{0i} = Q_i^{BPP}$, $T_{i0} = 0$, $[E_i, L_i] = \left[0, Q^{BPP}\right]$, $R_i = 0$, for all $i = 1, \dots, N$ and $Q = +\infty$. We call I^{SPLIT} the instance of the decision version of problem $SPLIT$ obtained when applying *AdSplit* to the chromosome $(1, \dots, N)$ for this instance.

The graph H associated with the I^{SPLIT} instance is constructed such that the cost on each arc (i, j) is given by $c_{ij} = \sum_{k=i+1}^{j} Q_k$ for all $i, j = 0, \dots, N$, $i < j$. We can notice that all paths from 0 to N in this graph have the same cost $(\sum_{k=1}^{N} Q_k)$. From the definition of I^{SPLIT}, it is obvious that the instance I^{BPP} can be polynomially transformed into an I^{SPLIT} instance.

We show that the answer for instance I^{BPP} is positive if and only if there exists a feasible solution of the $SPLIT$ problem that uses at most M^{BPP} vehicles.

If I^{BPP} is a positive instance, then there exists an assignment of objects to bins such that the number of used bins is lower than or equal to M^{BPP}. Considering the path in H that corresponds to all the arcs $(i - 1, i), i = 1, \dots, N$ and assigning trips $(0, i, 0)$ to the k^{th} vehicle if the i^{th} object is into the k^{th} bin, constructs a feasible solution for I^{SPLIT}.

On the other side let suppose that I^{SPLIT} is a positive instance. There exists a path ρ from 0 to N and an assignment of arcs to the M^{BPP} vehicles such that the sum of the duration of the trips assigned to a vehicle does not exceed the time limit Q^{BPP}. A solution for the I^{BPP} instance is constructed assigning object i to bin k if customer i is assigned to the k^{th} vehicle in ρ. The number of used bins is lower than or equal to M^{BPP} and their capacities are satisfied, which concludes the proof.

□

We propose a labelling procedure that selects arcs on H and assigns them to vehicles, in order to obtain the solution with minimum cost with respect to Equation (5.3).

Labels are associated with nodes in the graph. Each label associated with node i represents a partial path that goes from node 0 to node i in H and, then, a partial solution that serves all customers $\Psi_1, \ldots, \Psi_i$. Trips of this partial solution are represented by the arcs in the corresponding partial path.

Each label $\mathcal{L}$ has $M+3$ fields. Each field j, with $j = 1, \ldots, M$ records the availability time $T_j(\mathcal{L})$ of vehicle j, namely, the time the vehicle is available at the depot for starting the next service trip. Availability times are stored in decreasing order to better take advantage of the dominance rules introduced in the following. The $(M+1)^{\text{th}}$ field memorizes the total load infeasibility, and the $(M+2)^{\text{th}}$ the predecessor node. The last field stores the cost $c(\mathcal{L})$ of the partial solution represented by $\mathcal{L}$. When extending a label, M new labels are constructed, one for each possible allocation of the new trip to a vehicle. *Extending* a label $\mathcal{L}_i$ associated with node i through arc (i, j) means assigning trip $(0, \Psi_{i+1}, \ldots, \Psi_j, 0)$ to a certain vehicle. Identical labels associated with the same node are discarded.

Graph H is implicitly generated. Arc and solution costs are computed using relations introduced in Section 5.4.3. In particular, the cost of an arc $(i, \ldots, j+1)$ is calculated concatenating path $(0, \Psi_{i+1}, \ldots, \Psi_j)$ with path $(\Psi_{j+1}, 0)$, while label costs are updated using Relations (5.11)–(5.14) since vehicle availability times are stored in labels.

The label $\mathcal{L}$ with minimum cost $c(\mathcal{L})$ associated with node N is selected and the related solution is constructed (going backwards through the graph).

To speed up the procedure, dominated labels, accordingly with the following *dominance rule*, are discarded.

Dominance Rule 1 (Strong)**.** *Let $\mathcal{L}^1$ and $\mathcal{L}^2$ be two labels associated with the same node i. $\mathcal{L}^1$* strongly *dominates $\mathcal{L}^2$ if and only if*

$$c(\mathcal{L}^1) + \theta \sum_{j=1}^{M} \delta_j(\mathcal{L}^1, \mathcal{L}^2) \leq c(\mathcal{L}^2); \tag{5.15}$$

$$\delta_j(\mathcal{L}^1, \mathcal{L}^2) = \max\{0, T_j(\mathcal{L}^1) - T_j(\mathcal{L}^2)\}.$$

Roughly speaking, $\delta_j(\mathcal{L}^1, \mathcal{L}^2)$ represents the maximal additional penalization that can be introduced in the partial solution represented by $\mathcal{L}^1$ compared to the one represented by $\mathcal{L}^2$. If Inequality (5.15) holds, $\mathcal{L}^2$ cannot be extended in a better way than $\mathcal{L}^1$, and it is eliminated.

Table 5.1 shows all the labels generated by *AdSplit* when applied on the graph H depicted in Figure 5.2. The number of vehicles M is set to 2. Column *dom* indicates if the respective label is dominated in the sense of Dominance Rule 1. Only 3 labels associated with node 5 are non-dominated.

Label $(120, 110)$ with cost $c = 265$ associated with node 5 corresponds to the solution made up of three trips: $(0, 1, 0)$ assigned to one vehicle and trips $(0, 2, 0)$ and $(0, 3, 4, 5, 0)$

1				2				3				4				5			
cost	label	p	dom	cost	label	p	dom	cost	label	p	dom	cost	label	p	dom	cost	label	p	dom
10	(110,000)	0	no	40	(110, 70)	1	no	125	(110, 100)	1	no	190	(115, 110)	2	no	265	(120, 110)	2	no
				140	(95, 0)	0	no	180	(95, 75)	2	no	210	(115, 95)	3	no	285	(120, 95)	3	no
								230	(100, 70)	2	yes	250	(115, 75)	3	yes	300	(120, 115)	4	yes
								275	(100, 0)	0	no	250	(115, 100)	3	yes	300	(120, 100)	3	yes
												270	(115, 70)	2	yes	325	(120, 75)	3	yes
												295	(105, 100)	3	yes	345	(120, 70)	2	yes
												340	(115, 0)	2	no	370	(115, 70)	4	yes
												360	(105, 100)	3	yes	395	(120, 105)	4	yes
																415	(120, 0)	2	no

Table 5.1: Labels generated by *AdSplit*

performed by the other vehicle (in this order). Travelled distance is 125, while late arrival at customers $3, 4$ and 5 introduces a total TW violation of $70 \cdot \theta$.

Preliminary tests showed that a huge number of labels needs to be treated, which does not appear to be viable in the heuristic context (see Section 5.5.3 for details). For this reason a heuristic version of the Dominance Rule 1 is introduced as follows:

Dominance Rule 2 (Weak). *Label* $\mathcal{L}^1$ weakly *dominates* $\mathcal{L}^2$ *if and only if*

$$c(\mathcal{L}^1) + \theta \sum_{j=1}^{M} \delta_j(\mathcal{L}^1, \mathcal{L}^2) \leq \gamma c(\mathcal{L}^2), \tag{5.16}$$

$$c(\mathcal{L}^1) \leq c(\mathcal{L}^2). \tag{5.17}$$

where $\gamma \geq 1$.

For $\gamma = 1$, the weak dominance rule is equivalent to the strong version. When $\gamma > 1$, Inequality (5.16) is easier to be satisfied and a larger number of labels can be eliminated. Condition (5.17) is added because when $\gamma > 1$ label $\mathcal{L}^2$ can be dominated by label $\mathcal{L}^1$ even if $c(\mathcal{L}^2) < c(\mathcal{L}^1)$. Using the weak relation one expects the solution to be obtained quicker. On the other side the best decomposition of the chromosome can be missed.

The value of γ is dynamically adapted during the process according to the number of labels associated with each node. Precisely the following scheme is adopted:

$$\gamma = \begin{cases} \gamma + \frac{|\mathcal{L}_i|}{1000\mathcal{L}_{\text{threshold}}} & \text{if } |\mathcal{L}_i| > \mathcal{L}_{\text{threshold}} \\ \gamma - \frac{\mathcal{L}_{\text{threshold}}}{1000|\mathcal{L}_i|} & \text{if } |\mathcal{L}_i| < \mathcal{L}_{\text{threshold}} \end{cases} \tag{5.18}$$

where $|\mathcal{L}_i|$ is the number of labels associated with node i and $\mathcal{L}_{\text{threshold}}$ is a threshold parameter that indicates the number of labels that should be kept associated with each node. If after Relation (5.18) is applied, γ is lower than 1, it is set back to 1.

It is worth noting that the smaller $\mathcal{L}_{\text{threshold}}$ is, the quicker *AdSplit* is. On the other side, the larger $\mathcal{L}_{\text{threshold}}$ is, the better the quality of the solution obtained is.

5.4.5 Survivor strategy

When the population Π contains $\pi + \mu$ chromosomes, the *survivor* procedure is launched. It selects μ chromosomes based on their quality and their diversification contribution to the population as suggested by Vidal et al. [214, 213]. Diversity contribution $f(\Psi)$ is set as the average distance between a selected chromosome Ψ and its n_c closest neighbours in Π (forming set N_c). Distance $D(\Psi, \Psi_1)$ between chromosomes Ψ and Ψ_1 is measured by the *broken pair* distance that is the number of pairs of adjacent customers in Ψ that are broken in Ψ_1 (Prins [167]). Formally we have:

$$f(\Psi) = \frac{1}{n_c} \sum_{\Psi_1 \in N_c} D(\Psi, \Psi_1), \tag{5.19}$$

A biased fitness $bF(\cdot)$ is calculated for each chromosome as follows:

$$bF(\Psi) = r_F(\Psi) + (1 - \frac{n_e}{|\Pi|}) r_f(\Psi); \tag{5.20}$$

where $r_F(\Psi)$ and $r_f(\Psi)$ are the ranks of chromosome Ψ calculated based on fitness F and function f defined in Equation (5.19) respectively, and n_e is a parameter that ensures elitism properties during selection (see Vidal et al. [213] for a formal proof).

5.5 Computational results

In this section we present the results obtained with our procedure $\mathcal{A}^{CAF}$. We start introducing a set of instances (Section 5.5.1) for benchmark purposes: due to the novelty of the problem, no instances are available in the literature. In Section 5.5.2 we present the values of the parameters involved in the $\mathcal{A}^{CAF}$, determined using a meta-tuning procedure. Results obtained on the new set of instances are presented in Section 5.5.4, while in Section 5.5.5 results obtained by $\mathcal{A}^{CAF}$ on instances proposed by Hernandez et al. [105] for the MTVRPTW are given. $\mathcal{A}^{CAF}$ is coded in C++ and compiled with Visual Studio 2010. All the experiments are run on a Intel Xeon W3550 3.07GHz with a RAM of 12 Gb.

5.5.1 Instances for the MTVRPTWR

In this section we describe how we generate a set of instances for the MTVRPTWR. We use the instances introduced by Solomon [196] for the VRPTW as base instances and adapt them to the MTVRPTWR case. In Solomon [196] six groups of instances are generated, named R1, C1, RC1, R2, C2, RC2. Groups R1 and R2 have customers randomly located in a region, while they are clustered in groups C1 and C2. RC1 and RC2 instances contain a mix of randomly located and clustered customers. The time horizon is shorter in instances of groups C1, R1, RC1 than in instances of C2, R2, RC2. There are 56 instances in total.

Depot location as well as customer locations, demands, time windows and service times are set as in the original Solomon's instances. Release dates are calculated in three steps, based on the Algorithm 5 explained in the following, and given a rigidity parameter r.

In the first step (lines 1–7 of Algorithm 5) release dates $R^{(2)}$ are calculated based on Equation (5.21)

$$r = 1 - \frac{L_i - R_i^{(2)}}{T_H}. \tag{5.21}$$

Equation (5.21) (instead of Equation (5.1)) allows to univocally determine the value of $R_i^{(2)}$ once r is given. This is done in order to make instance replication easier. Inverting Equation (5.21) we obtain

$$R_i^{(2)} = L_i + T_H(r-1);$$

from which follows that

$$R_i^{(2)} \geq 0 \Leftrightarrow L_i + T_H(r-1) \geq 0 \Leftrightarrow r \geq 1 - \frac{L_i}{T_H}.$$

Then, when $r < 1 - \frac{L_i}{T_H}$, $R_i^{(2)}$ is negative. We fix the release date to zero in this case. Moreover, the decimal parts are truncated in order to work with integer values. Thus, the

Algorithm 5 Instance creation

for all $i = 1, \dots, N$ **do**
    if $L_i + T_H(r-1) \geq 0$ **then**
        $R_i = \lfloor L_i + T_H(r-1) \rfloor$
    else
        $R_i = 0$
    end if
end for
for all $i = 1, \dots, N$ **do**
    if $R_i + T_{0i} > L_i$ **then**
        $R_i = 0$
    end if
end for
Initialize Λ with customers with strictly positive release date and ranked with respect to non-decreasing release dates: $u < v \Rightarrow R_u \leq R_v$
$\Gamma = \emptyset; \quad L_\Gamma - T_\Gamma = \infty; \quad R_\Gamma = 0$
while $\Lambda \neq \emptyset$ **do**
    Get v^* first customer in Λ
    $\Lambda = \Lambda \setminus \{v^*\}$
    if $\Gamma = \emptyset$ **then**
        $\Gamma = \{v^*\}; \quad L_\Gamma - T_\Gamma = \kappa L_{v^*} - T_{0v^*}; \quad R_\Gamma = R_{v^*}$
    else
        $L_\Gamma - T_\Gamma = \min\{\kappa L_{v^*} - T_{0v^*}, L_\Gamma - T_\Gamma\}$
        if $R_{v^*} \leq L_\Gamma - T_\Gamma$ **then**
            $\Gamma = \Gamma \cup v^*; \quad R_\Gamma = R_{v^*}$
        else
            $\lambda = \Lambda \cup \{v^*\}$
            for all $v \in \Gamma$ **do**
                $R_v = R_\Gamma$
            end for
                $\Gamma = \emptyset$
        end if
    end if
end while

following scheme is adopted

$$R_i^{(2)} = \begin{cases} \lfloor L_i + T_H(r-1) \rfloor & \text{if } 1 - \frac{L_i}{T_H} \le r; \\ 0 & \text{otherwise.} \end{cases}$$

In the second step (lines 8–12 of Algorithm 5), to guarantee that each customer can be served by a round trip, $R_i^{(1)}$ is calculated as follows

$$R_i^{(1)} = \begin{cases} R_i^{(2)} & \text{if } R_i^{(2)} + T_{0i} \le L_i, \\ 0 & \text{otherwise.} \end{cases}$$

In the third step (lines 13–32 of Algorithm 5) the final release date values R_i, $i = 1, \dots, N$ are determined. Customers are *clustered* with respect to the release dates with the purpose to represent different truck arrivals at the depot. All the customers that belong to the same cluster Γ will be associated with the same release date R_Γ.

A list Λ is initialized with all the customers ordered by non-decreasing values of $R^{(1)}$, i.e., such that $i < j$ implies $R_i^{(1)} \le R_j^{(1)}$. Clusters Γ are constructed starting with a single customer v^1 and successively adding the following customers in Λ. A customer v^2 is added if and only if each customer $v \in \Gamma$ can be served by a round trip even if the merchandise is available at the depot at time $R_{v^2}^{(1)} (\ge R_v^{(1)})$, namely, if and only if

$$R_{v^2}^{(1)} + T_{0v} \le L_v \quad \forall v \in \Gamma. \tag{5.22}$$

The final release date R_v of each customer v in Γ is set to $R_{v^2}^{(1)}$. When the next customer v^2 to be inserted in Γ does not satisfy Relation (5.22), a new cluster is initialized and the procedure restarted.

In order to create different classes of instances, a parameter $\kappa \le 1$ is introduced in Equation (5.22). Hence, it becomes

$$R_{v^2}^{(1)} + T_{0v} \le \kappa L_v \quad \forall v \in \Gamma.$$

Different values of κ produce instances with different rigidity. In particular the higher is κ, the higher is the rigidity r.

The first step release dates $R_i^{(2)}$, $i = 1, \dots, N$ are determined setting $r = 0.5$. Then, for each Solomon instance, three instances are created using Algorithm 5 with values $\kappa = 0.25, 0.5, 0.75$. A fourth instance has all the release dates equal to zero and will be referred by $\kappa = 0$ for simplicity. There are in total 224 new instances.

Finally, vehicle capacities are half of the original values, while the number of vehicles M is set in order to have feasible or quasi-feasible solutions for the $\kappa = 0.75$ instance group. Since we force some release dates to be zero and we use different values of κ, the final rigidity of instances differs from 0.5. Values of the number of vehicles, actual rigidity r and tightness *tight* are reported in Table 5.2.

5.5.2 Tuning

The procedure makes use of some parameters that need to be set to values chosen into sensible ranges. After conducting preliminary tests we decided to fix the values of ζ_{ref}^+ and

		$\kappa = 0$		$\kappa = 0.25$		$\kappa = 0.5$		$\kappa = 0.75$	
	M	*tight*	r	*tight*	r	*tight*	r	*tight*	r
C101	12	0	0.606	0.085	0.664	0.129	0.690	0.161	0.709
C102	10	0	0.493	0.164	0.629	0.201	0.651	0.225	0.666
C103	11	0	0.350	0.261	0.576	0.290	0.595	0.311	0.609
C104	13	0	0.227	0.350	0.540	0.361	0.548	0.380	0.561
C105	11	0	0.579	0.098	0.647	0.152	0.678	0.190	0.702
C106	11	0	0.563	0.105	0.638	0.152	0.667	0.203	0.697
C107	11	0	0.553	0.102	0.627	0.151	0.656	0.206	0.689
C108	10	0	0.522	0.116	0.607	0.158	0.632	0.225	0.674
C109	10	0	0.467	0.139	0.573	0.192	0.605	0.257	0.644
R101	22	0	0.537	0.093	0.605	0.099	0.609	0.135	0.632
R102	19	0	0.424	0.179	0.571	0.185	0.574	0.222	0.601
R103	18	0	0.332	0.258	0.548	0.261	0.550	0.311	0.589
R104	17	0	0.241	0.332	0.522	0.334	0.523	0.387	0.566
R105	17	0	0.492	0.111	0.575	0.122	0.582	0.181	0.619
R106	15	0	0.391	0.192	0.548	0.201	0.553	0.260	0.594
R107	16	0	0.311	0.266	0.533	0.272	0.536	0.346	0.590
R108	16	0	0.231	0.335	0.514	0.336	0.515	0.382	0.550
R109	15	0	0.427	0.137	0.531	0.147	0.537	0.212	0.577
R110	15	0	0.362	0.204	0.517	0.223	0.527	0.299	0.575
R111	15	0	0.328	0.241	0.521	0.249	0.526	0.320	0.575
R112	18	0	0.283	0.289	0.500	0.314	0.515	0.469	0.618
RC101	19	0	0.492	0.104	0.569	0.120	0.577	0.175	0.611
RC102	17	0	0.406	0.179	0.546	0.187	0.551	0.244	0.587
RC103	18	0	0.332	0.249	0.532	0.257	0.537	0.325	0.584
RC104	19	0	0.256	0.317	0.514	0.317	0.515	0.391	0.570
RC105	18	0	0.430	0.153	0.542	0.167	0.550	0.220	0.584
RC106	16	0	0.427	0.141	0.531	0.156	0.540	0.227	0.584
RC107	18	0	0.365	0.202	0.514	0.212	0.520	0.280	0.563
RC108	18	0	0.302	0.268	0.501	0.280	0.508	0.380	0.571
C201	3	0	0.519	0.151	0.634	0.199	0.664	0.258	0.702
C202	4	0	0.393	0.239	0.600	0.274	0.623	0.318	0.651
C203	5	0	0.277	0.318	0.570	0.339	0.584	0.364	0.601
C204	5	0	0.147	0.405	0.530	0.416	0.538	0.431	0.548
C205	3	0	0.495	0.159	0.619	0.210	0.652	0.262	0.685
C206	3	0	0.469	0.175	0.607	0.225	0.639	0.287	0.679
C207	3	0	0.452	0.185	0.603	0.233	0.634	0.283	0.666
C208	3	0	0.445	0.183	0.592	0.237	0.626	0.290	0.660
R201	4	0	0.493	0.133	0.600	0.162	0.617	0.215	0.651
R202	2	0	0.365	0.222	0.567	0.247	0.581	0.284	0.606
R203	3	0	0.252	0.312	0.546	0.328	0.555	0.355	0.575
R204	3	0	0.141	0.395	0.519	0.397	0.521	0.420	0.540
R205	3	0	0.430	0.177	0.571	0.236	0.607	0.297	0.645
R206	3	0	0.318	0.262	0.548	0.313	0.578	0.365	0.612
R207	3	0	0.222	0.337	0.533	0.368	0.551	0.405	0.578
R208	3	0	0.126	0.409	0.514	0.410	0.514	0.456	0.548
R209	3	0	0.370	0.211	0.537	0.283	0.580	0.380	0.640
R210	3	0	0.334	0.250	0.540	0.310	0.578	0.381	0.623
R211	4	0	0.303	0.276	0.517	0.390	0.582	0.541	0.673
RC201	4	0	0.488	0.142	0.600	0.183	0.624	0.240	0.660
RC202	5	0	0.366	0.232	0.569	0.264	0.588	0.314	0.621
RC203	3	0	0.257	0.320	0.549	0.344	0.563	0.385	0.594
RC204	4	0	0.147	0.393	0.519	0.394	0.519	0.440	0.554
RC205	4	0	0.426	0.176	0.566	0.254	0.610	0.334	0.658
RC206	3	0	0.424	0.172	0.562	0.227	0.594	0.293	0.638
RC207	4	0	0.362	0.212	0.531	0.282	0.571	0.388	0.638
RC208	4	0	0.291	0.276	0.508	0.379	0.568	0.537	0.666

Table 5.2: Instance details

ζ^{-}_{ref} to 0.35 and 0.25 respectively, while θ and λ are initially set respectively to 20 and 2.

In order to determine the values of the remaining parameters, we run the Evolutionary Strategy with Covariance Matrix Adaptation proposed by Hansen and Ostermeier [101] on a limited set of instances. In particular $\mathcal{A}^{CAF}$ is run on C108, R104, RC106, RC208, obtained with $\kappa = 0.75$, and we obtained the values reported in Table 5.3.

	Parameter	Range	Final value
π	Dimension of population	$[1, 100]$	20
μ	Children generated at each generation	$[1, 100]$	30
n^e	Proportion of elite individuals $n_e = n^e \times \Pi$ (Eq. 5.19)	$[0.1, 1]$	0.20
n^c	Proportion of close individuals $n_c = n^c \times \Pi$ (Eq. 5.20)	$[0.1, 1]$	0.35

Table 5.3: Parameter Tuning

5.5.3 Setting of $\mathcal{L}_{\text{threshold}}$

The value of $\mathcal{L}_{\text{threshold}}$ is important to achieve the best compromise between solution quality and computational efficiency. To find a suitable value, we evaluate the impact of $\mathcal{L}_{\text{threshold}}$ on a set of 100 chromosomes. In order to avoid completely random chromosomes, we proceed as follows. A chromosome is randomly generated for instances C101 and C201 of group $\kappa = 0$. *AdSplit* first evaluates them with $\mathcal{L}_{\text{threshold}} = 10$, then they are improved by LS. The resulting chromosomes are re-evaluated by *AdSplit* with different values of $\mathcal{L}_{\text{threshold}}$. Average results obtained on the 100 evaluations are reported in Table 5.4. The value of $\mathcal{L}_{\text{threshold}}$ is indicated

	C101 $\kappa = 0$				C201 $\kappa = 0$			
$\mathcal{L}_{\text{threshold}}$	time (ms)	cost	time gap (%)	cost gap (%)	time (ms)	cost	time gap (%)	cost gap (%)
500	292111	2561.17	-	-	119933	2089.89	-	-
50	5446	2564.93	-98.14	0.15	2797	2089.93	-97.67	≈ 0
45	4299	2565.04	-98.53	0.15	2413	2089.93	-97.99	≈ 0
40	3134	2565.32	-98.93	0.16	2080	2089.93	-98.27	≈ 0
35	2273	2565.42	-99.22	0.17	1698	2089.93	-98.58	≈ 0
30	1550	2565.82	-99.47	0.18	1377	2089.93	-98.85	≈ 0
25	1013	2594.49	-99.65	1.30	1035	2089.93	-99.14	≈ 0
20	631	2656.85	-99.78	3.74	657	2089.93	-99.45	≈ 0
15	346	2713.37	-99.88	5.94	370	2089.93	-99.69	≈ 0
10	164	2788.42	-99.94	8.87	178	2089.93	-99.85	≈ 0
5	62	3185.31	-99.98	24.37	67	2089.93	-99.94	≈ 0
1	14	8619.91	-100.00	236.56	11	2242.80	-99.99	7.32

Table 5.4: Setting $\mathcal{L}_{\text{threshold}}$

in the first column of the table. A maximum of 500 labels is considered for $\mathcal{L}_{\text{threshold}}$. In this case, computational times are huge implying also that splitting chromosomes using the Strong Dominance Rule 1 is not time efficient. The Weak Dominance Rule 2 allows a quick evaluation preserving solution quality even with a few labels kept associated with each node. Symbol ≈ 0 means the value is approximately zero. Results show that for instance C201 a small cost deterioration is achieved when the value of $\mathcal{L}_{\text{threshold}}$ is very small. This can be explained by the fact that C201 is characterized by a low number of vehicles, that reduces the possible assignment of trips. Finally, it has been decided to set $\mathcal{L}_{\text{threshold}} = 15$.

5.5.4 Results on MTVRPTWR instances

$\mathcal{A}^{CAF}$ is run 5 times over the 224 instances. Each run is stopped after 5 minutes of computation time. Complete results are reported in Tables 5.5–5.6, each table being divided in two parts, exhibiting results for instances with $\kappa = 0, 0.25, 0.5, 0.75$. Column *instance* contains the name of the original Solomon instance. Columns *best* report the distance (*dist*) and the number of trips (*#trips*) of the best solution found on the 5 runs. Columns *average* report average distances and trips on the five found solutions. Column *#feas* indicates the number of runs the procedure found a feasible solution. A dash means no feasible solution has been found for the respective instance. It can be noticed that the best solution can be formed by a number of trips higher than the average.

Result analysis is reported on Table 5.7 and in Figure 5.3. Table 5.7 reports average results per group of instances. Columns *best* report average distance and number of trips of the corresponding best solution found by the procedure, while columns *average* report the average of the average values on instances of the same group. Column *% feas* indicates the percentage of feasible solutions found on the total number of runs (that is 5 times the number of instances forming a specified group).

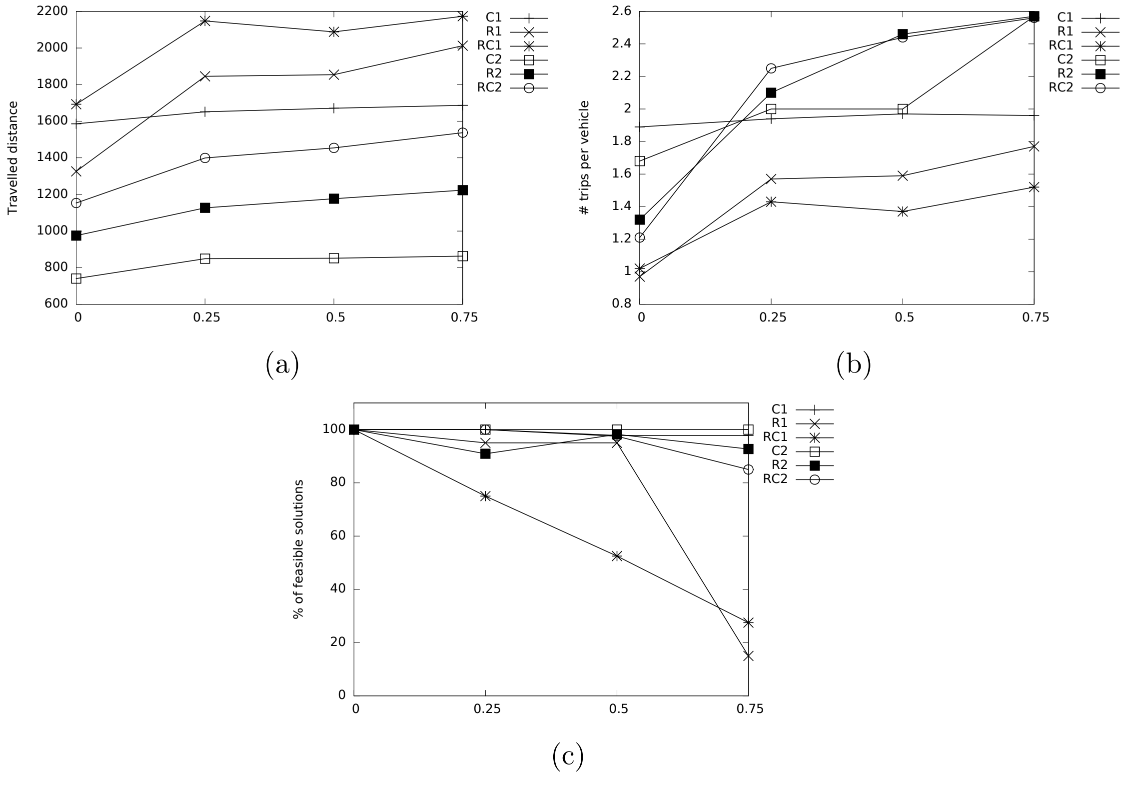


Figure 5.3: Result analysis

Procedure $\mathcal{A}^{CAF}$ always finds a feasible solution for instances with $\kappa = 0$ (Table 5.5 and 5.7). This validates the generation scheme presented in Section 5.5.1.

Rigidity and tightness of instances describe on average the difficulty of solving an instance: considering results grouped by type of instances and value of κ, it can be noticed (Figures 5.3a–5.3c, Table 5.7) that when the tightness and rigidity grow, the travelled distance and the number of trips per vehicle grow, while the number of feasible solutions found

instance	$\kappa = 0$					$\kappa = 0.25$				
	best		average		#feas	best		average		#feas
	dist	#trips	dist	#trips		dist	#trips	dist	#trips	
C101	1529.66	20	1568.27	20.8	5	1565.24	21	1585.29	21.0	5
C102	1675.75	22	1715.14	21.8	5	1759.23	22	1778.12	22.4	5
C103	1452.69	19	1524.98	19.6	5	1748.96	22	1810.23	22.6	5
C104	1384.78	19	1407.10	19.2	5	1774.75	22	1834.28	22.6	5
C105	1550.02	21	1709.18	22.2	5	1552.38	21	1601.48	21.4	5
C106	1592.13	20	1624.38	21.4	5	1594.68	21	1621.70	21.6	5
C107	1513.19	20	1527.41	20.0	5	1511.36	20	1536.45	20.0	5
C108	1545.94	21	1659.66	21.6	5	1518.79	20	1548.04	20.2	5
C109	1496.65	21	1538.70	20.2	5	1536.90	21	1547.45	20.4	5
R101	1671.77	22	1678.46	20.6	5	2102.53	32	2160.65	31.4	5
R102	1498.23	18	1502.58	18.0	5	2133.22	31	2146.89	32.5	2
R103	1288.44	16	1298.49	16.0	5	1924.76	27	1990.82	29.0	5
R104	1177.88	15	1190.59	15.0	5	1632.43	24	1711.13	25.6	5
R105	1421.19	16	1433.05	16.4	5	1848.76	25	1913.50	26.6	5
R106	1361.02	16	1365.95	16.2	5	1882.07	27	1959.79	28.4	5
R107	1235.15	16	1245.09	16.0	5	1830.08	26	1861.37	27.2	5
R108	1187.36	15	1193.28	15.0	5	1565.98	22	1622.82	23.0	5
R109	1307.25	17	1316.19	16.4	5	1750.73	25	1835.34	26.0	5
R110	1246.99	15	1253.13	15.0	5	1741.20	25	1779.37	26.6	5
R111	1236.23	17	1244.99	16.2	5	1803.39	26	1833.50	26.4	5
R112	1182.72	15	1191.47	15.6	5	1323.48	17	1329.28	17.0	5
RC101	1805.40	19	1828.30	19.0	5	2304.70	26	2398.29	28.6	5
RC102	1746.02	18	1759.63	18.2	5	-	-	-	-	0
RC103	1637.38	18	1641.92	18.0	5	2161.58	25	2319.11	28.0	5
RC104	1582.81	18	1583.46	18.0	5	1884.44	22	1963.69	22.8	5
RC105	1752.66	19	1759.14	18.4	5	2291.55	28	2367.93	29.0	2
RC106	1750.52	19	1764.37	18.8	5	2249.39	29	2266.68	28.0	3
RC107	1615.05	18	1618.37	18.0	5	1911.32	21	1980.79	22.8	5
RC108	1581.78	18	1587.03	18.0	5	1706.06	20	1737.62	19.4	5
C201	777.48	6	777.48	6.0	5	781.76	7	781.76	7.0	5
C202	718.69	6	724.85	6.0	5	913.97	7	914.23	7.0	5
C203	700.20	6	711.06	6.0	5	949.71	8	949.71	8.0	5
C204	695.12	6	698.17	6.0	5	966.98	7	977.18	7.2	5
C205	767.55	7	770.21	6.8	5	755.45	7	755.45	7.0	5
C206	747.14	6	750.42	6.0	5	796.57	7	797.31	7.0	5
C207	746.62	6	748.66	6.0	5	786.64	7	788.08	7.0	5
C208	741.58	6	742.09	6.0	5	820.57	8	828.71	7.8	5
R201	1272.47	4	1287.21	4.2	5	1403.33	8	1444.78	9.0	5
R202	1272.72	4	1278.20	4.2	5	1400.45	6	1452.05	6.6	5
R203	966.35	5	976.30	4.2	5	1140.24	6	1162.86	6.0	5
R204	779.22	3	787.31	3.6	5	1018.57	5	1027.23	5.8	5
R205	1074.75	5	1089.24	4.6	5	1141.30	8	1163.15	7.2	5
R206	944.58	4	962.93	4.2	5	1018.97	6	1034.26	6.0	5
R207	849.64	4	862.07	4.0	5	981.61	6	993.27	6.0	5
R208	735.49	4	738.52	3.8	5	905.53	4	912.03	4.6	5
R209	944.06	3	961.83	3.6	5	1050.79	6	1127.31	6.8	5
R210	985.66	4	1001.98	4.2	5	1149.92	7	1178.50	7.0	5
R211	772.99	5	779.80	4.2	5	891.09	6	900.95	6.6	5
RC201	1424.18	5	1459.26	5.2	5	1637.82	9	1690.93	9.8	5
RC202	1171.86	4	1196.66	4.4	5	1423.40	8	1508.23	9.4	5
RC203	1108.21	4	1150.94	5.0	5	1441.46	6	1485.27	8.2	5
RC204	806.44	4	809.59	4.2	5	1076.24	7	1082.30	6.8	5
RC205	1321.64	4	1354.18	4.6	5	1543.24	11	1585.42	10.4	5
RC206	1325.01	5	1385.87	5.4	5	1493.13	9	1554.81	9.8	5
RC207	1042.03	4	1050.24	4.6	5	1185.78	8	1228.64	8.2	5
RC208	803.59	4	818.23	4.0	5	1001.06	6	1060.34	7.0	5

Table 5.5: Results on new instances, $\kappa = 0$ and $\kappa = 0.25$

instance	$\kappa = 0.5$					$\kappa = 0.75$				
	best		average		#feas	best		average		#feas
	dist	#trips	dist	#trips		dist	#trips	dist	#trips	
C101	1579.19	22	1584.99	21.8	5	1591.91	22	1611.34	22.2	5
C102	1746.59	22	1774.92	22.4	5	1766.15	22	1793.42	22.2	5
C103	1842.84	24	1889.73	23.6	5	1900.10	23	1920.27	24.5	4
C104	1773.04	21	1814.93	21.8	5	1805.55	20	1890.63	21.8	5
C105	1603.96	22	1637.48	22.0	4	1600.94	21	1633.32	21.4	5
C106	1587.25	22	1605.18	21.8	5	1663.38	22	1695.00	21.8	5
C107	1518.75	20	1529.53	20.0	5	1538.41	20	1547.13	20.0	5
C108	1556.50	20	1662.16	21.4	5	1546.53	20	1569.45	20.2	5
C109	1506.85	20	1541.79	20.0	5	1512.26	20	1521.20	20.2	5
R101	2135.93	32	2236.49	33.0	5	2372.83	35	2372.83	35.0	1
R102	2102.70	31	2120.94	32.0	2	-	-	-	-	0
R103	1879.98	27	1959.64	28.6	5	-	-	-	-	0
R104	1637.50	22	1680.34	23.8	5	-	-	-	-	0
R105	1964.79	27	1987.37	28.6	5	2046.53	31	2135.91	32.0	2
R106	1900.14	28	1964.97	28.6	5	2043.12	30	2043.12	30.0	1
R107	1853.74	27	1888.95	27.6	5	-	-	-	-	0
R108	1562.17	22	1618.08	23.2	5	-	-	-	-	0
R109	1730.47	24	1819.57	26.0	5	1898.65	28	1898.65	28.0	1
R110	1682.08	24	1756.21	25.8	5	-	-	-	-	0
R111	1784.27	26	1884.33	27.6	5	-	-	-	-	0
R112	1320.07	17	1332.57	17.4	5	1540.39	23	1612.44	24.8	4
RC101	2484.09	30	2484.09	30.0	1	-	-	-	-	0
RC102	-	-	-	-	0	-	-	-	-	0
RC103	2194.26	25	2270.77	26.8	5	-	-	-	-	0
RC104	1896.29	22	1930.99	22.6	5	2175.03	27	2214.98	27.3	4
RC105	-	-	-	-	0	-	-	-	-	0
RC106	-	-	-	-	0	-	-	-	-	0
RC107	1918.73	21	2024.90	24.0	5	2249.07	28	2290.20	28.6	5
RC108	1718.04	19	1728.73	19.4	5	1985.12	24	2014.37	25.5	2
C201	788.37	7	788.37	7.0	5	815.58	6	815.58	6.0	5
C202	913.66	7	914.14	7.0	5	913.66	7	915.58	7.0	5
C203	952.09	8	962.94	8.0	5	952.46	8	952.47	8.0	5
C204	967.23	7	975.28	7.0	5	976.79	7	982.89	7.0	5
C205	762.06	7	762.06	7.0	5	778.45	6	778.45	6.0	5
C206	796.57	7	797.32	7.0	5	813.52	6	813.52	6.0	5
C207	784.22	7	789.49	7.0	5	805.76	6	806.23	6.2	5
C208	817.35	8	824.63	8.0	5	833.46	8	841.61	7.6	5
R201	1443.84	10	1464.64	8.6	5	1430.19	9	1455.41	8.6	5
R202	1425.40	9	1452.19	8.3	4	1452.75	9	1481.26	8.3	3
R203	1214.24	7	1242.41	7.6	5	1255.53	8	1287.27	8.6	5
R204	990.54	6	1022.97	5.8	5	987.98	6	1028.77	6.0	5
R205	1183.48	8	1256.04	9.2	5	1242.05	9	1266.00	9.0	5
R206	1069.98	7	1113.78	8.2	5	1111.86	8	1160.79	8.2	5
R207	1004.76	6	1032.92	6.8	5	1034.82	7	1054.83	6.8	5
R208	905.90	4	920.82	4.6	5	910.47	5	944.81	5.6	5
R209	1188.91	9	1237.35	9.6	5	1320.07	9	1327.13	9.3	3
R210	1228.73	9	1288.66	8.4	5	1268.23	8	1338.17	9.2	5
R211	902.45	6	910.85	6.6	5	1074.15	8	1114.96	7.8	5
RC201	1677.63	10	1784.00	11.8	5	1796.39	10	1874.02	11.4	5
RC202	1427.31	10	1522.27	10.6	5	1539.80	13	1599.26	12.2	5
RC203	1464.58	8	1480.79	8.8	4	1488.91	9	1493.40	9.0	2
RC204	1084.93	7	1086.09	7.0	5	1103.65	7	1123.24	7.2	5
RC205	1695.25	11	1734.99	11.4	5	1777.28	12	1808.49	11.4	5
RC206	1445.43	9	1575.79	9.0	5	1493.88	9	1594.45	9.2	5
RC207	1163.13	8	1328.49	9.6	5	1449.89	11	1502.53	10.6	5
RC208	1107.69	8	1122.44	7.6	5	1297.58	9	1304.99	8.5	2

Table 5.6: Results on new instances, $\kappa = 0.5$ and $\kappa = 0.75$

Instance Group	best		average		% feas	tight	rigidity
	dist	#trips	dist	#trips			
	$\kappa = 0$						
C1	1526.76	20.33	1586.09	20.76	100	0.00	0.48
R1	1317.85	16.50	1326.11	16.37	100	0.00	0.36
RC1	1683.95	18.38	1692.78	18.30	100	0.00	0.38
C2	736.80	6.13	740.37	6.10	100	0.00	0.40
R2	963.45	4.09	975.04	4.07	100	0.00	0.30
RC2	1125.37	4.25	1153.12	4.68	100	0.00	0.35
	$\kappa = 0.25$						
C1	1618.03	21.11	1651.45	21.36	100	0.16	0.61
R1	1794.89	25.58	1845.37	26.64	95.0	0.22	0.54
RC1	2072.72	24.43	2147.73	25.51	75.0	0.20	0.53
C2	846.45	7.25	849.05	7.25	100	0.23	0.59
R2	1100.16	6.18	1126.94	6.51	100	0.27	0.54
RC2	1350.27	8.00	1399.49	8.70	100	0.21	0.53
	$\kappa = 0.5$						
C1	1635.00	21.44	1671.19	21.64	97.8	0.20	0.64
R1	1796.15	25.58	1854.12	26.85	95.0	0.23	0.55
RC1	2042.28	23.40	2087.90	24.56	52.5	0.21	0.54
C2	847.69	7.25	851.78	7.25	100	0.27	0.62
R2	1141.66	7.36	1176.60	7.60	98.2	0.31	0.57
RC2	1383.24	8.88	1454.36	9.47	97.5	0.28	0.58
	$\kappa = 0.75$						
C1	1658.36	21.11	1686.86	21.59	97.8	0.24	0.66
R1	1980.30	29.40	2012.59	29.95	15.0	0.29	0.59
RC1	2136.41	26.33	2173.18	27.12	27.5	0.28	0.58
C2	861.21	6.75	863.29	6.73	100	0.31	0.65
R2	1189.83	7.82	1223.58	7.95	92.7	0.37	0.61
RC2	1493.42	10.00	1537.55	9.94	85.0	0.34	0.62

Table 5.7: Statistics on new instances

by the algorithm decreases. On the other side, there is no evident punctual correlation between rigidity and tightness, and the results obtained on a specific instance: for example instance R103 with $\kappa = 0.25$ has higher tightness and rigidity than instance RC102 (see Table 5.2), but 5 feasible solutions out of 5 runs are found for the former, while none for the latter.

Robustness of procedure $\mathcal{A}^{CAF}$ is proved by the small differences between best values and average values reported in Tables 5.5–5.6 and in Table 5.7.

5.5.5 Comparison with Hernandez et al. [105]

To evaluate the performance of $\mathcal{A}^{CAF}$, we run the procedure on instances generated by Hernandez et al. [105] for the MTVRP with TW. These instances are generated from Solomon's instances in groups C2, R2, RC2, considering the first 25 customers and M fixed to 2, and the first 50 customers and $M = 4$. Vehicle capacity is fixed to 100, loading time at the depot is trip dependent and in particular it is 0.2 times the sum of service times at customers in the trip. Travel times are the Euclidean distances rounded to the first decimal. Limitation into the number of customers is due to the exact nature of the algorithm proposed by Hernandez et al. [105]. Due to the heuristic nature of our algorithm, we consider as well the instances with all the 100 customers. Following the instance generation system of Hernandez et al. [105], we double the number of available vehicles used for instances with 50 customers. Then, 8 vehicles are available to serve the 100 customers.

Instances in groups C1, R1 and RC1 are not considered by Hernandez et al. [105] due to short time horizon that, in their opinion, would not allow vehicles to perform different trips.

$\mathcal{A}^{CAF}$ is run five times on each instance and it is stopped after 1 minute on instances with 25 customers and after 5 minutes on instances with 50 customers and with 100 customers. Results are reported in Tables 5.8–5.10.

The first column reports the instance name, columns *HRN* report the optimal value (column *opt*) found by Hernandez et al. [105]. A blank indicates they could not find the optimal solution. In some cases their algorithm provides a feasible solution which value is indicated in column *feas*.

Columns *best* report the travelled distance (*dist*) and the number of trips (*# trips*) that characterize the best solution found by $\mathcal{A}^{CAF}$in the five runs. Columns *average* indicate average values over the five runs. Bold numbers indicate the best known solution has been improved by $\mathcal{A}^{CAF}$ (we omitted the bold font when no solution value was available). Finally, column *# opt* reports the number of runs $\mathcal{A}^{CAF}$ finds the optimal solution on the five runs. A dash is reported when the optimal value is not available.

It can be observed that on small instances $\mathcal{A}^{CAF}$ finds the optimal solution on all the five runs for 23 out of 25 instances. It fails to find the optimal solution on only 2 runs in total, one for instance C201 and one for instance RC206. The average gap from the optimal value is respectively 0.079% and 0.003%. Moreover, a new best solution is obtained for instance RC204. On instances with 50 customers, our procedure fails to find the optimal solution only for instance RC202, while in the other four cases the optimal solution is retrieved 14 times out of 20 runs. 8 instances on the 9 with a feasible known solution are improved,

Instance	HRN		best		average		#opt
	opt	feas	dist	# trips	dist	# trips	
C201	380.8		380.8	3	381.10	5.4	4
C202	368.6		368.6	5	368.60	5.0	5
C203	361.7		361.7	5	361.70	5.0	5
C204	358.8		358.8	5	358.80	5.0	5
C205	377.2		377.2	5	377.20	5.0	5
C206	367.2		367.2	5	367.20	5.0	5
C207	359.1		359.1	5	359.10	5.0	5
C208	360.9		360.9	5	360.90	5.0	5
R201	554.6		554.6	4	554.60	4.0	5
R202	485.0		485.0	4	485.00	4.0	5
R203	444.2		444.2	4	444.20	4.0	5
R204	407.5		407.5	4	407.50	4.0	5
R205	448.4		448.4	4	448.40	4.0	5
R206	413.9		413.9	4	413.90	4.0	5
R207	400.1		400.1	4	400.10	4.0	5
R208	394.3		394.3	4	394.30	4.0	5
R209	418.3		418.3	4	418.30	4.0	5
R210	448.3		448.3	4	448.30	4.0	5
R211	400.1		400.1	4	400.10	4.0	5
RC201	660.0		660.0	6	660.00	6.0	5
RC202	596.8		596.8	6	596.80	6.0	5
RC203	530.1		530.1	6	530.10	6.0	5
RC204		520.3	**518.0**	6	518.00	6.0	-
RC205	605.3		605.3	6	605.30	6.0	5
RC206	575.1		575.1	6	575.12	6.0	4
RC207	528.2		528.2	6	528.20	6.0	5
RC208		506.4	506.4	6	506.40	6.0	-

Table 5.8: Results on Hernandez et al. [105] instances with $N = 25$ and $M = 2$

Instance	HRN		best		average		#opt
	opt	feas	dist	# trips	dist	# trips	
C201		717.9	**714.2**	10	714.20	10.0	-
C202		701.9	**700.1**	9	700.38	9.0	-
C203			688.0	9	689.34	9.0	-
C204			685.1	9	685.10	9.0	-
C205		706.6	**700.0**	9	703.52	9.8	-
C206			694.6	9	696.92	9.2	-
C207			689.7	9	690.38	9.0	-
C208			688.6	9	688.60	9.0	-
R201	909.8		909.8	9	917.08	9.0	1
R202	816.0		816.0	8	816.00	8.0	5
R203			742.4	8	743.40	8.0	-
R204			702.3	8	704.38	8.0	-
R205	807.3		807.3	8	808.74	8.0	3
R206		767.6	**758.2**	8	760.96	8.0	-
R207			715.7	8	715.70	8.0	-
R208			699.6	8	700.60	8.0	-
R209		749.6	**746.0**	8	746.00	8.0	-
R210			777.2	8	779.22	8.0	-
R211			717.4	8	722.02	8.0	-
RC201	1096.6		1096.6	10	1096.60	10.0	5
RC202	1001.6		1038.6	10	1038.60	10.0	0
RC203		945.8	**941.2**	10	941.20	10.0	-
RC204		915.9	915.9	10	915.90	10.0	-
RC205		1065.4	**1058.7**	10	1058.70	10.0	-
RC206			1027.4	11	1032.12	10.8	-
RC207		944.8	**941.7**	10	941.70	10.0	-
RC208			916.8	10	916.80	10.0	-

Table 5.9: Results on Hernandez et al. [105] instances with $N = 50$ and $M = 4$

Instance	best		average	
	dist	# trips	dist	# trips
C201	1488.9	19	1500.22	19.2
C202	1479.3	19	1486.94	19.0
C203	1467.3	19	1471.20	19.0
C204	1453.6	19	1455.46	19.0
C205	1477.1	19	1483.04	19.0
C206	1464.7	19	1473.38	19.0
C207	1464.2	19	1470.36	19.0
C208	1459.4	19	1465.86	19.0
R201	1449.7	16	1464.32	15.8
R202	1343.3	16	1352.70	15.8
R203	1222.2	15	1232.50	15.4
R204	1165.6	15	1172.54	15.0
R205	1292.2	15	1315.08	15.6
R206	1239.9	15	1249.76	15.4
R207	1194.3	15	1200.76	15.0
R208	1159.8	15	1164.56	15.0
R209	1234.5	16	1248.42	15.4
R210	1247.5	15	1253.24	15.6
R211	1170.5	15	1182.38	15.0
RC201	1843.6	18	1862.40	18.8
RC202	1733.9	18	1740.34	18.2
RC203	1618.6	18	1624.24	18.2
RC204	1579.1	18	1581.36	18.0
RC205	1759.8	18	1776.68	18.0
RC206	1731.1	18	1747.24	18.0
RC207	1656.7	18	1662.96	18.0
RC208	1580.9	18	1585.44	18.0

Table 5.10: Results on instances with $N = 100$ and $M = 8$ created as in Hernandez et al. [105]

while in the remaining case a same-cost solution is got. The average percentage gap from the optimal value is 0.03%. Feasible solutions are found for all the instances, included those with all 100 customers.

5.6 Conclusions and perspectives

In this paper we introduced a new problem, the Multi Trip Vehicle Routing Problem with Time Windows and Release Dates. It raises in city logistics context, where trucks deliver goods to city distribution centers (CDC) before they are delivered to final customers by eco-friendly vans. Optimization of van trips depends on the truck delivery plan to the CDC. Trucks arrive during the whole day, continuously bringing goods into the distribution system. Arrival of trucks to CDC is modelled associating a *release date* with each merchandise. It represents the moment the merchandise itself becomes available for final delivery.

We introduced a new set of instances on which we run the memetic algorithm we developed. Moreover, we run the algorithm on instances for the Multi Trip Vehicle Routing Problem with Time Windows for performance evaluation purposes. Results show the efficiency of our procedure.

An efficient labelling procedure is proposed to turn permutation of customers into solution that is an adaptation of the procedure proposed by Prins [166] for the VRP. It is designed for the MTVRPTWR case, but it can be used in the MTVRPTW context as well.

Associating a release date with each merchandise implicitly suppose the arrival of each truck to the depot is known in advance, at least before the operational planning is computed. Communication and organization between carriers and the management center is needed. Future studies could introduce some dynamism in the problem, considering part of the goods or the whole merchandise to arrive at the depot with no advanced notice. Optimization procedure needs to react to these events, reorganizing the planning quickly and efficiently.

Chapter 6

An Iterated Local Search for the Multi Commodity Multi Trip Vehicle Routing Problem with Time Windows

Abstract

The Multi Commodity Multi-Trip Vehicle Routing Problem with Time Windows calls for the determination of a routing planning to serve a set of customers that require products belonging to *incompatible* commodities. Two commodities are incompatible if they cannot be transported together into the same vehicle. Vehicles are allowed to perform several trips during the working day. The objective is to minimize the number of used vehicles.

We propose an Iterated Local Search that outperforms the previous algorithm designed for the problem. Moreover, we conduct an analysis on the benefit that can be obtained introducing the multi-trip aspect at the fleet dimensioning level. Results on classical VRPTW instances show that, in some cases, the fleet can be halved.

6.1 Introduction

The well known Vehicle Routing Problem (VRP) (Toth and Vigo [204] and Golden et al. [90]) is an $\mathcal{NP}$-hard combinatorial optimization problem where a set of geographically scattered customers has to be served by a fleet of vehicles. An implicit assumption of the VRP is that each vehicle can perform only one route in the planning horizon. Several practical situations allow vehicles to perform several trips during the working day. The problem that arises is the Multi Trip VRP (MTVRP), (Cattaruzza et al. [33], Olivera and Viera [156], Mingozzi et al. [140]).

This paper studies a variant of the MTVRP, where different commodities need to to be delivered to customers. Commodities are *incompatible*, i.e., they cannot be transported together into the same vehicle. On the other hand, the vehicles can transport different commodities in different trips.

The problem has been introduced by Battarra et al. [17] and called the Minimum Multiple Trip Vehicle Routing Problem (MMTVRP). The study was motivated by a real-world application, where a set of supermarkets requires deliveries of different commodities. To the best of our knowledge, no further study has been done on the MMTVRP.

Classical routing problems implicitly deal with different commodities. In particular, when vehicles can transport only one commodity, the problem is split in several sub-problems, one for each commodity, where a set of dedicated vehicles is available for the deliveries. On the other hand, if vehicles can carry different commodities at the same time, customer's demands of all commodity are collapsed into a single value, that represents its total request. Moreover, demands of different commodities are normalized into vehicle capacity units. As a consequence the resulting problem is a single-commodity VRP. Recently Archetti et al. [7] introduced a variant of the Split-Delivery VRP (SDVRP) where multiple customer visits are allowed only if the customer requires different commodities. In this case, commodities need to be considered explicitly.

In this paper we propose an effective Iterated Local Search (ILS) to solve the MMTVRP and a procedure *AdSplit* that turns permutations of customers into solutions. Moreover, a study is conducted on the benefit of introducing the multi-trip aspect in fleet dimensioning problems.

The paper is organized as follows. The problem and the notation are introduced in Section 6.2. Section 6.3 describes the ILS algorithm, while Section 6.4 presents the *AdSplit* procedure. Finally, Section 6.5 presents the results and Section 6.6 draws some conclusions.

6.2 Problem definition and notation

The problem we consider has been introduced by Battarra et al. [17] and arises in the distribution of merchandise to supermarkets. The main characteristics of the problem are the presence of time windows (TW) and the fact that goods belong to different commodities which cannot be transported in the same vehicle at the same time. Overtime is not allowed. Moreover, the number of used vehicles is a variable that needs to be minimized. This problem has been called by Battarra et al. [17] the Minimum Multiple Trip Vehicle Routing Problem (MMTVRP).

More precisely, the MMTVRP can be defined on a complete undirected graph $G = (V, E)$, where $V = \{0, \dots, N\}$ is the set of vertices and $E = \{(i, j)|i, j \in V, i < j\}$ the set of edges. Vertex 0 represents the depot and vertices $1, \dots, N$ the customers. A set of commodities B has to be delivered to the set of customers. Commodities are *incompatible* with each other, that means they cannot be transported together in the same vehicle. Therefore, it can be supposed that each customer requires only one commodity: if a customer requires more than one commodity, it can be replicated as many times as the number of commodities he requires, associating with each replication one commodity and the corresponding quantity to deliver. Hence, each customer i requires a quantity Q_i of the commodity b_i to be delivered during a TW indicated by $[E_i, L_i]$. Vehicles are allowed to arrive at the location of each customer i before the corresponding E_i and wait until E_i to start service. Service at customer takes S_i time units.

A homogeneous fleet of vehicles is located at the depot. Vehicles have a fixed capacity Q_b that depends on the commodity b. Moreover, the depot is open during the time interval

$[E_{0b}, L_{0b}]$. The loading time S_{0b} at the depot is given by the sum of two terms. The first is a constant time S_0. The second is a trip-dependent value equal to a constant time S_b multiplied by the quantity of commodity b transported in the trip. Vehicles cannot operate longer than a given *spread time* ST, while the total duration of each trip must not exceed a spread time ST_b dependent on the transported commodity. Also routing costs depend on the transported commodity throughout cost C_b for each kilometer travelled. Finally, covering the distance D_{ij} that separates each pair of vertices i, j requires T_{ij} time units. Travelling times and distances do not coincide.

The set of trips assigned to the same vehicle is called *journey*. The MMTVRP calls for the determination of a set of trips and an assignment of each trip to a vehicle that minimizes the number of used vehicles (i.e., journeys) and, in case of ties, minimizes the routing cost. Moreover, it satisfies the following conditions:

(1) each trip starts and ends at the depot;

(2) a single commodity is delivered along each trip;

(3) each trip transporting commodity b starts not earlier than E_{0b} and ends not later than L_{0b};

(4) trips assigned to the same vehicle do not overlap in time;

(5) each customer is served exactly once;

(6) service at customer i must start in the range $[E_i, L_i]$;

(7) the sum of the demands of the customers in any trip delivering commodity b does not exceed Q_b;

(8) each trip takes less than ST_b when delivering commodity b;

(9) each journey takes no longer than ST.

To simplify notation, the letter v will be used to indicate both the journey and the vehicle that performs it. When confusion can arise, details will be given. A solution will be indicated by the Greek letter ξ, while a trip by the letter σ. To indicate that journey v is part of the solution ξ, we will use $v \in \xi$. Analogously, $\sigma \in v$ indicates that trip σ is assigned to journey v. T_v indicates the duration of a journey v, while $T(\sigma)$ indicates the duration of trip σ. The duration of a trip (resp. journey) is defined as the difference between the moment the vehicle that performs it (resp. the last trip in it) is back at the depot and the moment loading operations for the trip (resp. first trip in the journey) start.

6.3 Algorithm

The only algorithm for the MMTVRP that we are aware of, has been proposed by Battarra et al. [17]. In this work a two-step heuristic is repeated in an iterated manner. The first step creates a set of trips generated by means of a heuristic for the VRP with time windows. The second step combines the obtained trips into feasible journeys and obtains a solution. The creation of trips makes use of a guidance mechanism. It penalizes the creation of numerous trips that overlap the same time interval and the creation of long-lasting trips. These trips do not facilitate the packing step.

In this section we present the iterated local search (ILS) procedure for the MMTVRP. The general scheme of any ILS is presented in Algorithm 6 (Lourenço et al. [135]).

Algorithm 6 General Iterated Local Search scheme

```
1: Create a solution ξ_0
2: Apply LS to ξ_0 and obtain ξ*
3: while Termination criteria are not met do
4:     Perturb ξ* to obtain ξ'
5:     Apply LS to ξ' and obtain ξ'*
6:     if ξ*' is accepted then
7:         ξ* = ξ*'
8:     end if
9: end while
```

An initial solution ξ_0 is generated and improved by a local search (LS) procedure. The local optimum that is obtained is indicated by ξ^*. The following steps are repeated, until predetermined termination criteria are not met. The solution ξ^* is perturbed (modified) and a new current solution $\xi^{'}$ is obtained. The LS is applied to $\xi^{'}$ and a solution $\xi^{*'}$ is obtained. If $\xi^{*'}$ is accepted (for example, based on its quality) it becomes the new current local optimum ξ^*.

Perturbation plays a key role in the diversification of the search: small perturbations are likely to create a $\xi^{'}$ that falls back into ξ^* after LS is applied, resulting in an inefficient exploration of the search space. On the other hand, large perturbations make the ILS comparable to a multi-start algorithm.

Our ILS, indicated with $\mathcal{A}_{ILS}$, manages permutations Ψ of the N customers, usually called *giant tour* in the literature. Initially, a permutation Ψ_0 is created as explained in Section 6.3.2 and the number of vehicles M is set to a valid upper bound, for example to N.

The initial solution ξ_0 is obtained from the giant tour Ψ_0 by the *AdSplit* procedure (see Section 6.4). A first local optimum ξ^* is obtained by applying LS (Section 6.3.3). If ξ^* is not feasible, it undergoes the *Repair* procedure (Section 6.3.4). At each step, the current ξ^* is perturbed in order to obtain solution $\xi^{'}$. A perturbation consists in crossing the permutation Ψ^* associated with ξ^*, with another permutation Ψ_1. The resulting permutation $\Psi^{'}$ undergoes *AdSplit* and $\xi^{'}$ is obtained. A new local minimum $\xi^{'*}$ is obtained by applying LS and, if needed, the *Repair* procedure. Let $M(\xi)$ and $D(\xi)$ indicate the number of vehicles and the routing cost of the solution ξ. If $\xi^{'*}$ uses of less vehicles than ξ^* or if $M(\xi^{'*}) = M(\xi^*)$ and $D(\xi^{'*}) < D(\xi^*)$, then $\xi^{'*}$ is accepted and becomes the current ξ^*.

The $\mathcal{A}_{ILS}$ algorithm is illustrated in Algorithm 7. We now explain in details the various steps of the algorithm.

Algorithm 7 $\mathcal{A}_{ILS}$

$M = N$, $M(\xi^*) = N$, $D(\xi^*) = \infty$
Create a permutation Ψ_0
Apply *AdSplit* to Ψ_0 to obtain ξ_0
Apply LS to ξ_0 and obtain ξ^*
if ξ^* is not feasible **then**
  Repair ξ^*
end if
while Termination criteria are not met **do**
  Create a permutation Ψ_1
  Cross Ψ_1 with Ψ^* to obtain $\Psi^{'}$
  Apply *AdSplit* to $\Psi^{'}$ to obtain $\xi^{'}$
  Apply LS to $\xi^{'}$ and obtain $\xi^{'*}$
  if $\xi^{'*}$ is not feasible **then**
    Repair $\xi^{'*}$
  end if
  if $\Big(M(\xi^{'*}) < M(\xi^*)\Big)$ or $\Big(M(\xi^{'*}) = M(\xi^*) \wedge D(\xi^{'*}) < D(\xi^*)\Big)$ **then**
    Set $\xi^* = \xi^{'*}$, $\Psi^* = \Psi^{'*}$, $M = M(\xi^{'*})$
  end if
end while

6.3.1 Objective function and search space

The strategic nature of the problem requires the minimization of the number of vehicles used, breaking ties in favor of solutions with lower routing cost. During the search phase, infeasibility with respect to time window and spread time violations is allowed. Given a solution ξ, time window violation is denoted $TW(\xi)$, and spread time violation $ST(\xi)$. The violation of the TW is calculated as proposed by Nagata et al. [145]. When a vehicle arrives late at a customer location, it is allowed to drive back in time in order to meet the TW. A penalization proportional to the late arrival is added to the objective function. Precisely, let t_i^{ξ} be the arrival time at customer i in solution ξ. Then,

$$TW(\xi) = \sum_{\substack{i \in V \\ t_i^{\xi} > L_i}} L_i - t_i^{\xi}.$$

Analogously, the spread time violation is calculated as the maximum between 0 and the difference between trip (resp. journey) duration and ST_b (resp. ST). Precisely,

$$ST(\xi) = \sum_{\substack{v \in \xi \\ T_v > ST}} (T_v - ST) + \sum_{v \in \xi} \sum_{\substack{\sigma \in v \\ T(\sigma) > ST_{b_\sigma}}} (ST_{b_\sigma} - T(\sigma))$$

where b_σ is the commodity transported in trip σ. Capacity constraints are always respected and each trip transports only one commodity.

The cost $c(\xi)$ of a solution ξ is defined as:

$$c(\xi) = \alpha M(\xi) + D(\xi) + T(\xi) + \theta TW(\xi) + \varpi ST(\xi) \tag{6.1}$$

where θ and ϖ are the penalization factors, while α is a coefficient that represents the cost of using a vehicle.

Note that in our problem, trips are packed into vehicles with respect to the time dimension, but solutions are evaluated by considering the number of vehicles used and the routing cost. Note also that a higher routing cost can correspond to a shorter trip duration due, for example, to the reduction of waiting times. Shorter trips in terms of duration, are more suitable in terms of journey packaging and consequently can lead to a solution that makes use of a smaller fleet. Hence, including the term $T(\xi)$ in the solution cost may help driving the search towards solutions with a balance between routing cost and duration.

6.3.2 Giant tour creation

This section introduces the procedure *CommodityCreate* that creates the permutation Ψ at steps 2 and 9 of Algorithm 7. Such permutation is initialized with a randomly selected customer i whose demand is stored in $\tilde{Q}$, i.e., $\tilde{Q} = Q_i$. The next customer z is selected as

$$z = \arg \min_{\substack{j:b_j=b_i \\ E_i+S_i+T_{ij} \leq L_j}} \{L_j - E_i - S_i - T_{ij}\}. \tag{6.2}$$

If such customer z exists and $\tilde{Q} + Q_z \leq Q_{b_z}$ is respected (i.e., the vehicle capacity is not violated), z is inserted as the next customer in Ψ, otherwise another customer is randomly chosen, $\tilde{Q}$ is set to zero and the procedure is repeated until all the customers are in Ψ.

We observe that procedure *CommodityCreate* (by Equation (6.2)), tries to concatenate sequences of customers that can be transported into the same trip, taking into account time windows and capacity constraints. Procedure *AdSplit* (see Section 6.4) turns a permutation Ψ into a solution by splitting it into trips that are assigned to vehicles. We experimentally found that procedure *CommodityCreate* creates permutations that can be better split by *AdSplit* than randomly constructed permutations. This is illustrated by the results reported in Table 6.1, where *CommodityCreate* is compared with *RndCreate* that randomly creates a permutation on the real-world instances described in Section 6.5.2. For each test instance, 100 giant tours are generated with each procedure and evaluated by means of *AdSplit*. In Table 6.1 average results over the 100 giant tours are reported, time expressed in milliseconds. It can be noticed that *CommodityCreate* creates a giant tour that on average provides a better solution once evaluated by *AdSplit*. In addition, although it takes more than twice the computational time used by *RndCreate*, in both cases CPU times are in the order of milliseconds. Gap values are calculated as the difference between the values obtained by *CommodityCreate* and *RndCreate*, divided by the value obtained by *RndCreate*.

6.3.3 Local Search

Each time a giant tour is created, a solution ξ is obtained from it by means of the *AdSplit* procedure (detailed in Section 6.4). The LS is then applied to ξ in order to obtain a better quality solution.

The LS procedure considers four classical moves used in the VRP context: relocation, swap, 2-opt and 2-opt* (see, e.g., Prins [166]). Moreover, swaps between sequences of customers of different sizes are implemented. We do not limit the size of the segments due to the restricted capacity of the vehicles that allow few deliveries per trip, hence reducing the

	RndCreate		CommodityCreate			
	value		value		gap	
day	cost	time (ms)	cost	time (ms)	cost	time (ms)
1	45574100	4.36	44040300	11.39	-3.4%	161.2%
2	62313900	5.93	57854900	13.26	-7.2%	123.6%
3	47121500	4.68	45028800	12.64	-4.4%	170.1%
4	61398600	5.30	57709900	14.97	-6.0%	182.5%
5	63003400	5.93	57801500	12.94	-8.3%	118.2%
6	21381300	3.12	18811100	6.86	-12.0%	119.9%

Table 6.1: Comparison of the procedures for the creation of the initial permutation

number of segments of each trip. Finally, relocation and swap of trips among vehicles are implemented. These moves exploit the peculiar characteristic of the problem of allowing vehicles to perform several trips. Moves are evaluated according to Equation (6.1).

As already mentioned, time windows violation is calculated as proposed by Nagata et al. [145]: when a vehicle arrives late at customer location, it is allowed to drive back in time in order to meet the time window. A penalization proportional to the late arrival is added into the objective function. This penalization scheme has been extended by Vidal et al. [214]. In particular, inter- and intra-route moves are considered as a concatenation of segments. Segments can start or end with a depot. The concatenation operator will be indicated by the sign $\oplus$. For example, swapping customer w_i in trip $(0, w_1, \ldots, w_{n_1}, 0)$ with customer u_j in trip $(0, u_1, \ldots, u_{n_2}, 0)$ can be viewed as $(0, w_1, \ldots, w_{i-1}) \oplus (u_j) \oplus (w_{i+1}, \ldots, w_{n_1}, 0)$ and $(0, u_1, \ldots, u_{j-1}) \oplus (w_i) \oplus (u_{j+1}, \ldots, u_{n_2}, 0)$.

Let $\rho_1 = (v_1^1, \ldots, v_{n_1}^1)$ and $\rho_2 = (v_1^2, \ldots, v_{n_2}^2)$ be two segments. We suppose the customers in ρ_1 and ρ_2 require the same commodity b_{ρ_1}. Vidal et al. [214] propose to calculate the duration $T(\rho_1 \oplus \rho_2)$, the time windows violation $TW(\rho_1 \oplus \rho_2)$, the earliest $E(\rho_1 \oplus \rho_2)$ and the latest $L(\rho_1 \oplus \rho_2)$ departure times from the location of the first customer in ρ_1, which correspond to the minimum values of $T(\rho_1 \oplus \rho_2)$ and $TW(\rho_1 \oplus \rho_2)$, the routing cost $D(\rho_1 \oplus \rho_2)$ and the total demand $Q(\rho_1 \oplus \rho_2)$ as follows[1]:

$$T(\rho_1 \oplus \rho_2) = T(\rho_2) + T(\rho_1) + T_{v_{n_1}^1, v_1^2} + \Delta_{WT}; \tag{6.3}$$

$$TW(\rho_1 \oplus \rho_2) = TW(\rho_1) + TW(\rho_2) + \Delta_{TW}; \tag{6.4}$$

$$E(\rho_1 \oplus \rho_2) = \max\{E(\rho_2) - \Delta, E(\rho_1)\} - \Delta_{WT}; \tag{6.5}$$

$$L(\rho_1 \oplus \rho_2) = \min\{L(\rho_2) - \Delta, L(\rho_1)\} + \Delta_{TW}; \tag{6.6}$$

$$D(\rho_1 \oplus \rho_2) = D(\rho_1) + D(\rho_2) + C_{b_{\rho_1}} D_{v_{n_1}^1, v_1^2}; \tag{6.7}$$

$$Q(\rho_1 \oplus \rho_2) = Q(\rho_1) + Q(\rho_2) \tag{6.8}$$

where

$$\Delta = T(\rho_1) - TW(\rho_1) + T_{v_{n_1}^1, v_1^2};$$

$$\Delta_{WT} = \max\{E(\rho_2) - \Delta - L(\rho_1), 0\};$$

[1] To be precise, the equivalent of Equation (6.7) in Vidal et al. [214] refers to the travelled distance and corresponds to the case where $C_b = 1$ for all commodities b.

$$\Delta_{TW} = \max \{E(\rho_1) + \Delta - L(\rho_2), 0\}.$$

Quantities are initialized by setting, for each customer i, $T(i) = S_i$, $TW(i) = 0$, $E(i) = E_i$, $L(i) = L_i$, $D(i) = 0$ and $Q(i) = Q_i$.

Relations (6.3)–(6.8) are developed for the VRP, where each vehicle performs only one trip σ. Consequently, the vehicle assigned to σ can leave the depot during the time interval $[E(\sigma), L(\sigma)]$ in order to minimize the trip duration and the TW violation. In problems where vehicles are allowed to perform several trips as in our case, we do not have the guarantee that each trip σ can leave the depot during the time interval $[E(\sigma), L(\sigma)]$. This, for example, happens when the trip σ^- that is performed right before σ, by the same vehicle, arrives back at the depot at $t_{\sigma^-} > L(\sigma)$. The following relations, (Vidal et al. [214]):

$$T(\rho)(t) = T(\rho) + \max \{0, E(\rho) - t\}; \tag{6.9}$$
$$TW(\rho)(t) = TW(\rho) + \max \{0, t - L(\rho)\} \tag{6.10}$$

allow to calculate the segment duration and TW penalization as functions of the exact starting time t of the segment itself. Moreover, performing a segment ρ takes $T(\rho)(t) - TW(\rho)(t)$, if the vehicle starts at time t. This allows to calculate arrival times of vehicles at the depot and then to know when they are ready for a new trip (see Cattaruzza et al. [31]). Moreover, this relation allows to calculate the spread time violations.

Let m be a particular move considered in the LS. Let suppose that m involves a trip $\sigma \in v$. Variation in the quantities $T(\sigma)$, $TW(\sigma)$, $E(\sigma)$, $L(\sigma)$, $D(\sigma)$ and $Q(\sigma)$ can be calculated in $O(1)$ using Relations (6.3)–(6.8). Note also that a variation in σ can be propagated to the following trips. For example an increment in $T(\sigma)$ can delay the trip completion time and then delay the start of its subsequent trip σ^+, with a possible increment in the TW penalization $TW(\sigma^+)$. Consequently, the full cost variation due to m can be calculated in $O(|v|)$ where $|v|$ represents the number of trips assigned to the vehicle v.

Due to the strategic objective of minimizing the fleet, the LS needs to reduce the number of used vehicles. In order to achieve this, we first detect a vehicle that can presumably be easily removed, by assigning the customers it has to serve to other vehicles. We, hence, define the *worst-packed* vehicle to be the non-empty vehicle v^* such that

$$v^* = \arg\min_{v} \frac{\sum_{\sigma \in v} T(\sigma)}{ST} \prod_{\sigma \in v} \frac{Q(\sigma)}{Q_{b_\sigma}} \tag{6.11}$$

where b_σ indicates the commodity transported in trip σ. Equation (6.11) selects the vehicle v^* which journey does not well exploit the spread time available (by factor $\frac{\sum_{\sigma \in v} T(\sigma)}{ST}$) and which trips do not employ the available capacity (by factor $\prod_{\sigma \in v} \frac{Q(\sigma)}{Q_{b_\sigma}}$).

Let ξ be the current solution and $c(\xi)$ be its cost defined as in Equation (6.1). As before, let m be a particular move considered by the LS and ξ^m be the solution that is obtained by implementing m. The cost variation $\Delta_m = c(\xi^m) - c(\xi)$ is defined. If m involves v^*, a quantity equal to α is added (resp. subtracted) to Δ_m if at least one customer is added to (resp. removed from) v^*. This fosters moves that remove customers from v^* and discourage moves that replenish it. This is inspired by the work of Lodi et al. [134] for the multi-dimensional bin packing problem: in order to reduce the number of used bins, a candidate bin is detected and tried to be emptied by inserting objects in the other bins.

Initially, each one of the moves listed at the beginning of the section is declared *active* and a weight is associated with it. At each step of the LS, vehicle v^* is determined and a move M is probabilistically selected based on its weight. The neighbourhood that m defines is fully explored. If the exploration fails in finding a better solution, the move becomes *inactive* and cannot be selected again until the current solution is improved. When all the moves are inactive the LS terminates. Weights are updated as in Cattaruzza et al. [33]. Once the LS terminates, trips are concatenated one after the other and depots are removed to obtain a new giant tour $\tilde{\Psi}$.

6.3.4 *Repair* procedure

When LS terminates, it produces a solution ξ that can be infeasible. In this case, ξ undergoes the *Repair* procedure, that tries to obtain from it a feasible solution. First, an infeasible trip is detected. Then, part of that trip is moved to a new vehicle in order to eliminate its infeasibility. The operation is repeated until infeasibility is eliminated or the $M(\xi^*)$ vehicles are used. Due to the need of using a new vehicle for this operation, ξ is repaired only if it uses of strictly less vehicles than $M(\xi^*)$. Then, LS is reapplied with θ multiplied by 10000 to avoid re-inserting infeasibility in the solution, or to try to eliminate the still present ones. Computational experiments showed that the spread time is infrequently violated. Hence, the value of ϖ is not modified during the *Repair* procedure.

6.3.5 Crossover

The classical *order crossover* is used to obtain a new permutation Ψ from the permutation Ψ^* associated with the best solution found so far and a new generated permutation Ψ_1. Two cutting points are randomly determined and all the customers in between in Ψ^* are copied in Ψ. It is then completed by circularly sweeping Ψ_1 and inserting those customers that are not yet in Ψ. A simple example is depicted in Figure 6.1 where the cutting points are denoted by the vertical bars.

$$
\begin{array}{lccc|cccc|ccc}
 & & & & i=4 & & & j=7 & & & \\
 & & & & \downarrow & & & \downarrow & & & \\
\Psi^*: & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\Psi_1: & 7 & 5 & 1 & 3 & 2 & 8 & 9 & 4 & 10 & 6 \\
 & & & & & & & & & & \\
\Psi: & 2 & 8 & 9 & 4 & 5 & 6 & 7 & 10 & 1 & 3
\end{array}
$$

Figure 6.1: Ordered Crossover

6.4 *AdSplit* Procedure

The *AdSplit* procedure obtains a solution from a permutation Ψ of the N customers. *AdSplit* is inspired by the procedure proposed by Prins [166] to obtain a VRP solution from a given permutation.

AdSplit works on an auxiliary graph H. Its construction is explained in Section 6.4.1. Then, a path from the starting node 0 to the final node N is selected and each arc (representing a trip) is assigned to a vehicle. The selection of the arcs forming the path and the assignment of trips (arcs) to vehicles is done by a labelling procedure explained in Section 6.4.2.

6.4.1 Graph construction

The auxiliary graph is indicated by $H = (V', A')$. V' contains $N+1$ nodes indexed from 0 to N. Arc (i, j), $i < j$, represents a trip σ_{i+1}^{j} serving customers from Ψ_{i+1} to Ψ_j in the order they are in Ψ, namely, $\sigma_{i+1}^{j} = (0, \Psi_{i+1}, \ldots, \Psi_j, 0)$. Arc (i, j) is added to A' if customers Ψ_k, $k = i+1, \ldots, j$, require the same commodity b and $\sum_{k=i+1}^{j} Q_k \leq Q_b$ (i.e., the capacity constraint is respected). With each arc (i, j) is associated the cost

$$c_{ij} = D(\sigma_{i+1}^{j}) + T(\sigma_{i+1}^{j}) + \theta TW(\sigma_{i+1}^{j}) + \varpi ST(\sigma_{i+1}^{j}), \tag{6.12}$$

given by the sum of the routing cost, the duration, the time window and the spread time violation penalized by factors θ and ϖ of trip σ_{i+1}^{j}. The construction of H is illustrated on Figure 6.2, on a small example with 5 customers and 2 commodities. White customers require one commodity, grey customers the other commodity. Data are given in Table 6.2. $\theta = 2$, $Q = 40$. To simplify the exposition, the service time at the depot is always considered null and the spread times are large enough that spread time violations do not occur.

Note that c_{ij} is the cost of trip σ_{i+1}^{j} as it leaves the depot at $t \in [E(\sigma_{i+1}^{j}), L(\sigma_{i+1}^{j})]$. As a consequence, penalizations due to later departure from the depot cannot be taken into account by the routing costs c_{ij}. These penalizations considered by the labelling procedure presented in the next section. For example, trip $\sigma_2^2 = (0, 2, 0)$, (that corresponds to arc $(1, 2)$) has a cost of $c_{12} = D(\sigma_2^2) + T(\sigma_2^2) + +\theta TW(\sigma_2^2) + \varpi ST(\sigma_2^2) = 30 + 35 + 0 + 0 = 65$. $[E(\sigma_2^2), L(\sigma_2^2)] = [35, 60]$. Let suppose that trips σ_1^1 and σ_2^2 are assigned to the same vehicle. $T(\sigma_1^1) = 15$ and $[E(\sigma_1^1), L(\sigma_1^1)] = [95, 115]$. Hence, the vehicle is back at the depot after serving customer 1, not earlier than $t = 95 + 15 = 110$. If the vehicle starts performing trip σ_2^2 at $t = 110$, it would leave later than $L(\sigma_2^2) = 60$. This implies a TW violation equal to 50 (applying Equation (6.9)) with an increase in the trip cost equal to 100 (because $\theta = 2$).

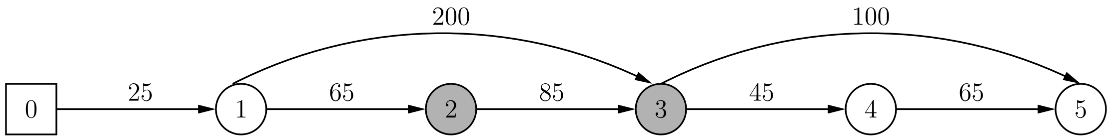


Figure 6.2: Auxiliary graph H

Data					
	S_i	Q_i	E_i	L_i	Commodity
v_0	-	0	0	200	-
v_1	5	20	100	120	0
v_2	5	20	50	75	1
v_3	5	20	50	75	1
v_4	5	20	50	100	0
v_5	5	20	50	100	0

Distance matrix						
	v_0	v_1	v_2	v_3	v_4	v_5
v_0	0	5	15	20	10	15
v_1	5	0	20	20	15	15
v_2	15	20	0	40	20	30
v_3	20	20	40	0	30	10
v_4	10	15	20	30	0	20
v_5	15	15	30	10	20	0

Table 6.2: Data for H in Figure 6.2

6.4.2 Assignment of trips to vehicles

Once H is computed, a solution can be obtained by a path from node 0 to node N, where each arc represents a trip. Depending on the way trips are assigned to vehicles, different solutions can be constructed from the same path. We propose a labelling procedure to select the path and to assign the trips (represented by arcs) to vehicles.

A label $\mathcal{L}_i$ associated with node i represents a path that goes from node 0 to node i and the corresponding partial solution that serves customers $\Psi_1, \ldots, \Psi_i$. Trips serving these customers are represented by the corresponding arcs in the path. Basically, when a label is extended with an arc (trip), one new label is created for each possible assignment of this trip to a vehicle. To do so, as many fields as the best number of vehicles found so far (M) should be considered in the label definition, to keep information on the trip assignments. One drawback of this approach would, however, be the very large number of labels generated if M is large. In order to deal with this difficulty, we slightly adapt this scheme and introduce a parameter $\tilde{M}$ to limit the number of simultaneously *open* vehicles. By *open* vehicles we mean that the new trip can only be assigned to these vehicles. Hence, the label definition is as follows.

A label $\mathcal{L}$ is formed by $\tilde{M} + 3$ fields. The first $\tilde{M}$ fields contain the availability times of the vehicles at the depot, while fields $\tilde{M} + 1$, $\tilde{M} + 2$ and $\tilde{M} + 3$ contain the total routing cost, the total TW violation and the total spread time violation. The cost $c(\mathcal{L})$ of a label $\mathcal{L}$ is the cost of the partial solution it represents.

Starting from node 0, labels are extended to the following nodes. In particular a label $\mathcal{L}_i$ associated with node i is extended to label $\mathcal{L}_j$ associated with node j through arc (i, j), namely, by assigning trip $(\Psi_{i+1}, \ldots, \Psi_j)$ to a certain vehicle. A predetermined number $\tilde{M}(< M)$ of vehicles are kept *open*, i.e., trips can be assigned to them. If the assignment does not violate spread time constraints, the extended label is kept. Otherwise it is discarded. When assigning a trip to all the open vehicles fails, the vehicle that would produce the biggest spread time violation is *closed*, i.e., no more trips can be assigned to it. A new empty vehicle is then declared *open* and the trip is assigned to it. A closed vehicle cannot be re-opened later.

Considering all the possible assignments of trips to vehicles can be highly time consuming, especially when the number of vehicles is large. Allowing trips to be assigned to a smaller number $\tilde{M}$ of vehicles reduces the combinations and speeds-up the procedure. When the number of vehicles is relatively small, $\tilde{M}$ can be set to M. Note that $\tilde{M}$ is not necessarily sufficient to serve all the customers.

Quantities defined in Equations (6.3)–(6.8) for trip σ_{i+1}^{j} are calculated whenever an arc (i, j) is considered. This allows to determine exact availability times of each vehicle when extending the labels.

A heuristic version of the dominance rule introduced in Cattaruzza et al. [31] is adopted to limit furthermore the number of labels associated with each node.

Dominance Rule 3. *Given two labels $\mathcal{L}_1$ and $\mathcal{L}_2$ associated with the same node i, we say that $\mathcal{L}_1$ dominates $\mathcal{L}_2$ if and only if*

$$c(\mathcal{L}_1) + \theta \sum_{j=1}^{\tilde{M}} \delta_j(\mathcal{L}_1, \mathcal{L}_2) \leq \gamma c(\mathcal{L}_2) \tag{6.13}$$

where $\delta_j(\mathcal{L}_1, \mathcal{L}_2) = \max\{T_j(\mathcal{L}_1) - T_j(\mathcal{L}_2), 0\}$ and $\gamma \geq 1$.

Intuitively, the summation in the Relation (6.13) evaluates the maximal TW penalization that can be introduced into the open vehicles of the solution represented by $\mathcal{L}_1$ with respect to the solution represented by $\mathcal{L}_2$, and without introducing any penalization in the solution represented by $\mathcal{L}_2$. This maximal penalization is the sum on all the open vehicles of the maximal penalization that can be introduced in each open vehicle in the solution ξ_1 represented by $\mathcal{L}_1$ with respect to the solution ξ_2 represented by $\mathcal{L}_2$. In turn, the maximal penalization that can be introduced in a particular open vehicle j in ξ_1 with respect to the open vehicle j in ξ_2 is the difference of their duration times (if this difference is positive) multiplied by the penalty θ. The parameter θ is the same as the one introduced in Equation (6.1). The bigger is γ the weaker is the dominance rule, the quicker a solution can be obtained from a permutation. Its value is dynamically adapted while traversing the graph based on the number of labels associated with each node. Precisely the following scheme (Cattaruzza et al. [31]) is adopted:

$$\gamma = \begin{cases} \gamma + \frac{|\mathcal{L}_i|}{1000\mathcal{L}_{\text{threshold}}} & \text{if } |\mathcal{L}_i| > \mathcal{L}_{\text{threshold}} \\ \gamma - \frac{\mathcal{L}_{\text{threshold}}}{1000|\mathcal{L}_i|} & \text{if } |\mathcal{L}_i| < \mathcal{L}_{\text{threshold}} \end{cases} \tag{6.14}$$

where $|\mathcal{L}_i|$ is the number of labels associated with node i and $\mathcal{L}_{\text{threshold}}$ is a threshold parameter that indicates the number of labels that is targeted to be kept with each node.

6.5 Computational results

This section describes the computational experiments conducted and reports the results obtained by our algorithm. The $\mathcal{A}_{ILS}$ algorithm is coded in C++, compiled with Visual Studio 2010 and it is run on a Intel Xeon W3550 3.07GHz and RAM of 12 Gb.

Our testing is conducted first on a set of real-world MMTVRP instances proposed by Battarra et al. [17]. Then, we tested the potential benefit of multiple trips versus traditional single trip solutions by considering VRPTW instances from the literature.

6.5.1 Parameter setting

Three main parameters are involved in $\mathcal{A}_{ILS}$ and need to be set. Two are used in the Dominance Rule 3 and are γ and $\mathcal{L}_{\text{threshold}}$. The third is $\tilde{M}$. The value of γ is initially set to 10, while $\tilde{M}$ is set to 20. Extensive preliminary tests showed that the procedure becomes computationally efficient when γ approached that value, while $\tilde{M} = 20$ leaves a certain freedom in trip assignment, keeping the procedure computationally efficient. Moreover, $\mathcal{L}_{\text{threshold}}$ is set to 15 as in Cattaruzza et al. [32].

Different schemes for the penalties used in Equation (6.1) were tested in a preliminary phase and the setting that gave the best results is the following: α is initially set to M^2, while ϖ and θ are set to N. Penalty α is doubled whenever an iteration does not reduce the number of used vehicles and it is set back to its initial value otherwise. The other penalties are kept constant.

6.5.2 Results on Battarra et al.'s instances

Experiments are conducted on six real-world instances proposed by Battarra et al. [17]. Supermarkets located in a regional territory need to be weekly supplied with three commodities. The time unit is equivalent to one minute. The time horizon is 1440 time units long, i.e. one day, while all the spread times ST and ST_b are set to 840, i.e., 14 hours. Service time at the depot takes 15 minutes plus 1 unit of time for each unit of quantity loaded into the vehicle. Vehicle capacity is the same for each commodity, since demand quantities are normalized to a common unit. $C_b = 1$ for all the three commodities b. Hence, the routing cost corresponds to the travelled distance, to which we refer in the presentation of the computational results.

Table 6.3 reports the data related to each instance (day 1 to day 6). Abbreviations VG, F and NP stand for Vegetables, Fresh-Food and Non-Perishable products that represent the three commodities that cannot be transported together. The values $\bar{Q}_b$, $\bar{E}_b$, $L_b^{\max}$, $\bar{W}_b$ represent average values over the N_b customers associated with commodity b. In particular $\bar{Q}_b$ is the ratio of average demand of customers with respect to the vehicle capacity, $\bar{E}_b$ is the average TW start time, $L_b^{\max}$ is the maximal TW closing time and $\bar{W}_b$ is the average TW width.

Note that some values in Table 6.3 (marked with an asterisk) are different than those in Battarra et al. [17] (Table 1, p. 3048), since the values there were incorrect.

Table 6.4 reports results. Columns M indicate the number of vehicles used, *dist* the total travelled distance, *# trips* the number of trips. Columns *gap* report gap values with respect to solutions obtained by Battarra et al. [17]. Gap values are calculated as the difference between values obtained by Battarra et al. [17] and values obtained by $\mathcal{A}_{ILS}$ divided by values obtained by Battarra et al. [17]. We recall the strategic nature of the objective: the number of used vehicles needs to be minimized first. Ties are broken in favor of solutions with lower travelled distance.

The algorithm is run 5 times on each instance and stopped after 10 minutes of computing time. Battarra et al. [17] algorithm was run approximately for the same time on each instance. Hence, due to computing time equivalence, this information is not inserted in Table 6.4.

day	N	Vegetables				
		N_{VG}	Q_{VG}	E_{VG}	$L_{VG}^{\max}$	W_{VG}
1	394	167	0.24	260.87	540	144.67
2	473	182*	0.26	255.36	540	151.04
3	403	175	0.20	260.60	540	146.46
4	465	174	0.21*	252.96	540	151.09
5	481	184	0.27*	258.61	540	148.59
6	297	183	0.30	256.26	540	150.38
Average	419*	178	0.25	257.44	540	148.71

day	N	Fresh Food				
		N_F	Q_F	E_F	$L_F^{\max}$	W_V
1	394	84	0.10	397.50	1080	108.93
2	473	149*	0.09	424.03*	1140	246.14*
3	403	91	0.11*	427.91	1140	232.91
4	465	123	0.09*	451.83	1140*	233.90
5	481	144	0.09*	403.65	1140	218.54
6	297	67	0.10*	477.54	1140	328.88
Average	419*	110*	0.10*	430.41*	1130*	228.22*

day	N	Non-perishable				
		N_{NP}	Q_{NP}	E_{NP}	$L_{NP}^{\max}$	W_{NP}
1	394	143	0.64	614.06	1140	127.76
2	473	142	0.66	585.63	1140	138.80
3	403	137	0.65	607.23	1140	142.55
4	465	168	0.61	599.11	1140	137.86
5	481	153	0.66*	591.96	1140	135.88
6	297	47	0.64*	617.23	1110	102.77
Average	419*	132	0.64*	602.54	1135	130.94

Table 6.3: Statistics on Battarra et al. [17] instances. Values marked with an asterisk are corrected with respect to the original paper.

Battarra et al. [17]		$\mathcal{A}_{ILS}$ Average					$\mathcal{A}_{ILS}$ Best				
M	dist	M	dist	# trips	gap M	gap dist	M	dist	# trips	gap M	gap dist
83	34004	80.8	32212.8	164.2	-2.7%	-5.3%	80	32296	164	-3.6%	-5.0%
92	35725	84.8	35091.1	175.2	-7.8%	-1.8%	84	34737	177	-8.7%	-2.8%
75	33908	69.8	33058.5	152.8	-6.9%	-2.5%	69	33173	152	-8.0%	-2.2%
91	38680	84.8	37678.2	172.2	-6.8%	-2.6%	84	38007	172	-7.7%	-1.7%
94	38205	86.4	37058.6	187.2	-8.1%	-3.0%	86	36775	188	-8.5%	-3.7%
63	18985	56.2	18751.6	107.8	-10.8%	-1.2%	55	19232	110	-12.7%	1.3%

Table 6.4: Results on Battarra et al. [17] instances

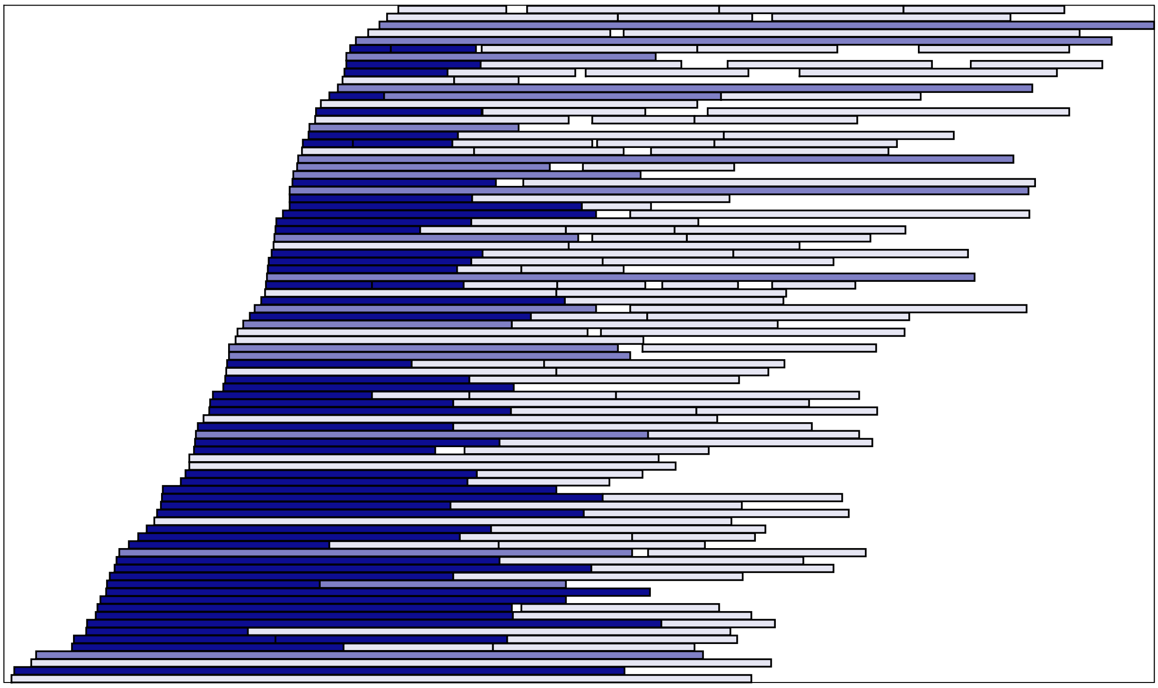


Figure 6.3: Multi-trips of day 5 solution: different color represents a different commodity

Columns $\mathcal{A}_{ILS}$ *Average* report average results on the five runs, while columns $\mathcal{A}_{ILS}$ *Best* report the best solution obtained. It can be noticed that better solutions are always obtained. Fewer vehicles are needed to perform delivery on each day. In particular, the average gap of saved vehicles goes from 2.7% for day 1, to 10.8% for day 6. Moreover, travelled distances are always reduced even if less vehicles are used. The best solution found by $\mathcal{A}_{ILS}$ for day 5 is drawn in Figure 6.3.

Each row represents a vehicle and each rectangle a trip. Rectangles filled with different colors, represent trips that serve a different commodity. The horizontal axis represents the time. From the figure it can be noticed that most of the journeys are composed by more than one trip and that trips assigned to the same vehicle actually deliver different commodities.

6.5.3 MTVRP versus VRP in fleet dimensioning problems

In this section we evaluate the possible benefit, at a fleet sizing level, of letting the vehicles perform several trips. In order to conduct such analysis we run our algorithm $\mathcal{A}_{ILS}$ on the well-known instances proposed by Solomon [196] and on those proposed by Gehring and Homberger [82], both designed for the VRPTW, and respectively indicated as SLM and HMB in the following. The 56 SLM instances are divided into 6 groups: C1, C2, R1, R2, RC1, RC2. Customers are clustered in groups C1 and C2, while they are randomly located in R1 and R2. Groups RC1 and RC2 contain a mix of clustered and randomly located customers. Working days are longer in groups C2, R2, RC2. HMB instances are divided into 5 families, each including 60 instances with 200, 400, 600, 800 and 1000 customers respectively. For each family, instances are constructed, as in Solomon [196], grouping instances based on customer locations: groups C1, C2, R1, R2, RC1, RC2 are then formed with 10 instances each. Results obtained on SLM instances are reported in Section 6.5.3, while those on HMB instances are reported in Section 6.5.3. Both families of instances have been addressed in the literature with the objective of minimizing the fleet size first, and the travelled distance second.

Solutions obtained on VRPTW instances allowing vehicles to perform multi-trips will be referred to as Multi Trip VRPTW (MTVRPTW) solutions. The MMTVRP generalizes the MTVRPTW. Hence, the latter can be heuristically solved using $\mathcal{A}_{ILS}$.

Results on Solomon's instances

Algorithm $\mathcal{A}_{ILS}$ is run five times on each of the SLM instance, with the objective of reducing the number of vehicles taking advantage from the multi-trip possibility. Each run is stopped after 10 minutes of computation. In three cases, we were able to find a solution where the size of the fleet is decreased by one vehicle: customers in instances C103, C104 and C109 can be served by 9 vehicles, while the best known VRPTW solution for these instances uses 10 vehicles. Results are reported in Table 6.5 for these three instances. Best known solutions of SLM instances can be found on the dedicated page of Transportation Optimization Portal (TOP) of Sintef web site [192]. Column *Instance* reports the instance name, columns *Best known VRPTW* report data related to VRPTW solutions, while columns *MTVRPTW* report data related to the solutions found by $\mathcal{A}_{ILS}$. Columns *dist* indicate the travelled distance, *% dist* the percentage increase of travelled distance of the MTVRP solution with respect to the best known VRPTW solution. Finally column *trips* shows the number of trips performed by the vehicles. Detailed solutions are reported in Appendix B.

Instance	Best known VRPTW		MTVRPTW			
	M	dist	M	dist	% dist	trips
C103	10	828.06	9	1072.32	29.5%	10
C104	10	824.78	9	937.646	13.7%	10
C104	10	828.94	9	958.105	15.6%	11

Table 6.5: Results on Solomon's instances

Solutions for instances C103 and C104 are characterized by a vehicle that performs two trips, while solution for C109 has two vehicles performing two trips. Travelled distances grow by about 15% for instances C104 and C109, and 30% for instance C103.

Results on Gehring and Homberger's instances

The $\mathcal{A}_{ILS}$ algorithm is also run five times on each of the 300 HMB instances. Each run is stopped after 30 minutes of computation. Tables 6.6–6.10 report all the instances for which $\mathcal{A}_{ILS}$ was able to reduce the fleet size. Column *Instance* indicates the instance name that is made up by three fields: the instance group, the family that identifies the size and the instance number: for example instance C1_2_3 is the third instance of the family of instances with 200 customers and belongs to group C1 (clients are clustered). The other column names are as in Section 6.5.3.

Instance	Best known VRPTW		MTVRPTW			
	M	dist	M	dist	% dist	trips
C1_2_3	18	2707.35	17	3277.13	21.0%	21
C1_2_4	18	2643.31	17	2935.30	11.0%	21
R1_2_2	18	4039.86	17	4942.63	22.3%	21
R1_2_3	18	3381.96	15	4228.01	25.0%	21
R1_2_4	18	3057.81	11	3765.33	23.1%	20
R1_2_5	18	4107.86	17	5617.44	36.7%	25
R1_2_6	18	3583.14	15	4610.35	28.7%	26
R1_2_7	18	3150.11	12	4227.69	34.2%	23
R1_2_8	18	2951.99	10	3497.76	18.5%	20
R1_2_9	18	3760.58	15	4884.75	29.9%	23
R1_2_10	18	3301.18	13	3946.04	19.5%	21
R2_2_4	4	1981.29	3	2504.05	26.4%	6
R2_2_8	4	1849.87	2	2378.44	28.6%	4
R2_2_9	4	3092.04	3	3509.99	13.5%	5
R2_2_10	4	2654.97	3	3041.22	14.5%	6
RC1_2_2	18	3249.05	15	4131.22	27.2%	22
RC1_2_3	18	3008.33	12	3691.10	22.7%	19
RC1_2_4	18	2852.62	9	3257.46	14.2%	20
RC1_2_5	18	3371.00	16	4286.53	27.2%	22
RC1_2_6	18	3324.80	16	4184.29	25.9%	22
RC1_2_7	18	3189.32	15	3848.66	20.7%	22
RC1_2_8	18	3083.93	14	3627.40	17.6%	22
RC1_2_9	18	3081.13	14	3657.52	18.7%	23
RC1_2_10	18	3000.30	14	3396.07	13.2%	20
RC2_2_4	4	2038.56	3	2648.15	29.9%	6
RC2_2_9	4	2175.04	3	2725.01	25.3%	5
RC2_2_10	4	2015.60	3	2417.22	19.9%	5

Table 6.6: Results on Gehring and Homberger's instances with 200 customers

Instance	Best known VRPTW		MTVRPTW			
	M	dist	M	dist	% dist	trips
C1_4_4	36	6803.24	33	8296.59	22.0%	42
C1_4_10	36	6860.63	35	8514.54	24.1%	41
R1_4_2	36	8926.70	32	11681.60	30.9%	42
R1_4_3	36	7821.95	26	10506.60	34.3%	40
R1_4_4	36	7282.78	20	9078.26	24.7%	39
R1_4_5	36	9242.43	33	14067.10	52.2%	47
R1_4_6	36	8373.71	28	11532.10	37.7%	43
R1_4_7	36	7641.22	22	10451.70	36.8%	41
R1_4_8	36	7275.13	18	8779.20	20.7%	39
R1_4_9	36	8719.19	28	12314.80	41.2%	44
R1_4_10	36	8113.93	24	10373.20	27.8%	42
R2_4_3	8	5911.07	7	7965.91	34.8%	12
R2_4_4	8	4241.47	5	6561.51	54.7%	12
R2_4_5	8	7129.03	7	8693.30	21.9%	10
R2_4_6	8	6122.60	6	8075.53	31.9%	14
R2_4_7	8	5018.53	5	7165.41	42.8%	10
R2_4_8	8	4018.01	5	5772.93	43.7%	10
R2_4_9	8	6400.10	6	7945.18	24.1%	11
R2_4_10	8	5791.79	6	7174.28	23.9%	10
RC1_4_2	36	7905.66	31	10027.70	26.8%	42
RC1_4_3	36	7540.59	24	9746.32	29.3%	39
RC1_4_4	36	7310.35	20	8368.21	14.5%	38
RC1_4_5	36	8249.63	33	10464.00	26.8%	42
RC1_4_6	36	8177.80	33	10353.40	26.6%	40
RC1_4_7	36	7957.64	31	9771.01	22.8%	42
RC1_4_8	36	7760.23	29	9506.63	22.5%	43
RC1_4_9	36	7767.43	29	9308.95	19.8%	42
RC1_4_10	36	7609.21	28	9027.87	18.6%	41
RC2_4_4	8	3631.01	7	5514.25	51.9%	11
RC2_4_9	8	4551.80	7	6504.61	42.9%	10
RC2_4_10	8	4278.61	7	5522.47	29.1%	10

Table 6.7: Results on Gehring and Homberger's instances with 400 customers

Instance	Best known VRPTW		MTVRPTW			
	M	dist	M	dist	% dist	trips
C1_6_4	56	13558.54	55	15337.70	13.1%	59
R1_6_2	54	18863.43	46	25477.80	35.1%	69
R1_6_3	54	17040.40	38	22610.40	32.7%	60
R1_6_4	54	15819.62	29	19476.10	23.1%	63
R1_6_5	54	19771.90	49	30653.40	55.0%	70
R1_6_6	54	18041.87	40	25101.10	39.1%	63
R1_6_7	54	16615.13	33	22322.90	34.4%	59
R1_6_8	54	15716.15	26	19472.60	23.9%	59
R1_6_9	54	18708.67	44	28504.40	52.4%	67
R1_6_10	54	17801.43	36	23102.50	29.8%	60
R2_6_3	11	11200.10	10	15744.50	40.6%	14
R2_6_4	11	8029.37	7	12836.30	59.9%	18
R2_6_5	11	15096.20	10	19611.70	29.9%	17
R2_6_6	11	12506.57	8	18026.20	44.1%	20
R2_6_7	11	10066.34	8	14549.90	44.5%	16
R2_6_8	11	7609.96	6	11996.60	57.6%	17
R2_6_9	11	13377.56	9	18267.40	36.6%	22
R2_6_10	11	12253.47	8	16440.00	34.2%	16
RC1_6_2	55	16044.93	43	21438.90	33.6%	66
RC1_6_3	55	15273.98	35	19443.70	27.3%	62
RC1_6_4	55	14839.61	26	18179.70	22.5%	58
RC1_6_5	55	16693.26	47	22649.50	35.7%	63
RC1_6_6	55	16632.03	45	22505.80	35.3%	66
RC1_6_7	55	16145.64	42	21582.70	33.7%	62
RC1_6_8	55	15978.70	40	20271.10	26.9%	60
RC1_6_9	55	15922.60	40	20323.30	27.6%	62
RC1_6_10	55	15740.26	38	19493.00	23.8%	61
RC2_6_4	11	7076.49	8	11202.30	58.3%	15
RC2_6_10	11	9078.64	10	12448.00	37.1%	14

Table 6.8: Results on Gehring and Homberger's instances with 600 customers

Instance	Best known VRPTW		MTVRPTW			
	M	dist	M	dist	% dist	trips
R1_8_2	72	32598.51	63	45491.70	39.6%	91
R1_8_3	72	29506.45	49	40118.90	36.0%	82
R1_8_4	72	26838.04	36	34716.40	29.4%	75
R1_8_5	72	33861.43	69	59269.30	75.0%	89
R1_8_6	72	31154.87	53	44764.30	43.7%	78
R1_8_7	72	29010.78	44	39184.80	35.1%	77
R1_8_8	72	27766.11	35	35014.10	26.1%	76
R1_8_9	72	32629.99	61	54382.90	66.7%	92
R1_8_10	72	31187.35	49	43853.50	40.6%	83
R2_8_3	15	17741.68	11	28108.30	58.4%	26
R2_8_4	15	13219.06	10	23779.70	79.9%	34
R2_8_5	15	24285.89	13	33484.00	37.9%	24
R2_8_6	15	20480.79	11	31686.30	54.7%	25
R2_8_7	15	16697.82	10	25569.00	53.1%	26
R2_8_8	15	12748.16	8	21379.50	67.7%	28
R2_8_9	15	22402.79	11	33563.50	49.8%	29
R2_8_10	15	20459.29	10	31129.70	52.2%	34
RC1_8_2	72	29034.99	61	36158.60	24.5%	81
RC1_8_3	72	27905.64	51	34754.80	24.5%	80
RC1_8_4	72	27395.14	37	31756.40	15.9%	77
RC1_8_5	72	30277.12	66	40002.00	32.1%	87
RC1_8_6	72	30262.33	66	41289.30	36.4%	87
RC1_8_7	72	29862.44	59	38727.50	29.7%	82
RC1_8_8	72	29194.16	57	37558.50	28.7%	79
RC1_8_9	72	28978.35	56	35732.50	23.3%	78
RC1_8_10	72	28797.79	54	36338.50	26.2%	79
RC2_8_4	15	11006.56	14	19614.80	78.2%	29
RC2_8_9	15	15359.99	14	22229.50	44.7%	21
RC2_8_10	15	14454.62	13	20785.50	43.8%	23

Table 6.9: Results on Gehring and Homberger's instances with 800 customers

Instance	Best known VRPTW		MTVRPTW			
	M	dist	M	dist	% dist	trips
R1_10_2	91	49105.21	79	68653.4	39.8%	118
R1_10_3	91	45237.29	61	60502.2	33.7%	105
R1_10_4	91	42787.19	46	55992.1	30.9%	96
R1_10_5	91	51830.36	87	84327.3	62.7%	111
R1_10_6	91	47849.05	66	66041.8	38.0%	106
R1_10_7	91	44435.5	55	58941.7	32.6%	100
R1_10_8	91	42485.38	44	53170.6	25.2%	99
R1_10_9	91	50490.49	79	80817	60.1%	113
R1_10_10	91	48294.71	66	71050.5	47.1%	101
R2_10_3	19	25053.8	15	40903.9	63.3%	33
R2_10_4	19	18039.77	12	34515.2	91.3%	44
R2_10_5	19	36335.72	18	55406.9	52.5%	33
R2_10_6	19	30223.14	15	45554	50.7%	30
R2_10_7	19	23381.36	13	39746.9	70.0%	41
R2_10_8	19	17598.63	11	30565.6	73.7%	38
R2_10_9	19	33131.99	14	49412.1	49.1%	40
R2_10_10	19	30598.69	13	43716.6	42.9%	32
RC1_10_2	90	44129.42	76	56448.8	27.9%	105
RC1_10_3	90	42487.54	61	52138.5	22.7%	99
RC1_10_4	90	41613.58	45	49299.6	18.5%	94
RC1_10_5	90	45564.81	84	65915.9	44.7%	114
RC1_10_6	90	45303.67	84	67271.4	48.5%	109
RC1_10_7	90	44903.8	74	62628.8	39.5%	104
RC1_10_8	90	44366.01	68	59091.4	33.2%	98
RC1_10_9	90	44280.84	69	58346.2	31.8%	99
RC1_10_10	90	43896.78	67	56352.6	28.4%	99
RC2_10_4	18	15747.13	16	26204.3	66.4%	29
RC2_10_8	18	23787.26	17	33226.6	39.7%	24
RC2_10_9	18	23116.15	17	33786.1	46.2%	29
RC2_10_10	18	22076.9	15	30592.7	38.6%	21

Table 6.10: Results on Gehring and Homberger's instances with 1000 customers

The values of the best known VRPTW solutions are collected from the dedicated page of TOP of Sintef web site [193]. The fleet size is reduced for a total of 126 cases out of 300: namely, 27 cases for instances with 200 customers, 31 for instances with 400 customers, 29 for instances with 600 and 800 customers, and 20 for instances with 1000 customers. It is noticeable that for instances RC1_6_4, R1_8_8 and R1_10_8 less than half vehicles are needed (26, 35 and 44 instead of 55, 72 and 91 respectively) while for instances R2_2_8, RC1_2_4, R1_4_8, R1_8_4 and RC1_10_4 the size of the fleet is halved. It can be noticed that in these (but not only) particular cases the vehicles used in the VRPTW solutions are equal to $\left\lceil \frac{\sum_{i=1}^{N} Q_i}{Q} \right\rceil$, meaning that there is no possibility to further reduce the fleet size allowing one trip per vehicle.

Table 6.11 and Figure 6.4 give an insight on the fleet reduction reasons. Table 6.11 compares the best VRPTW solution and the MTVRPTW solution we found for instance R1_2_8. Comparison has been proposed for instance R1_2_8 because of the important fleet size reduction we obtain introducing the multi-trip aspect and because details of the best known VRPTW solution are available in the dedicated page of TOP Sintef web site [193]. Columns *Best known VRPTW solution* report details of the best known VRP solution for this instance, while columns *MTVRPTW* report details of the MTVRPTW solution we found. Column v indicates the vehicle, column *trip* the trip, columns D, T_v, T_t, Q report respectively the travelled distance, the travelled time of vehicle v, the travelled time of trip t, and the quantity transported in the trip. Column T_H *usage* reports the percentage of time in which the vehicle travels with respect to the depot time window width (that is the planning horizon T_H), while columns Q *usage* report the loading factor. The last row indicates the total travelled distance, and the average percentage of T_H and Q usage.

It can be observed that in the VRPTW solution, only 64.7% of the time horizon is used on average, while vehicles are on average almost full loaded . A first conclusion that can be drawn from this information is that capacity constraints are more restrictive than time constraints. Considering a fleet of vehicles with a larger capacity would be beneficial. A second possibility would be to allow vehicles to perform multiple trips as depicted in the MTVRPTW part of the table. This leads to a planning that on average exploits a larger part of the time horizon (88% in the considered example) while using less vehicles. On the contrary the loading factor decreases by 10%.

In Figure 6.4, each dot represents a solution for which the best known VRPTW solution is detailed in TOP Sintef web site [193]. We then calculated the percentage of T_H usage of each of these solutions regardless of whether we were able to reduce the fleet with the multi-trip possibility (these particular instances correspond to a point with a null abscissa). Three observations can be deducted from the figure. First, when the VRPTW solution averagely uses 90% or more of T_H, the fleet size is not likely to be reduced even by allowing multiple trips. On the contrary, when vehicles are used for less than 70% of T_H, it is probably beneficial to let vehicles be re-loaded and re-routed. Finally, when the journeys cover between 70% and 90% of the T_H, the multi-trip aspect can or cannot be useful depending on the particular instance. From this observation we can conclude the need of developing algorithms that efficiently solve both the VRPTW and the MTVRPTW.

Best known VRPTW solution						MTVRPTW solution							
v	D	T_v	Q	T_H usage	Q usage	v	trip	D	T_t	T_v	Q	T_H usage	Q usage
1	115.16	350.00	197	55.2%	98.5%	1	1	102.74	162.74	492.62	100	77.7%	50.0%
2	190.16	458.79	199	72.4%	99.5%		2	187.88	317.88		199		99.5%
3	222.78	626.78	199	98.9%	99.5%		3	2.00	12.00		9		4.5%
4	165.40	624.74	200	98.5%	100.0%	2	1	280.12	430.66	598.02	200	94.3%	100.0%
5	233.99	363.99	199	57.4%	99.5%		2	107.37	167.37		144		72.0%
6	131.97	416.92	198	65.8%	99.0%	3	1	137.56	247.56	599.22	196	94.5%	98.0%
7	188.53	406.50	200	64.1%	100.0%		2	221.66	351.66		187		93.5%
8	158.58	268.58	197	42.4%	98.5%	4	1	199.48	339.48	627.65	173	99.0%	86.5%
9	119.32	209.32	199	33.0%	99.5%		2	188.18	288.18		180		90.0%
10	179.46	279.46	198	44.1%	99.0%	5	1	252.30	392.30	599.39	197	94.5%	98.5%
11	213.48	489.87	199	77.3%	99.5%		2	127.08	207.08		186		93.0%
12	82.55	457.04	193	72.1%	96.5%	6	1	217.94	339.72	595.75	165	94.0%	82.5%
13	176.21	477.54	197	75.3%	98.5%		2	156.03	256.03		200		100.0%
14	132.25	387.33	196	61.1%	98.0%	7	1	188.85	298.85	578.22	199	91.2%	99.5%
15	136.95	236.95	200	37.4%	100.0%		2	179.38	279.38		198		99.0%
16	210.90	496.77	199	78.4%	99.5%	8	1	179.34	269.34	525.92	197	83.0%	98.5%
17	68.22	461.18	143	72.7%	71.5%		2	156.58	256.58		198		99.0%
18	226.09	367.80	200	58.0%	100.0%	9	1	151.88	241.88	597.13	186	94.2%	93.0%
							2	235.26	355.26		199		99.5%
						10	1	226.16	380.12	380.12	200	60.0%	100.0%
	2951.99			64.7%	97.6%			3497.76				88.2%	87.8%

Table 6.11: Comparison of VRPTW and MTVRPTW solutions for R1_2_8 Gehring and Homberger instance

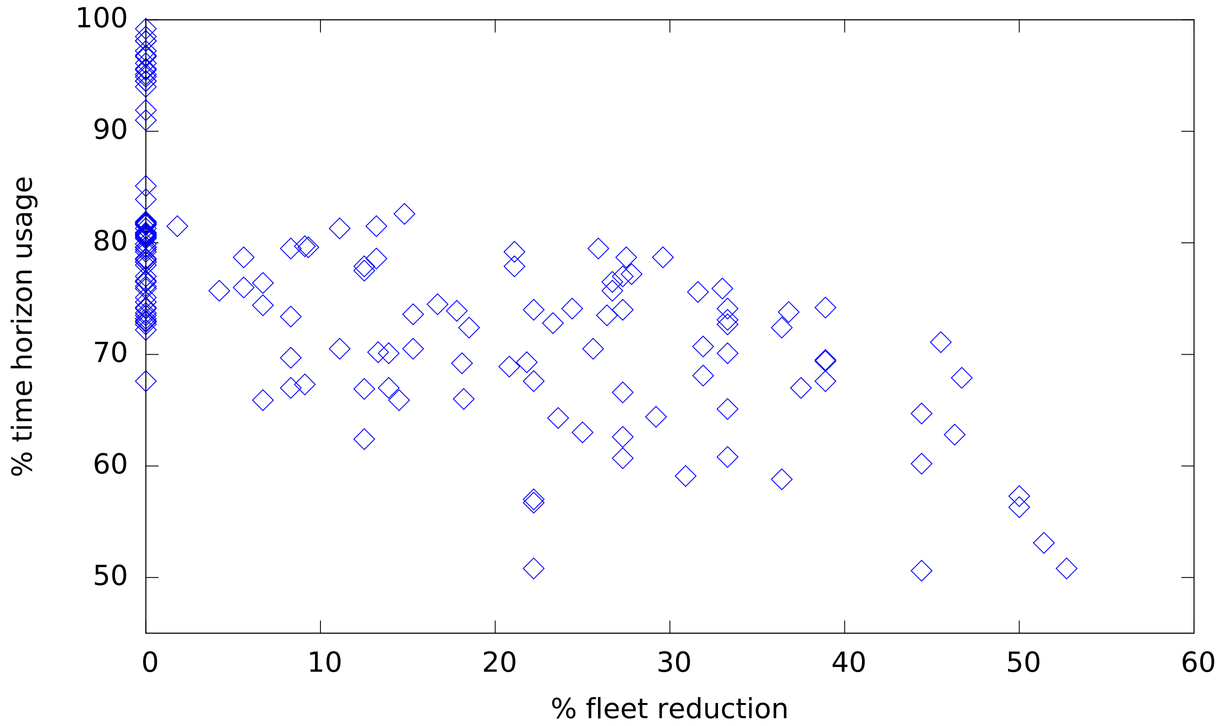


Figure 6.4: Instances analysis fleet reduction on time usage

6.6 Conclusions

This paper studied the minimum multiple trip vehicle routing problem. It mainly differs from the classical multi-trip VRP with time windows for the presence of incompatible commodities that need to be delivered to customers and the strategic nature of the objective function that requires the minimization of the fleet size. Moreover, some side constraints are present, such as limit on journey and trip duration.

An effective iterated local search algorithm, called $\mathcal{A}_{ILS}$ is presented for this problem. The algorithm outperforms results obtained by Battarra et al. [17] on real-world instances.

Further experiments were conducted on Solomon's [196] and Gehring and Homberger's [82] instances for the VRPTW to evaluate the benefit of introducing the multi-trip aspect in fleet sizing problems. The fleet is reduced 3 times out of 56 on Solomon's instances and 126 out of 300 on Gehring and Homberger's instances. Interesting results were obtained, showing that in particular cases (for instances RC1_6_4, R1_8_8 and R1_10_8) less than half vehicles are needed to serve all the customers in a multi-trip context. The detailed analysis of the results shows that in most of the cases the fleet can be reduced because the available time horizon is not well exploited and vehicles are left unused at the depot. Further analysis pointed out that when vehicles are used for a percentage that ranges between 70% and 90% of the working day, benefits of the introduction of the multi-trip possibility are case-dependent. As a consequence, there is a clear need of developing algorithms that can efficiently solve both the VRPTW and the MTVRPTW in order to get the most convenient solution in each case, rather than algorithms dedicated to just one of the two problems.

Future research should fill this gap and be conducted in order to study schemes that can propose competitive solutions for the MTVRP (resp. MTVRPTW), but being able to produce high quality VRP (resp. VRPTW) solutions when performing several trips is not convenient.

Chapter 7

MODUM Vehicle Routing Problem

7.1 Introduction

The MODUM project aims at studying a mutualized delivery system based on a ring of City Distribution Centers (CDC) strategically located in the outskirt of the city. Carriers that need to supply clients, deliver goods to one of the available CDC rather than directly at client location. A CDC is a logistic platform used to transfer goods from one vehicle to another. Usually it is characterized by no stockage or short-storage period.

Different studies and projects have been carried out on the efficiency of such delivery systems. In some cases the results were positive and the perspectives promising, other projects failed. A success depends on several factors and actors. The CDC must be well located: they should be close to the city center, but easy to reach for trucks that usually arrive through highways. The whole system needs to be well dimensioned in order to be able to receive all the goods, but being not too costly. Authorities need to impose and, especially, to enforce limitation on city center access, in order to force trucks to stop at the CDC instead of going up to the clients. On the other side, carriers ask for an efficient system: the goods need to be at customers on time, making carriers saving possibly money and/or time. Delivery to customers should be made by eco-friendly vehicles, that we will call *vans*, in order to reduce air and noise pollution.

The MODUM project intends to evaluate such a delivery system, finding the key aspects that can make the project successful if implemented. Moreover, it aims at developing a decision-support tool based on discrete event simulation, that can help practitioners dimensioning the system regarding the city it would be located.

The designed system operates during the work-day and trucks continuously deliver merchandise to CDC. New requests are integrated into the current planning in an on-line fashion. Unserved requests, or merchandise dropped off at the CDC one day in advance will be used to constitute the initial van routing planning for the day after.

The goal of this chapter is to present the vehicle routing problem that is solved at each step of the simulation tool. A delivery plan is initially made for each working day based on the requests already present in the system. Then, the plan is updated in an on-line fashion each time a new request is revealed. The simulation provides different performance indicators that are used to evaluate the system.

This chapter is organized as follows. In Section 7.2 the MODUM system is informally described. A static version of the MODUM vehicle routing problem is described in Section 7.3, while its on-line counterpart in Section 7.4. The simulator is described in Section 7.5 while Section 7.5 concludes this chapter.

Note that the development of the simulator is out of the scope of this thesis. This development was headed by another partner in the MODUM project. Furthermore, this task was scheduled at the end of the project and it is not finished yet. Also, one important assignment related to the use of the simulator was the design and collection of realistic data, that were managed by a third partner. When writing this thesis, this task is not finished neither.

For these reasons, the algorithms that are presented in this chapter are currently being implemented and included in the simulator. Hence, the algorithms are not evaluated nor used for experimental analysis. The goal of the chapter is rather to present the goal that ultimately guided our work and the thinking about the design of the simulator, to which we largely contributed.

7.2 MODUM system

The general scheme of the MODUM system is depicted in Figure 7.1.

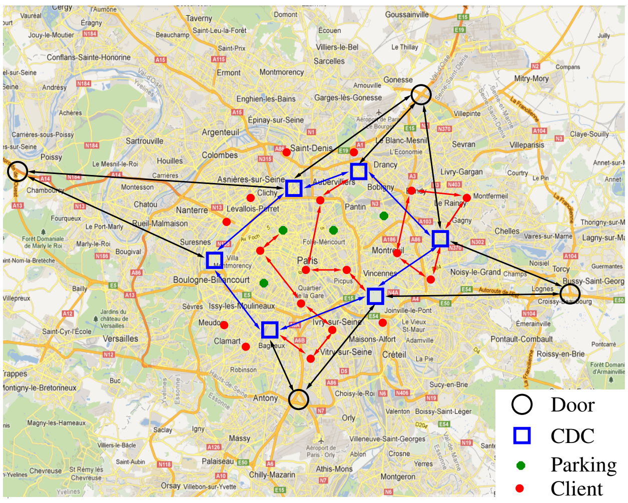


Figure 7.1: MODUM project scheme

It considers both the inbound and the outbound flow of merchandise. We start by describing the first. Trucks deliver merchandise to CDC (blue squares), from where *vans* are used to accomplish final deliveries to customer locations indicated by red spots. Trucks are supposed to enter the system (approach the city) from the *doors* (black circles). A door represents the source of the merchandise and can be an airport, a port, a train station, an industrial zone or a highway junction. The system considers the presence of *parking lots* (green points) that can be used by vans.

All the CDC are linked with a shuttle that circularly and regularly visits them (blue arrows in Figure 7.1) and transfers merchandise from one CDC to another. For example, a truck unloads merchandise at the CDC located at the north of the city center, and goods need to be delivered in the south of the city. It could be convenient to use the shuttle to displace the merchandise to a closer CDC, rather then send a van on the other side of the city.

Once merchandise arrives at the right CDC it is transferred to vans in charge of accomplishing the final delivery to customers. Delivery trips originate at a certain CDC and end at the same CDC, at another CDC or at a parking lot. Clients must be visited during their time windows: waiting at customer location is not allowed. A vehicle can instead go to a parking and wait there if it is beneficial.

Outbound flows work as follows. Goods are picked-up at client locations and sent toward the doors passing through the CDC. Pickup trips always end at a certain CDC, but they can start from the same CDC, another CDC or from a parking lot. In all cases, the van is initially empty. Once the merchandise is at the CDC it can be moved to another more convenient CDC by the shuttle or transferred into a truck and sent toward a door.

We suppose that trips are either pure delivery or pure pickup trips (i.e., only deliveries or only pickups are performed all along the trip). When the distinction does not need to be made, the world *service* will be used to indicate one of the two kind of trips. Each van is assigned to a *journey*, i.e., to a set of service trips that the van itself has to perform in sequence. Trips in the same journey must not overlap in time.

The system considers the possibility of renting free vans. In that case, a user can rent a van, specifying the locations (CDC or parking lot) and the times at which he would like to get and to return back the vehicle. This will be called *free service*.

Operationally, one can suppose that a carrier calls the *operation center* (OPC) and provides the information about which CDC and at what time it will reach it to drop off merchandise. Moreover, it will specify the final destination of the goods. The OPC runs a quick algorithm to evaluate the possibility of integrating the service in the current planning in order to either accept or reject it. If the request is accepted, it has to be inserted in the current delivery plan. All the CDC are considered as potential start points of the delivering trips. If the best CDC is not the one where the truck has arrived, the shuttle is used to move the goods to a more suitable one. This makes the goods available for being loaded in a van and starting the final delivery potentially later than the exact time they entered the system.

A planning is first created before the working day starts with all the known requests. This planning is constructed by solving a static problem, introduced in the following section. During the working day, new requests are inserted in the planning and a dynamic problem is solved.

7.3 MODUM VRP: static version

In this section we formally introduce the static MODUM Vehicle Routing Problem (M-VRP-S). The M-VRP-S can be defined on a complete directed graph $G = (V, A)$, with

$V = I \cup U \cup K$, where

- I is the set of customers (zones to be serviced);
- U is the set of mutualized City Distribution Centers (CDC);
- K is the set of parking lots.

With each arc (i, j) in A is associated a travel distance D_{ij}, and a travel time T_{ij}, indicating the time needed to reach vertex j from vertex i.

Three different requests can be addressed to the OPC: a delivery, a pickup and a free-service request. Let us introduce the three following sets:

- Δ_{del} is the set of requested deliveries;
- Δ_{col} is the set of requested collections;
- Δ_{free} is the set of free-service requests.

Each delivery request $\delta \in \Delta_{del}$ is defined by a vector of dimension 7. In particular

$$\delta = (ori_\delta, R_\delta, dest_\delta, E_\delta, L_\delta, S_\delta, Q_\delta),$$

where

- $ori_\delta \in U$ is the CDC where goods become available for final distribution;
- R_δ is the date goods become available at ori_δ;
- $dest_\delta \in I$ is the customer to which goods need to be delivered;
- E_δ is the earliest moment service can start at $dest_\delta$;
- L_δ is the latest moment service can start at $dest_\delta$;
- S_δ is the service time at customer at $dest_\delta$;
- Q_δ is the load of the requested delivery.

Requested collections $\delta \in \Delta_{col}$ are characterized as well by an analog 7 dimension vector

$$\delta = (ori_\delta, due_\delta, dest_\delta, E_\delta, L_\delta, S_\delta, Q_\delta)$$

where

- $ori_\delta \in I$ represents the customer from where the merchandise has to be picked up;
- due_δ denotes the last possible arrival of goods at a certain CDC;
- $dest_\delta \in U$ is the CDC to which the merchandise needs to be delivered;

- E_δ is the earliest moment service can start at ori_δ;
- L_δ is the latest moment service can start at ori_δ;
- S_δ is the service time at customer at ori_δ;
- Q_δ is the load of the requested to be picked-up.

Waiting is not possible at customer locations. Instead, vans are allowed to go to a parking lot or CDC and wait.

Free-service requests $\delta \in \Delta_{free}$ are represented with a four dimension vector

$$\delta = (ori_\delta, dest_\delta, R_\delta, due_\delta),$$

where

- $ori_\delta \in K \cup U$ is the parking or the CDC at which is picked up the free-service vehicle;
- $dest_\delta \in K \cup U$ is the parking or the CDC at which it has to be left once the travel is achieved;
- R_δ is the time the vehicle needs to be available at ori_δ;
- due_δ specifies the time the vehicle needs to be returned at $dest_\delta$.

A constant time S_{CDC} is needed for loading or unloading operations at each CDC.

A fleet F, constituted by $|F|$ identical vans, is considered. Each van has a capacity Q. At the beginning of each working day, f_i^{init} vans are available at position $i \in U \cup K$. Vans carry out a set of successive trips, where a trip takes one of the following forms:

- a delivery trip originates at a CDC, where the loading of goods requested by the customers is performed; it then visits some customers and terminates at a CDC (not necessarily the initial one) or at a parking lot;
- a collection trip starts either at a CDC or at a parking lot, visits customers and terminates at a certain CDC, where the picked-up goods are unloaded;
- free-service trip.

Note that each vehicle is emptied at the end of a given trip. Moreover, we suppose that vans do not perform pick-ups and deliveries in the same trip. Trips are limited by a maximal traveled distance D_{max}. When vans are electrical vehicles, this constraint models the limited autonomy that this kind of vehicles usually have.

Tables and 7.1 and 7.2 below recapitulate notation.

The M-VRP-S calls for the determination of a set of trips (as previously defined) and an assignment of each trip to a certain van, in order to minimize routing costs and such that:

- all the requests are satisfied;

I	set of customers (zones to be serviced);
U	set of mutualized City Distribution Centers (CDC);
K	set of parking lots involved in the self-service distribution system;
V	vertices of the graph: $V = I \cup U \cup K$;
A	arcs of the graph: $A = V \times V$;
G	graph: $G = (V, A)$;
Δ_{del}	set of requested deliveries;
Δ_{col}	set of requested collections;
Δ_{free}	set of free-service requests;
F	set of vans;
D_{ij}	travel distance between i and j, where $i, j \in V$;
T_{ij}	travel time between i and j, where $i, j \in V$;
S_{CDC}	loading or unloading time at CDC;
Q	vehicle capacity;
D_{max}	maximal traveled distance for a trip;
f_i^{init}	number of vans available at the beginning of the day at location i for $i \in U \cup K$;

Table 7.1: General notation

$\delta = (ori_\delta, R_\delta, dest_\delta, E_\delta, L_\delta, S_\delta, Q_\delta)$	delivery request ($\delta \in \Delta_{del}$); origin (CDC):ori_δ, R_δ; destination (customer): $dest_\delta, E_\delta, L_\delta, S_\delta, Q_\delta$;
$\delta = (ori_\delta, due_\delta, dest_\delta, E_\delta, L_\delta, S_\delta, Q_\delta)$	collection request ($\delta \in \Delta_{col}$); origin (customer): $ori_\delta, E_\delta, L_\delta, S_\delta, Q_\delta$; destination (CDC): $dest_\delta, due_\delta$;
$\delta = (ori_\delta, dest_\delta, R_\delta, due_\delta)$	free-service request ($\delta \in D_{free}$); origin (CDC or parking): ori_δ, R_δ; destination (CDC or parking): $dest_\delta, due_\delta$.

Table 7.2: Notation for requests

- the sum of the demands of customers in delivery and collection trips does not exceed Q;
- the total travelled distance of each collection and delivery trip does not exceed $D_{\max}$;
- trips assigned to the same van do not overlap;
- vans arrive at customers located at $dest_\delta$ not before than E_δ and not later than L_δ;
- free-service trips are characterized by vans that arrive at ori_δ not later than R_δ and at $dest_\delta$ not later than due_δ ;
- requests satisfied by the same collection trip have the same destination CDC $dest_\delta$;
- requests satisfied by the same delivery trip originate at the same CDC ori_δ;
- collection trips do not arrive at the CDC to unload goods later than the minimum due date associated with requests served in the trip;
- delivery trips do not leave from the starting CDC before the maximum release date associated with requests served in the trip;
- vans arrive at the final CDC or parking lot not later than the system closure.

7.4 MODUM VRP: dynamic version

The system MODUM considers dynamic aspects. In particular, not all requests are known at the beginning of the day and some of them occur during operations. The OPC receives these requests and decides whether the system can satisfy them. In addition to previous data, each request is characterized by an arrival time R_δ^{inf}. It indicates the time at which the request becomes known to the OPC, and can be viewed as the time a carrier or a client contacts the OPC to ask for a service.

The OPC should provide a quick answer about the possibility of the system to satisfy the request. In order to achieve this, a quick re-optimization procedure is carried out each time the OPC is called. In particular, one tries to insert the request in the current planning including it in one of the existing trips. The procedure does not consider delivery trips that have already started (that is, loading has started at a CDC) at the time the new information arrives. For collection trips, new requests can be inserted in trips as long as it does not imply a redirection of the vehicle: if the vehicle is heading towards a customer, it is not permitted to insert a new request before visiting the customer itself.

The client receives a positive response if the re-optimization procedure finds a feasible insertion of the request in the planning. In this case the request enters the system and must be satisfied. If the re-optimization procedure returns a negative answer, the request is rejected.

7.5 Insights on the simulator

7.5.1 Client requests and instance requests

The OPC receives solicitations from clients that ask for a service represented by a request $\tilde{\delta}$, that we now call *client* requests. A client request concerns the actual MODUM system or the simulator. The OPC carries out a re-optimization and decides to reject or to accept $\tilde{\delta}$. In the latter case, OPC produces an *instance* requests δ that is considered when solving the vehicle routing problem. Instance requests constitute sets $\Delta_{del} \cup \Delta_{col} \cup \Delta_{free}$ used for the vehicle routing optimization. Client requests represent new requests entered by a user of the system.

Client requests $\tilde{\delta}$ are defined by the same set of characteristics as instance requests δ, but some changes in the values are possible when generating δ from the corresponding $\tilde{\delta}$, as explained in the following. Data for client requests are indexed with *car* so as to distinguish them from data from instance requests. New client requests are managed depending on their type.

7.5.2 Handling of new client requests

Delivery requests

When a client asks for a new delivery request, at time R_{car}^{inf}, he has to select an entering CDC, indicated by ori_{car}, and to give complete information of the request, providing values $(ori_{car}, R_{car}, dest_{car}, E_{car}, L_{car}, S_{car}, Q_{car})$. Because of the shuttle system, the merchandise can be moved to another CDC, indicated by u^*, from which the final delivery will start. Hence u^* can differ from ori_{car}. More specifically, at time R_{car}^{inf}, that is when the client request becomes known, every possible CDC is evaluated as starting point for the final delivery. Depending on the result, u^* is selected among one CDC, or the request $\tilde{\delta}$ is rejected.

The evaluation and selection of u^* is performed as follows. For each CDC $u \in U$, we first evaluate (considering current bookings in the shuttle system) the earliest possible arrival time to CDC u when departing from CDC ori_{car} not earlier than time R_{car}. We then deduce the earliest possible departure time R_u from the CDC u for the sake of final delivery. For each CDC $u \in U$, it thus provides a possible instance request $\delta = (u, R_u, dest_{car}, E_{car}, L_{car}, S_{car}, Q_{car})$. The impact Δ_u on the cost of the solution when introducing the request is quickly computed. The CDC u minimizing Δ_u is selected. In case none of the CDC is convenient, the request is rejected.

Once a CDC is selected, the corresponding booking is carried out in the shuttle and a new instance request δ, defined as $\delta = (u, R_u, dest_{car}, E_{car}, L_{car}, S_{car}, Q_{car})$, is generated.

Collection requests

An equivalent policy is proposed for collection requests. When a client asks for a new collection request, at time R_{car}^{inf}, it has to select any CDC for $dest_{car}$ and to give complete information for the request: $(ori_{car}, due_{car}, dest_{car}, E_{car}, L_{car}, S_{car}, Q_{car})$. Because of the shuttle system, the CDC to which goods will be unloaded once collected is not necessarily $dest_{car}$. At time R^{inf}_{car}, that is when the carrier request becomes known, every possible CDC is evaluated. Depending on the result, either a CDC is selected or the request is rejected.

The evaluation and selection of CDC is calculated as follows. For each CDC $u \in U$, we first evaluate (considering current bookings in the shuttle system) the latest possible time due_u to depart from CDC u and reach CDC $dest_{car}$ on time for an outbound departure at time due_{car}. For each CDC $u \in U$, it thus provides a possible instance request $(ori_{car}, due_u, u, E_{car}, L_{car}, S_{car}, Q_{car})$. The impact Δ_u on the cost of the solution of introducing the request is quickly computed. The CDC u minimizing Δ_u is selected. In case none of the CDC is convenient, the request is rejected.

Once a CDC is selected, the corresponding booking is carried out in the shuttle and a new instance request δ, defined as $\delta = (ori_{car}, due_u, u, E_{car}, L_{car}, S_{car}, Q_{car})$, is generated.

Free-service requests

When a carrier asks for a new free-service, at time R^{inf}_{car}, it selects a set of acceptable parkings to get the vehicle, with associated dates, and a set of acceptable parkings to leave the vehicle, with associated dates. Every combination is evaluated by sending the corresponding instance request to the optimization tool. The best combination is retained and the corresponding instance request is generated. In the case none of the combination is convenient, the request is rejected.

Note that the motivation for generating the instance request at time R^{inf}_{car} is twofold. First, because of the rejection mechanism, it is important to ensure that a feasible solution will exist when a request is accepted. Accepting the request and proceeding to the shuttle booking (and thus selecting the CDC for the instance request) later would be possible but more complex to handle. Second, the sooner the instance request, the more anticipation is possible. The price to pay is a lack of flexibility in the selection of the CDC for the instance request, which might have been postponed after time R^{inf}_{car}.

Note also that because of this policy, moves using the shuttles need not to be considered in the vehicle routing model.

7.5.3 Capacities at CDC

Capacity constraints at CDC are not considered in the vehicle routing model. We sketch here a possible framework for the simulation. Performance indicators could measure the use of available spaces and evaluate the practical feasibility of the system. Three zones would be defined in the CDC: a zone for the transfer among CDC, a storage area and the doors. When arriving at the CDC, goods (for delivery) are located in the transfer area; they are then transferred to their destination CDC. Once arriving at their destination CDC, they are placed in the storage area. Then, they are moved to doors a constant time, called *preparation time*, before the arrival of the vehicle. During this time plus the loading time, the door is booked for the trip. For each of the three zones, a capacity constraint could be defined (maximal quantity of goods for the first two zones, number of doors for the last one).

7.5.4 Day structure

For each day H, the OPC is open, i.e., can be contacted, during a given time window and manages all the new requests $\tilde{\delta}$. Clients can contact the OPC during day H asking for a service in day $H+1$. The solution that is constructed, then covers days H and $H+1$, i.e., for each van, a two-days journey is determined. In particular, each van terminates its journey of day H at a certain location that will be the starting location of the first trip in journey $H+1$. During the opening hours of the OPC, the online algorithm runs in order to manage the new requests.

At the end of day H, i.e., when the OPC closes, the off-line algorithm is launched. It optimizes the planning provided by the on-line algorithm for following day $H+1$.

The choice of working with a two-days planning is useful in balancing the system. In particular, vans terminates their journey on day H based on requests of day $H+1$. Hence, there is no need of an explicit repositioning strategy.

7.5.5 Result analysis

The system would be evaluated analyzing its behavior under different scenarios, characterized, for example, by a different number of CDC, by CDC with different locations, or by different request acceptance policies; the type and the size of the used vans or the transit frequency of the shuttle; the number of the parking lots and their location. The simulator will generate different indicators that describe the behavior of the system under each scenario. Analysis will be conducted comparing these indicators.

Such indicators can be the total cost of the system, the amount of CO_2 emissions, the number of violated time windows, the total tardiness (measured as the difference between the late arrival at client location and the time window closing time), the number of used vans, the average loading factor.

The analysis of the system will be done on realistic data (extracted by a project partner) representative of the flows of merchandise in a medium-size city, as could be Lyon (France), with about 800 customers involved.

7.6 Conclusions and perspectives

In this chapter we described the mutualized delivery system proposed in MODUM and we formally introduced the rich routing problem that arises in this context. We gave an insight of the simulator that is currently under development and that will be used to evaluate the performances of such a system through the indicators that it will provide.

Once the simulator is finalized, different scenarios will be evaluated and the analysis conducted on the results that are obtained.

Chapter 8

Conclusions and perspectives

City logistics has become an important research topic in the last few years. Efficient delivery systems are studied and new solutions are sought. Solutions that look for achieving economic savings as well as respecting the environment, and creating pleasant cities by revitalizing downtowns. In this context the MODUM project studies a new and efficient delivery system, based on a ring of City Distribution Centers (CDC) located around the city.

A first contribution of the thesis is to analyze the movements of merchandise in the urban context. The investigation highlights how urban good movements are performed by delivery trips with several customer visits. This leaves space for the optimization of the planning that can result in travelling time, travelling distance and pollution reductions.

Nevertheless, when dealing with urban systems, such an efficient delivery planning is not easy to achieve. The metropolitan environment has peculiar characteristics as rush hours, traffic jams, restriction policies, or other characteristics that are enhanced in this context, as car accidents, that deeply affect the routing design and need to be taken into account. Effective operations pass across the understanding of the urban environment.

This thesis contributes to this essential task with the analysis of works done by researchers in the context of urban logistic and with their collection into a survey. Furthermore, from these works we extrapolate the main characteristics that a routing problem should take into account to produce an efficient delivery planning. We considered each characteristic in the survey and proposed the related work. This gives an insight on the subject. The goal of the work is to provide to future researchers a document that provides an overview on routing in cities, and guides the reader toward further and more specific in-depth analysis.

One particular characteristic we detected in the previous phase, is the *multi-trip* aspect. Delivery trips in city centers are often made by small and eco-friendly vehicles, called *vans*. Autonomy or capacity limitation limits the length of the trip that, therefore, normally takes less time than the working day. Vehicles can then be re-used several times, in order to exploit the whole time horizon. The academic problem that arises is the Multi-Trip Vehicle Routing Problem (MTVRP).

This *modus operandi* is common to city logistics systems: structural arrangement of cities (due to medieval inheritance) and ecological motivations force practitioners to take into account vans for final deliveries. However, the urban context is not the sole in which multiple trips per day are considered. Supermarket supply, petrol station replenishment, garbage collection and livestock collection are few examples of routing applications in which the multi-trip aspect has been considered. Moreover, recent works admit multiple trips per vehicle in production scheduling problems and inventory routing problems.

Despite its practical interest, the MTVRP has not been intensively studied by researchers and the literature is not as large as it probably would deserve. This thesis contributes in filling this gap and proposes heuristic solution methods for the MTVRP, for the MTVRP with Time Windows and Release Dates, and for the Multi Commodity MTVRP with Time Windows. An additional contribution of this thesis in the multi-trip context is the gathering of all the works done on the subject into a survey. It is the first complete survey on the MTVRP. It shows the limit of exact algorithms, compares results and gives references to classical benchmark of instances. Moreover, a section examines the works with a practical interest or motivation. This part of the work should encourage other researchers to put their efforts in this particular area of routing problems.

As already mentioned, we developed three algorithms for multi-trip routing problems. We first developed a population-based algorithm for the MTVRP where chromosomes are permutations of the customers, usually called *giant-tour*. A split procedure (based on the work of Prins [166]) turns chromosomes into solutions. Moreover, we proposed a tailored local search operator. To the best of our knowledge, to date, only operators peculiar to the VRP has been used in the multi-trip context, with the only exception of basic relocations and swaps of trips among vehicles. The operator we proposed combines classical moves for the VRP that do not improve the current solution, and the swap of trips among different vehicles, looking for a global improvement. This is the first step in developing dedicated operators that exploit the particular structure of the problem. The results obtained are the state-of-the-art on the classical benchmark of instances.

A second work introduced a new problem: the MTVRP with Time Windows and Release Dates (MTVRPTWR). A release date is associated with each merchandise and it represents the instant the merchandise itself becomes available for delivery at the depot. It models the dependence of *external* and *internal* flows. External flows are trips made by heavy trucks that bring (resp. pick up) merchandise to (resp. at) a CDC, while internal flows are trips made by vans from a CDC to the customers or vice-versa. Conversations with practitioners from private companies certified the practical interest of the problem, as well as the absence of academic work on the subject.

We proposed a hybrid genetic algorithm for the MTVRPTWR. Giant-tours are turned into solutions by means of a labelling procedure that generalizes the split procedure that we introduced for the MTVRP. We created a set of instances for benchmark purposes. We used the well-known Solomon's instances as base instances. In particular, the same release date is possibly associated with different customers: this simulates the arrival of a truck at the depot. Different families of instances are introduced, each characterized by a different average tightness of the release date with respect to the time window upper bound.

Contextually with the introduction of the MTVRPTWR, several directions are now opened and left for future works. One is the development of exact methods. Even if exact methods are unlikely to be efficient in solving real-size problems, they can be used to evaluate heuristic algorithms. A second and natural direction is the consideration of dynamic aspects. The merchandise arrives at the depot through highway axes and urban streets. Hence trucks can suffer delays due to unfavorable traffic conditions. Drivers can communicate their delay to the operation center that can modify the release date and update the planning accordingly. Dynamic aspects can be introduced even considering the possibility that some trucks can notify their arrival during the working day. In both cases, the use of new communicating technologies makes possible the link between truck drivers and the operation center that coordinates the delivery operations. A third sub-class of problems that can derive from the MTVRPTWR considers stochastic aspects. Stochastic travel times can be considered at the internal flow level as well as at the external flow level. In the latter case, uncertainty in travel times of roads traversed by trucks can be reflected into the release dates.

The last variant of the MTVRP that we treated into the thesis, organizes the delivery planning of incompatible commodities, namely, that cannot be transported in the same vehicle at the same time. It has been called the Multi Commodity MTVRP with Time Windows and first introduced by Battarra et al. [17]. The problem is strategical rather than operational and the number of routed vehicles needs to be minimized. An Iterated Local Search (ILS) is proposed for a set of instances arose in a real context. Results are the new state-of-the-art for the problem: the fleet is reduced on all the instances with up to 10%.

In addition, an analysis is conducted on the potential benefit of allowing vehicles to perform several trips and its impact in the fleet sizing. Since the Multi Commodity MTVRP with Time Windows generalizes the MTVRP with Time Windows, we run our ILS on the classic Solomon's and Gehring and Homberger's instances designed for the VRPTW and with the objective of minimizing the fleet first and the traveled distance second. Results showed that in some cases the fleet can be halved only letting vehicles to perform several trips, while in others it is unlikely that the introduction of the multi-trip aspect can reduce the number of used vehicles. Further analysis showed that the reason of the possible size reduction is that the time horizon is not well exploited and vehicles are left unused at the depot.

The contribution of this part of the work is not only from an algorithmic point of view, with the development of an efficient procedure. The results obtained and their analysis should make the reader aware of the importance of the multi-trip aspect in achieving efficient delivery planning. Here, with the world efficient we do not only refer to the operational level, but to the strategical level as well, where we look for the minimization of the fleet size and the related costs.

Research directions that arise from this work and the analysis of the related results we obtained is the interest in designing algorithms that efficiently solve both a VRP problem and its multi-trip counterpart. The current common procedure is to evaluate algorithms for multi-trip routing problems on tailored instances where the limited capacity of vehicles forces them to perform several trips. Methods are not evaluated on instances that admit an optimal VRP solution. Consequently, their ability of finding VRP solutions is not evaluated. The analysis we conducted, showed that in some situations a MTVRP solution is required to exploit efficiently the time horizon, in other cases a VRP solution is sufficient. From this, the need of efficient algorithms for both problems.

The thesis terminates with a formal definition of the rich routing problem that arises in the MODUM project and guided our work. The problem involves several depots and parking lots used to park vans. Vans make several trips during the working day and make both pickup and delivery services (although in separate trips). Trips start at a certain depot and terminate at the same or at another depot or even at a parking lot. Release dates are associated with each merchandise to deliver, while due dates are associated with each merchandise to pick up at customers.

As the design of the MODUM system inspired the introduction of the routing problems with release dates, such systems can be the motivation for the introduction of routing problem with due dates associated with the merchandise that needs to be picked-up at customers and brought at the depot. While routing problems with release dates model pure delivery problems, problems with due dates would model pure pickup problems and define another practical interesting class of routing problems. A natural extension would then be the consideration of both release and due dates into the same problem with the consequently introduction of a third class as the intersection of the two.

At the time of concluding this thesis, the simulator in charge of evaluating the system was under development. The objective is to evaluate the system operating in different scenarios. The evaluation will be based on different indicators as the cost of the system, the amount of CO_2 emissions, the number of violated time windows, the number of used vans, the average loading factor. We did not consider time-dependent travel times, that usually characterizes city centers and urban zones. Constant travel times are then used for a first evaluation of the system. However, further research should consider time-dependent travel times as well as stochastic travel times for a better representation of the metropolitan environment. Furthermore, the MODUM system considers a shuttle that links all the depots and moves merchandise from one depot to another one, for example, more suitable for the final delivery. Future research can study the organization of the shuttle evaluating the possibility of optimizing the schedule based on the exact demand rather than having a pre-scheduled service. Each of the directions we listed can improve the system considered in MODUM and enhance its functionality. We are optimistic that the simulation will demonstrate the efficiency of the system on condition that the right tactical and strategical choices have been made. Future work should be done into the amelioration of these systems that we strongly believe are a concrete possibility for the construction of livable and people oriented cities.

List of Figures

1 Exemple sur une petite instance avec cinq clients 6

2 Exemple de graphe auxiliaire 7

3 Tournées pour le cinquième jour: différentes couleurs représentent différents produits 8

4 Schéma étudié dans le projet MODUM 9

2.1 UGM classification 30

2.2 Time-dependent travel time $\tau_{ij}(t)$ for an edge (i, j). Time and traveling time in hours 39

2.3 General Two-Level distribution system 42

2.4 Two-Level Problems 43

3.1 An example in which the optimal MTVRP solution costs less than the optimal VRP solution 59

3.2 Algorithm comparison on benchmark instances 62

3.3 Initial configuration 62

3.4 Pejorative move. In bold trips involved in m 62

3.5 After re-packing the trips. In bold trips involved in re-packing phase 63

4.1 Example with 5 customers: demands in brackets, $T_H = 45$, $Q = 50$, $\theta = \lambda = 2$ 86

4.2 Auxiliary graph (to each arc (i, j), cost c_{ij} is assigned as defined in Equation 4.2) 86

4.3 MTVRP solution obtained from arcs forming the shortest path ($F(\Psi) = 288$) 87

4.4 Best MTVRP solution for Ψ ($F(\Psi) = 273$) 88

4.5 MTVRP solution obtained considering rotations of customers in the same trip ($F(\Psi) = 241$) 88

4.6 OX operator 90

4.7 Initial configuration 93

4.8 Pejorative move. In bold trips involved in M 93
4.9 After *Swp*. In bold trips involved in *Swp* 93
4.10 Notation 93

5.1 The last two trips arrive at the depot before they have left it 120
5.2 The auxiliary graph for chromosome $\Psi = (1, 2, 3, 4, 5)$, data as in Figure 5.1, $Q = 60$ and $\theta = \lambda = 2$ 122
5.3 Result analysis 132

6.1 Ordered Crossover 149
6.2 Auxiliary graph H 150
6.3 Multi-trips of day 5 solution: different color represents a different commodity 155
6.4 Instances analysis fleet reduction on time usage 164

7.1 MODUM project scheme 168

List of Tables

2.1 Road occupancy rates . . . 32
2.2 Percentage of deliveries with respect to the total number of deliveries . . . 33
2.3 Percentage of routes with respect to the total number of routes . . . 34
2.4 Percentage of weight delivered with respect to the total delivered weight . . . 34
2.5 Articles cited that concern TDVRP . . . 40
2.6 Articles cited that concern multi-level distribution . . . 45
2.7 Articles cited that concern DVRP . . . 48
2.8 Articles cited that concern MTVRP . . . 51

3.1 Instances' details of Taillard et al. [198] . . . 60
3.2 Results on MTVRP instances . . . 61
3.3 Results on MTVRPTW instances by Hernandez et al. [105] . . . 66
3.4 Comparison of Hernandez et al. [106] and Macedo et al. [136] on instances with short t_{max} and non-truncated distances . . . 68
3.5 Comparison of Hernandez et al. [106] and Macedo et al. [136] on instances with long t_{max} and non-truncated distances . . . 69
3.6 Comparison of Hernandez et al. [106] and Azi et al. [10] on instances with short t_{max} and truncated distances . . . 70
3.7 Comparison of Hernandez et al. [106] and Azi et al. [10] on instances with long t_{max} and truncated distances . . . 71
3.8 Results on Azi et al. [14] instances . . . 73

4.1 Labels associated with nodes of H during the assignment procedure . . . 88
4.2 Instances' details . . . 96
4.3 Parameter Tuning . . . 97
4.4 Notation for computational results . . . 98

4.5 Feasibility check on the 42 instances in G1 99
4.6 Feasibility check on the 56 instances in G2 100
4.7 Feasible solutions on the 42 instances in G1 102
4.8 Feasible solutions on the 56 instances in G2 103
4.9 Feasible solutions on the 6 instances in G3 104
4.10 Non-feasible solutions on the 5 unsolved instances in G3 105
4.11 Comparison with OV on the 42 instances in G1 107
4.12 Comparison with OV on the 56 instances in G2 108
4.13 Machines used in previous papers 109
4.14 CPU times comparison. Times expressed in seconds 109

5.1 Labels generated by *AdSplit* 125
5.2 Instance details 130
5.3 Parameter Tuning 131
5.4 Setting $\mathcal{L}_{\text{threshold}}$ 131
5.5 Results on new instances, $\kappa = 0$ and $\kappa = 0.25$ 133
5.6 Results on new instances, $\kappa = 0.5$ and $\kappa = 0.75$ 134
5.7 Statistics on new instances 135
5.8 Results on Hernandez et al. [105] instances with $N = 25$ and $M = 2$ 137
5.9 Results on Hernandez et al. [105] instances with $N = 50$ and $M = 4$ 138
5.10 Results on instances with $N = 100$ and $M = 8$ created as in Hernandez et al. [105] 139

6.1 Comparison of the procedures for the creation of the initial permutation . . . 147
6.2 Data for H in Figure 6.2 151
6.3 Statistics on Battarra et al. [17] instances. Values marked with an asterisk are corrected with respect to the original paper. 154
6.4 Results on Battarra et al. [17] instances 154
6.5 Results on Solomon's instances 156
6.6 Results on Gehring and Homberger's instances with 200 customers 157
6.7 Results on Gehring and Homberger's instances with 400 customers 158
6.8 Results on Gehring and Homberger's instances with 600 customers 159
6.9 Results on Gehring and Homberger's instances with 800 customers 160

6.10 Results on Gehring and Homberger's instances with 1000 customers 161

6.11 Comparison of VRPTW and MTVRPTW solutions for R1_2_8 Gehring and Homberger instance 163

7.1 General notation 172

7.2 Notation for requests 172

A.1 New feasible solution for CMT4_T_H^1_7 xiii

B.1 Solution on C103 Solomon's instances that makes use of 9 vehicles xv

B.2 Solution on C104 Solomon's instances that makes use of 9 vehicles xvi

B.3 Solution on C109 Solomon's instances that makes use of 9 vehicles xvi

List of Algorithms

1 Memetic Algorithm outline 85
2 *AdSplit* 89
3 Combined LS 96

4 $\mathcal{A}^{CAF}$ outline 117
5 Instance creation 128

6 General Iterated Local Search scheme 144
7 $\mathcal{A}_{ILS}$ 145

Appendix A

New feasible solution for CMT$4_T_H^1_7$

The procedure found a new feasible solution for problem CMT4_T_H^1_7 that is detailed in Table A.1 where v, r, τ_r and l_r indicate the vehicle, the route, its travelling time and its load.

v	r	τ_r	l_r	
1	1	152.00	195	0, 18, 60, 84, 114, 8, 46, 124, 47, 36, 143, 49, 64, 11, 126, 63, 90, 70, 101, 69, 0
2	1	150.42	200	0, 51, 103, 71, 65, 136, 35, 135, 34, 78, 121, 29, 24, 134, 25, 55, 130, 54, 0
3	1	97.33	200	0, 40, 73, 75, 56, 23, 67, 39, 139, 4, 110, 149, 26, 0
	2	55.68	174	0, 53, 138, 12, 109, 80, 150, 68, 116, 76, 111, 27, 0
4	1	73.68	196	0, 50, 102, 33, 81, 9, 120, 129, 79, 3, 77, 28, 0
	2	80.10	198	0, 146, 52, 106, 7, 82, 48, 123, 19, 107, 62, 148, 88, 127, 0
5	1	56.16	187	0, 96, 104, 99, 93, 85, 61, 5, 118, 89, 0
	2	95.96	199	0, 132, 1, 122, 30, 20, 66, 128, 131, 32, 108, 10, 31, 0
6	1	89.36	200	0, 59, 98, 91, 16, 141, 86, 113, 17, 45, 125, 83, 0
	2	64.60	156	0, 105, 21, 72, 74, 133, 22, 41, 145, 115, 2, 58, 0
7	1	36.35	130	0, 112, 147, 6, 94, 95, 117, 13, 0
	2	116.94	200	0, 137, 87, 144, 57, 15, 43, 42, 142, 14, 38, 140, 44, 119, 100, 37, 92, 97, 0

Table A.1: New feasible solution for CMT4_T_H^1_7

Appendix B

Detailed results on Solomon instances

Tables B.1–B.3 report the detailed solutions on Solomon's instances that use 9 vehicles when multi usage of vehicles is allowed (instead of 10 vehicles as in the best known VRP solution).

	C103
vehicle	trip
1	0 65 63 62 74 0
	0 84 97 100 99 96 95 98 0
2	0 7 12 14 16 15 19 18 34 29 24 0
3	0 13 17 8 10 11 9 6 4 2 1 3 5 0
4	0 67 55 54 53 56 58 64 68 66 69 61 72 0
5	0 81 78 76 71 70 73 77 79 80 82 83 90 0
6	0 20 25 27 37 38 39 36 52 50 49 47 0
7	0 32 33 31 35 30 28 26 23 22 21 0
8	0 43 42 41 40 44 46 45 48 51 60 59 57 0
9	0 87 86 85 92 93 94 88 89 91 75 0
dist	1072.32

Table B.1: Solution on C103 Solomon's instances that makes use of 9 vehicles

	C104
vehicle	trip
1	0 67 63 62 74 0 0 21 22 26 23 29 33 32 0
2	0 13 17 18 19 15 16 14 12 20 0
3	0 28 30 34 36 39 38 37 35 31 52 49 47 0
4	0 41 40 59 53 56 58 60 54 64 69 65 0
5	0 98 94 92 93 97 100 99 96 95 5 0
6	0 81 78 76 71 70 73 77 79 80 72 61 66 0
7	0 90 87 86 83 82 84 85 88 89 91 3 7 0
8	0 68 55 57 44 45 48 51 50 46 42 43 0
9	0 24 27 25 8 10 11 9 6 4 2 1 75 0
dist	937.65

Table B.2: Solution on C104 Solomon's instances that makes use of 9 vehicles

	C109
vehicle	trip
1	0 57 55 54 53 56 58 60 59 0 0 26 23 22 21 0
2	0 43 42 41 0 0 67 62 74 72 61 64 68 66 69 0
3	0 25 27 32 31 40 44 46 45 48 51 50 47 0
4	0 13 17 18 19 15 16 14 12 0
5	0 20 24 29 33 35 37 38 39 36 34 52 49 0
6	0 3 98 96 95 94 92 93 97 100 99 0
7	0 63 81 78 76 71 70 73 77 79 80 0
8	0 65 90 87 86 83 82 84 85 88 89 91 75 0
9	0 5 7 8 10 28 30 11 9 6 4 2 1 0
dist	958.11

Table B.3: Solution on C109 Solomon's instances that makes use of 9 vehicles

The first row indicates the instance name, while the last row reports the travelled distance. Columns *vehicle* report the vehicle index, while columns *trip* detail the trips.

Bibliography

[1] E.-H. Aghezzaf, B. Raa, and H. Van Landeghem. Modeling inventory routing problems in supply chains of high consumption products. *European Journal of Operational Research*, 169(3):1048–1063, 2006.

[2] B.H. Ahn and J.Y. Shinm. Vehicle-routing with time windows and time-varying congestion. *Journal of Operational Research Society*, 42(5):393–400, 1991.

[3] F. Alonso, M.J. Alvarez, and J.E. Beasley. A tabu search algorithm for the periodic vehicle routing problem with multiple vehicle trips and accessibility restrictions. *Journal of Operational Research Society*, 59(7):963–976, 2008.

[4] C. Ambrosini and J.L. Routhier. Objectives, methods and results of surveys carried out in the field of urban freight transport: an international comparison. *Transport Reviews*, 24(1):57–77, 2004.

[5] S. Anderson, J. Allen, and M. Browne. Urban logistics - how can it meet policy makers' sustainability objectives? *Journal of Transport Geography*, 13(1):71–81, 2005.

[6] N. Ando and E. Taniguchi. Travel time reliability in vehicle routing and scheduling with time windows. *Networks and Spatial Economics*, 6(3):293–311, 2006.

[7] C. Archetti, A.M. Campbell, and M.G. Speranza. Multi-commodity vs. single-commodity routing. Technical report, Università di Brescia, 2013.

[8] A. Attanasio, J.F. Cordeau, G. Ghiani, and G. Laporte. Parallel tabu search heuristic for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing*, 30(3):377–387, 2004.

[9] P. Avella, M. Boccia, and A. Sforza. Solving a fuel delivery problem by heuristic and exact approaches. *European Journal of Operational Research*, 152(1):170–179, 2004.

[10] N. Azi. *Méthodes Exactes et Heuristiques pour le Problème de Tournées avec Fenêtres de Temps et Réutilisation de Véhicules*. PhD thesis, Université de Montréal, 2010.

[11] N. Azi, M. Gendreau, and J.-Y. Potvin. An exact algorithm for a single-vehicle routing problem with time windows and multiple routes. *European Journal of Operational Research*, 178(3):755–766, 2007.

[12] N. Azi, M. Gendreau, and J.-Y. Potvin. An exact algorithm for a vehicle routing problem with time windows and multiple routes. *European Journal of Operational Research*, 202(3):756–763, 2010.

[13] N. Azi, M. Gendreau, and J.-Y. Potvin. A dynamic vehicle routing problem with multiple delivery routes. *Annals of Operations Research*, 199(1):103–112, 2012.

[14] N. Azi, M. Gendreau, and J.-Y. Potvin. An adaptive large neighborhood search for a vehicle routing problem with multiple routes. *Computers & Operations Research*, 41:167–173, 2014.

[15] R. Baldacci, A. Mingozzi, R. Roberti, and R. Wolfler Calvo. An exact algorithm for the two-echelon capacitated vehicle routing problem. *Operations Research*, 61(2):298–314, 2013.

[16] M. Barkaoui and M. Gendreau. An adaptive evolutionary approach for the real-time vehicle routing and dispatching. *Computers & Operations Research*, 40(7):1766–1776, 2013.

[17] M. Battarra, M. Monaci, and D. Vigo. An adaptive guidance approach for the heuristic solution of a minimum multiple trip vehicle routing problem. *Computers & Operations Research*, 36(11):3041–3050, 2009.

[18] H.B. Bendall and A.F. Stent. A scheduling model for a high speed containership service: A hub and spoke short-sea application. *International Journal of Maritime Economics*, 3:262–277, 2001.

[19] R. Bent and P. Van Hentenryck. Waiting and relocation strategies in online stochastic vehicle routing. In *Proceedings of the 20th international joint conference on Artificial intelligence*, pages 3041–3050, 2007.

[20] G. Berbeglia, J.F. Cordeau, and G. Laporte. Dynamic pickup and delivery problems. *European Journal of Operational Research*, 202(1):8–15, 2010.

[21] L. Bertazzi, M. Savelsbergh, and M.G. Speranza. In *The Vehicle Routing Problem - Last Advances and New Challenges*, Operations Research Computer Science Interfaces, chapter Inventory Routing, pages 49–72. Springer, 2008.

[22] J. Błażewicz, K. Ecker, E. Pesch, G. Schmidt, and J. Węglarz. In *Handbook on Scheduling - From Theory to Application*, International Handbook on Information Systems, chapter Scheduling on One Processor, pages 73–136. Springer, 2007.

[23] A. Bonnafous. Les merchandises en ville: le problème méthodologique de l'apréhension statistique. In *L'intégration des marchandises dans le système des déplacements urbains*, Etudes & Recherches, pages 86–91. Laboratoire d'economie des transports, Lyon, France, 2001.

[24] J. Brandão and A. Mercer. A tabu search algorithm for the multi-trip vehicle routing problem. *European Journal of Operational Research*, 100(1):180–191, 1997.

[25] J. Brandão and A. Mercer. The multi-trip vehicle routing problem. *Journal of the Operational Research Society*, 49:799–805, 1998.

[26] J. Branke, M. Middendorf, G. Noeth, and M. Dessouky. Waiting strategies for dynamic vehicle routing. *Transportation Science*, 39(3):298–312, 2005.

[27] O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, part I: Route construction and local search algorithms. *Transportation Science*, 39(1):104–118, 2005.

[28] O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, part II: Metaheuristics. *Transportation Science*, 39(1):119–139, 2005.

[29] M. Browne, J. Allen, and J. Leonardi. Evaluating the use of an urban consolidation centre and electric vehicles in central london. *IATSS Research*, 35(1):1–6, 2011.

[30] M. Browne, J. Allen, T. Nemoto, and D. Patier. Reducing social and environmental impacts of urban freight transport: A review of some major cities. *Procedia - Social and Behavioral Sciences*, 39:19–33, 2012.

[31] D. Cattaruzza, N. Absi, and D. Feillet. The multi-trip vehicle routing problem with time windows and release dates. Technical Report 2014 - 01, Ecole des Mines de Saint-Etienne, 2014.

[32] D. Cattaruzza, N. Absi, D. Feillet, O. Guyon, and X. Libeaut. The multi-trip vehicle routing problem with time windows and release dates. In *MIC 2013*. Singapore, August 2013.

[33] D. Cattaruzza, N. Absi, D. Feillet, and T. Vidal. A memetic algorithm for the multi trip vehicle routing problem. *European Journal of Operational Research*, 236, 2014.

[34] D. Cattaruzza, N. Absi, D. Feillet, and D. Vigo. An iterated local search for the multi-commodity multi-trip vehicle routing problem with time windows. Technical Report 2014 - 02, Ecole des Mines de Saint-Etienne, 2014.

[35] T.-C. Chang and C.-Y. Lee. Machine scheduling with job delivery coordination. *European Journal of Operational Research*, 158(2):470–487, 2004.

[36] T.S. Chang and H.M. Yen. City-courier routing and scheduling problems. *European Journal of Operational Research*, 223(2):489–498, 2012.

[37] Z.-L. Chen. Integrated production and outbound distribution scheduling: Review and extensions. *Operations Research*, 58(1):130–148, 2010.

[38] Z.-L. Chen and H. Xu. Dynamic column generation for dynamic vehicle routing with time windows. *Transportation Science*, 40(1):74–88, 2006.

[39] M. Christiansen, K. Fagerholt, B. Nygreen, and D. Ronen. In *Handbooks in Operations Research & Management Science*, chapter Maritime Transportation, pages 189–284. North Holland, 2007.

[40] M. Christiansen, K. Fagerholt, and D. Ronen. Ship routing and scheduling: Status and perspectives. *Transportation Science*, 38(1):1–18, 2004.

[41] N. Christofides, A Mingozzi, and P. Toth. *The vehicle routing problem*, pages 315–338. Wiley, Chichester, 1979.

[42] G. Clarke and J.W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581, 1964.

[43] L.C. Coelho, J.-F. Cordeau, and G. Laporte. Thirty years of inventory routing. *Transportation Science*, 48(1):1–19, 2014.

[44] R.G. Conrad and M.A. Figliozzi. Algorithms to quantify impact of congestion on time-dependent real-world urban freight distribution networks. *Transportation Research Record: Journal of the Transportation Research Board*, 2168, 2010.

[45] F. Cornillier, F.F. Boctor, G. Laporte, and J. Renaud. A heuristic for the multi-period petrol station replenishment problem. *European Journal of Operational Research*, 191:295–305, 2008.

[46] F. Cornillier, F.F. Boctor, G. Laporte, and J. Renaud. An exact algorithm for the petrol station replenishment problem. *Journal of Operational Research Society*, 59:607–615, 2009.

[47] F. Cornillier, G. Laporte, F.F. Boctor, and J. Renaud. The petrol station replenishment problem with time windows. *Computers & Operations Research*, 36(3):919–935, 2009.

[48] G. Cornillier, F.F. Boctor, and J. Renaud. Heuristic for the multi-depot petrol station replenishment problem with time windows. *European Journal of Operational Research*, 220(2):361–369, 2012.

[49] T.G. Crainic. *City Logistics*, pages 181–212. Tutorials in Operations Research 2008 - State-of-the-Art Decision Making Tools in the Information-Intensive Age. INFORMS, 2008.

[50] T.G. Crainic, F. Errico, W. Rei, and N. Ricciardi. Integrating c2e and c2c traffic into city logistics planning. *Procedia - Social and Behavioral Sciences*, 39:47–60, 2011.

[51] T.G. Crainic, Y. Gajapl, and M. Gendreau. Multi-zone multi-trip vehicle routing problem with time windows. Technical Report 2012 - 36, Centre interuniversitaire de recherche sur les réseaux d'enterprise, la logistique et le transport, Université de Montréal, Montréal, QC, Canada , 2012.

[52] T.G. Crainic, S. Mancini, G. Perboli, and R. Tadei. Clustering-based heuristic for the two-echelon vehicle routing problem. Technical Report 2008 - 46, Centre interuniversitaire de recherche sur les réseaux d'enterprise, la logistique et le transport, Université de Montréal, Montréal, QC, Canada , 2008.

[53] T.G. Crainic, S. Mancini, G. Perboli, and R. Tadei. A grasp with path-relinking metaheuristic for the two-echelon vehicle routing problem. In *MIC 2011*, pages 251–258. Udine, Italy, July 2011.

[54] T.G. Crainic, G. Perboli, S. Mancini, and R. Tadei. Two-echelon vehicle routing problem: A satellite location analysis. *Procedia - Social and Behavioral Sciences*, 2(3):5944–5955, 2010.

[55] T.G. Crainic, N. Ricciardi, and G. Storchi. Advanced freight transportation systems for congested urban areas. *Transportation Research Part C*, 12(2):119–137, 2004.

[56] T.G. Crainic, N. Ricciardi, and G Storchi. Models for evaluating and planning city logistics systems. *Transportation Science*, 43(4):432–454, 2009.

[57] A. Şen and K. Bülbül. A survey on multi trip vehicle routing problem. In *VI International Logistics & Supply Chain Congress*. Instanbul, 2008.

[58] G.B. Dantzig and J.H Ramser. The truck dispatching problem. *Management Science*, 6:80–91, 1959.

[59] A. Del Pia and C. Filippi. A variable neighborhood descent algorithm for a real waste collection problem with mobile depots. *International Transactions in Operational Research*, 13(2):125–141, 2006.

[60] L. Delaître and C. De Barbeyrac. Improving an urban distribution centre, the french case of samada monoprix. *Procedia - Social and Behavioral Sciences*, 39:753–769, 2012.

[61] A.V. Donati, R. Montemanni, N. Casagrande, A.E. Rizzoli, and L.M. Gambardella. Time dependent vehicle routing problem with a multi ant colony system. *European Journal of Operational Research*, 185(3):1174–1191, 2008.

[62] M. Drexl. Synchronization in vehicle routing – a survey of vrps with multiple synchronization constraints. *Transportation Science*, 46(3):297–316, 2012.

[63] M. Dror, editor. *Arc Routing: Theory, Solutions and Applications*. Kluwer Academic Publisher, 2000.

[64] B. Durand and J. Gonzalez-Feliu. Urban logistics and e-grocery: have proximity delivery services a positive impact on shopping trips? *Procedia - Social and Behavioral Science*, 39:510–520, 2012.

[65] D. Egger and Ruesch M. Best practice handbook year 3 - road pricing and urban freight transport urban freight platforms. Technical report, BESTUFS, 2002.

[66] J.F. Ehmke. Analysis of floating car data. In *Integration of Information and Optimization Models for Routing in City Logistics*, International Series in Operations Research & Management Science, pages 59–78. Springer, 2012.

[67] J.F. Ehmke. Provision of distance matrices. In *Integration of Information and Optimization Models for Routing in City Logistics*, International Series in Operations Research & Management Science, pages 83–104. Springer, 2012.

[68] J.F. Ehmke and D.C. Mattfeld. Data allocation and application for time-dependent vehicle routing in city logistics. *European Transport/Trasporti Europei*, 46:24–35, 2010.

[69] J.F. Ehmke and D.C. Mattfeld. Integration of information and optimization models for vehicle routing in urban areas. *Procedia - Social and Behavioral Sciences*, 20:110–119, 2011.

[70] J.F. Ehmke, S. Meisel, and D.C. Mattfeld. Floating car based travel times for city logistics. *Transportation Research Part C*, 21(1):338–352, 2012.

[71] J.F. Ehmke, A. Steinert, and D.C. Mattfeld. Advanced routing for city logistics service providers based on time-dependent travel times. *Journal of Computational Science*, 3(4):193–205, 2012.

[72] K. Fagerholt. Optimal fleet design in a ship routing problem. *International Transaction in Operational Research*, 9(5):453–464, 1999.

[73] K. Fagerholt and H. Lindstad. Optimal policies for maintaining a supply service in the norwegian sea. *Omega*, 28(3):269–275, 2000.

[74] M.A. Figliozzi. A route improvement algorithm for the vehicle routing problem with time dependent travel times. In *TRB 88th Annual Meeting Compendium of Papers DVD*, 2009.

[75] M.A. Figliozzi. The impacts of congestion on time-definitive urban freight distribution networks co_2 emission levels: Results from a case study in portland, oregon. *Transportation Research Part C*, 19(5):766–778, 2011.

[76] M.A. Figliozzi. The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics. *Transportation Research Part E*, 48(3):616–636, 2012.

[77] M.L. Fisher. Optimal solution for vehicle routing problem using minimum k-trees. *Operations Research*, 42(4):626–642, 1994.

[78] B. Fleischmann. The vehicle routing problem with multiple use of vehicles. Technical report, Fachbereich Wirtschaftswissenschaften, Universität Hamburg, 1990.

[79] B. Fleischmann, M. Gietz, and S. Gnutzmann. Time-varying travel times in vehicle routing. *Transportation Science*, 38(2):160–173, 2004.

[80] B. Fleischmann, S. Gnutzmann, and E. Sandvoß. Dynamic vehicle routing based on online traffic information. *Transportation Science*, 38(4):420–433, 2004.

[81] Organization for Economic Co-operation and Development. Delivering the goods: 21st century challenges to urban goods transport. Technical report, OECD Publishing, `http://www.oecdbookshop.org/`, 2003.

[82] H. Gehring and J. Homberger. A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows. In *EUROGEN99*. 1999.

[83] H.N. Geismar, G. Laporte, L. Lei, and C. Sriskandarajah. The integrated production and transportation scheduling problem for a product with a short lifespan. *INFORMS Journal of Computing*, 20(1):21–33, 2008.

[84] M. Gendreau, F. Guertin, J.-Y. Potvin, and É. Taillard. Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Science*, 33(4):381–390, 1999.

[85] M. Gendreau, A. Hertz, and G. Laporte. New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, 40(6):1086–1094, 1992.

[86] B. Gendron and F. Semet. Formulations and relaxations for a multi-echelon capacitated location-distribution problem. *Computers & Operations Research*, 36(5):1335–1355, 2009.

[87] M. Gentili and P.B. Mirchandani. Locating sensors on traffic networks: Models, challenges and research opportunities. *Transportation Research Part C*, 24:227–255, 2012.

[88] G. Ghiani, F. Guerriero, G. Laporte, and R. Musmanno. Real-time vehicle routing: Solutions concepts, algorithms and parallel computing strategies. *European Journal of Operational Research*, 151(1):1–11, 2003.

[89] B. Gillet and L. Miller. A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2):340–349, 1974.

[90] B. Golden, S. Raghavan, and E. Wasil. *The Vehicle Routing Problem - Last Advances and New Challenges*. Operations Research Computer Science Interfaces. Springer, 2008.

[91] J. Gonzalez-Feliu. *Models and methods for the City Logistic. The Two-Echelon Capacitated Vehicle Routing Problem*. PhD thesis, Politecnico di Torino, 2008.

[92] J. Gonzalez-Feliu. Two-echelon freight transport optimisation: Unifying concepts via a systematic review. *Working Papers on Operations Management*, 2(1):18–30, 2011.

[93] J. Gonzalez-Feliu, F. Toilier, and J.L. Routhier. End consumer movement generation in french medium urban areas. *Procedia - Social and Behavioural Sciences*, 2(3):6189–6204, 2010.

[94] P. Grangier, M. Gendreau, F. Lehuédé, and L.-M. Rousseau. Synchronisation entre tournées dans un problème de tournées de véhicules multi-échelons. In *14^{e} conférence de la Société Française de Recherche Opérationelle et Aide à la Décision*, Troyes, France, March 2013.

[95] I. Gribkovskaia, B.O. Gullberg, K.J. Hovden, and S.W. Wallace. Optimization model for a livestock collection problem. *International Journal of Physical Distribution & Logistics Management*, 36(2):136–152, 2006.

[96] T. Grünert. GTS Systems & Consulting GmbH, 2013. Private communication.

[97] H. Grzybowska and J Barceló. Decision support system for real-time urban freight management. *Procedia - Social and Behavioral Sciences*, 39:712–725, 2012.

[98] A.R. Güner, A. Murat, and R.B. Chinnam. Dynamic routing under recurrent and non-recurrent congestion using real-time its information. *Computers & Operations Research*, 39(2):358–373, 2012.

[99] G. Gutin and A.P. Punnen. *The Traveling Salesman Problem and Its Variations*, volume 12 of *Combinatorial Optimization*. Kluwer Academin Publisher, 2002.

[100] E.E. Halvorsen-Weare, K. Fagerholt, L.M. Nonås, and B.E. Asbjørnslett. Optimal fleet composition and periodic routing of offshore supply vessels. *European Journal of Operational Research*, 223(2):508–517, 2012.

[101] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[102] H. Hashimoto, M. Yagiura, and T. Ibaraki. An iterated local search algorithm for time-dependent vehicle routing problem with time windows. *Discrete Optimization*, 5(2):434–456, 2008.

[103] G. Hasle and O. Kloster. *Geometric Modelling, Numerical Simulation, and Optimization*, pages 397–436. Operations Research Computer Science Interfaces. Springer, 2007.

[104] V.C. Hemmelmayr, J.F. Cordeau, and T.G. Crainic. An adaptive large neighbourhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & Operations Research*, 39(12):3215–3228, 2012.

[105] F. Hernandez, D. Feillet, R. Giroudeau, and O. Naud. An exact algorithm to solve the multi-trip vehicle routing problem with time windows. Technical report, 2013.

[106] F. Hernandez, D. Feillet, R. Giroudeau, and O. Naud. A new exact algorithm to solve the multi-trip vehicle routing problem with time windows and limited duration. *4OR*, doi:10.1007/s10288-013-0238-z, 2013.

[107] A.V. Hill and W.C. Benton. Modelling intra-city time-dependent travel speeds for vehicle scheduling problems. *Journal of Operational Research Society*, 43(4):343–351, 1992.

[108] F.L. Hitchcock. The distribution of a product from several sources to numerous facilities. *Journal of Mathematical Physics*, 20(2):224–230, 1941.

[109] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[110] M.E. Horn. Efficient modeling of travel in networks with time-varying link speeds. *Networks*, 36(2):80–90, 2000.

[111] S. Ichoua, M. Gendreau, and J.-Y. Potvin. Diversion issues in real-time vehicle dispatching. *Transportation Science*, 34(4):426–438, 2000.

[112] S. Ichoua, M. Gendreau, and J.-Y. Potvin. Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research*, 144(2):379–396, 2003.

[113] S. Ichoua, M. Gendreau, and J.-Y. Potvin. Exploiting knowledge about future demands for real-time vehicle dispatching. *Transportation Science*, 40(2):211–225, 2006.

[114] S. Ichoua, M. Gendreau, and J.-Y. Potvin. *Planned Route Optimization for Real-Time Vehicle Routing*, pages 1–18. Springer, 2007.

[115] S. Irnich. A unified modeling and solution framework for vehicle routing and local search-based metaheuristics. *INFORMS Journal on Computing*, 20(2):270–287, 2008.

[116] S.K. Jacobsen and O.B.G. Madsen. A comparative study of heuristic for a two-level routing-location problem. *European Journal of Operational Research*, 5(6):378–387, 1980.

[117] M. Jepsen, S. Spoorendonk, and S. Ropke. A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem. *Transportation Science*, 4(1):23–37, 2013.

[118] D.E. Kaufman and R.L Smith. Fatest paths in time-dependent networks for intelligent vehicle-highway systems application. *IVHS Journal*, 1(1):1–11, 1993.

[119] S. Kim, M. Lewis, and C. White. Optimal vehicle routing with real-time traffic information. *Intelligent Transportation Systems, IEEE Transactions on*, 6(2):178–188, 2005.

[120] A. Kleff. PTV Group, Karlsruhe, Germany, 2013. Private communication.

[121] C. Koc and I. Karaoglan. A branch and cut algorithm for the vehicle routing problem with multiple use of vehicles. Proceedings of the 41st International Conference on Computers & Industrial Engineering. Los Angeles, USA, 2011.

[122] U. Kohler. How to change urban traffic with city logistics. volume 1 of *Control in transportation systems 2000 : a proceedings volume from the 9th IFAC Symposium*. Braunschweig, Germany, 2001.

[123] A.L. Kok, E.W Hans, and J.M.J. Shutten. Vehicle routing under time-dependent travel times: the impact of congestion avoidance. Technical Report 267, Beta Research School for Operations Management and Logistics, University of Twente, 2009.

[124] A.L. Kok, E.W. Hans, J.M.J. Shutten, and W.H.M Zijm. A dynamic programming heuristic for a vehicle routing with time-dependent travel times and required breaks. *Flexible Services and Manufacturing Journal*, 22(1–2):83–108, 2010.

[125] S. Kritzinger, K.F. Doerner, R.F. Hartl, G. Kiechle, H. Stadler, and S.S. Manohar. Using traffic information for time-dependent vehicle routing. *Procedia - Social and Behavioral Sciences*, 39, 2011.

[126] Y. Kuo. Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. *Computers & Industrial Engineering*, 59(1):157–165, 2010.

[127] Y. Kuo, C.C. Wang, and P.Y. Chuang. Optimizing goods assignment and the vehicle routing problem with time-dependent travel speeds. *Computers & Industrial Engineering*, 57(4):1385–1392, 2009.

[128] A. Larsen. *The Dynamic Vehicle Routing Problem*. PhD thesis, Technical University of Denmark (DTU), 2001.

[129] A. Larsen, O.B.G. Madsen, and M.M Solomon. In *The Vehicle Routing Problem - Latest Advances and New Challenges*, number 43 in Operations Research Computer Science Interfaces, chapter Recent Developments in Dynamic Vehicle Routing Systems, pages 199–218. Springer, 2008.

[130] Y. Lee, J. Jung, and K. Lee. Vehicle routing scheduling for cross-docking in the supply chain. *Computers & Industrial Engineering*, 51(2):247–256, 2006.

[131] L. Lei, S. Liu, A. Ruszczynski, and S. Park. On the integrated production, inventory, and distribution routing problem. *IIIE Transactions*, 38(11):955–970, 2006.

[132] J.K. Lenstra and H.G. Rinnooy Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227, 1981.

[133] C.K.Y. Lin and R.C.W. Kwok. Multi-objective metaheuristic for a location routing problem with multiple use of vehicles on real data and simulated data. *European Journal of Operational Research*, 175(3):1833–1849, 2006.

[134] A. Lodi, S. Martello, and D. Vigo. Tspack: A unified tabu search code for multi-dimensional bin packing problems. *Annals of Operations Research*, 131(1–4):203–213, 2004.

[135] H.R. Lourenço, O.C. Martin, and T. Stüzle. Iterated local search: Framework and applications. In *Handbook of Metaheuristics - Second Edition*, International series in operations research and management science, chapter 12, pages 363–397. Springer, 2010.

[136] R. Macedo, C. Alves, J.M. Valério deCarvalho, F. Clautiaux, and S. Hanafi. Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model. *European Journal of Operational Research*, 214(3):536–545, 2011.

[137] R. Macedo, S. Hanafi, F. Clautiaux, C. Alves, and J.M. Valério de Carvalho. Generalized disaggregation algorithm for the vehicle routing problem with time windows and multiple routes. 1st International Conference on Operations Research and Enterprise Systems, pages 305–312. 2012.

[138] W. Maden, R. Eglese, and D. Black. Vehicle routing and scheduling with time-varying data: A case study. *Journal of Operational Research Society*, 61:515–522, 2010.

[139] C. Malandraki and M.S Daskin. Time dependent vehicle routing problems: formulations, properties and heuristic algorithms. *Transportation Science*, 26(3):185–200, 1992.

[140] A. Mingozzi, R. Roberti, and P. Toth. An exact algorithm for the multi-trip vehicle routing problem. *INFORMS Journal on Computing*, 25(2):193–207, 2013.

[141] S. Mitrović-Minić and G. Laporte. Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B*, 38(7):635–655, 2004.

[142] P. Moscato and C. Cotta. A modern introduction to memetic algorithms. In *Handbook of Metaheuristics - Second Edition*, International series in operations research and management science, chapter 6, pages 141–183. Springer, 2010.

[143] J. Muñuzuri, P. Cortés, J. Guadix, and L. Onieva. City logistics in spain: Why it might never work. *Cities*, 29(2):133–141, 2012.

[144] J. Muñuzuri, R. Grosso, P. Cortés, and J. Guadix. Estimating the extra costs imposed on delivery vehicles using access time windows in a city. *Computers, Environment and Urban System*, 41:262–275, 2013.

[145] Y. Nagata, O. Bräysy, and W. Dullaert. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, 37(4):724–737, 2010.

[146] G. Nagy and S. Salhi. Location-routing: Issues, models and methods. *European Journal of Operational Research*, 177(2):649–672, 2007.

[147] F. Neri and C. Cotta. Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation*, 2:1–14, 2012.

[148] P.K. Nguyen, T.G. Crainic, and M. Toulouse. A tabu search for time-dependent multi-zone multi-trip vehicle routing problem with time windows. *European Journal of Operational Research*, 231(1):43–56, 2013.

[149] V.-P. Nguyen, C. Prins, and C. Prodhon. Solving the two-echelon location routing problem by a grasp reinforced by a learning process and path relinking. *European Journal of Operational Research*, 216(1):113–126, 2012.

[150] A.G.N. Noveas, E.M. Frazzon, and P.J. Burin. Dynamic vehicle routing in over congested areas. In *Dynamics in Logistics, Second International Conference, LDIC 2009*, pages 49–58, 2011.

[151] Commission of the European Communities. Action plan on urban mobility. Technical Report COM(2009) 490, `http://ec.europa.eu`, 2009.

[152] Commission of the European Communities. Full evaluation report - satellite based traffic management for smes. Technical report, `www.civitas.eu/index.php?id=79\&sel\_menu=23\&measure\_id=250`, 2009.

[153] Commission of the European Communities. A sustainable future for transport: Towards an integrated, technology-led and user friendly system. Technical Report COM(2009) 279, `http://ec.europa.eu`, 2009.

[154] Commission of the European Communities. Roadmap to a single european transport area - towards a competitive and resource efficient transport system. Technical Report COM(2011) 144, `http://ec.europa.eu`, 2011.

[155] K.W. Ogden. *Urban Goods Transportation: A Guide to Policy and Planning*. Ashgate, Hants, 1992.

[156] A. Olivera and O. Viera. Adaptive memory programming for the vehicle routing problem with multiple trips. *Computers & Operations Research*, 34(1):28–47, 2007.

[157] J.D. Ortúzar and L.G. Willumsen. *Modelling Transport*. Wiley, 4th edition, 2011.

[158] D. Patier, B. David, V. Deslandres, and R. Chalon. A new concept for urban logistics: Delivery area booking. In *The Eight International Conference on City Logistics*. Bali, Indonesia, 2013.

[159] D. Patier and J.L. Routhier. *Transport Survey Methods. Keeping up with a changing world*, chapter How to Improve the Capture of Urban Goods Movement Data, pages 251–287. Emerald, Bingley, UK, 2009.

[160] G. Perboli, R. Tadei, and D. Vigo. The two-echelon capacitated vehicle routing problem: Models and math-based heuristic. *Transportation Science*, 45(3):364–380, 2011.

[161] R.J. Petch and S. Salhi. A multi-phase constructive heuristic for the vehicle routing problem with multi trips. *Discrete Applied Mathematics*, 133(1–3):69–92, 2004.

[162] V. Pillac, M. Gendreau, C. Guéret, and A.L. Medaglia. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11, 2013.

[163] M.L. Pinedo. *Planning and Scheduling in Manufacturing and Services*, chapter Manufacturing Models, pages 19–36. Springer Series in Operations Research. Springer, 2001.

[164] J.-Y. Potvin, Y. Xu, and I. Benyahia. Vehicle routing and scheduling with dynamic travel times. *Computation & Operations Research*, 33(4):1129–1137, 2006.

[165] C. Prins. Efficient heuristic for the heterogeneous fleet multitrip vrp with application to a large-scale real case. *Journal of Mathematical Modelling and Algorithms*, 1:135–150, 2002.

[166] C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12):1985–2002, 2004.

[167] C Prins. Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence*, 22(6):916–928, 2009.

[168] C. Prins, N. Labadi, and M. Reghioui. Tour splitting algorithms for vehicle routing problems. *International Journal of Production Research*, 47(2):507–535, 2009.

[169] H.N. Psaraftis. Dynamic vehicle routing: Status and prospects. *Annals of Operations Research*, 61(1):143–164, 1995.

[170] V. Pureza and G. Laporte. *Waiting and Buffering Strategies for the Dynamic Pickup and Delivery Problem with Time Windows*, pages 199–218. Number 43 in Operations Research Computer Science Interfaces. Springer, 2008.

[171] H.J. Quak. Improving urban freight transport sustainability by carriers - best practices from the netherlands and the eu project citylog. *Procedia - Social and Behavioral Sciences*, 39:158–171, 2012.

[172] H.J. Quak and M.B.M de Koster. Delivering goods in urban areas: How to deal with urban policy restrictions and the environment. *Transportation Science*, 43(2):211–227, 2009.

[173] A.G. Qureshi and S. Hanaoka. Analysis of the effects of a cooperative delivery system in bangkok. In *Recent Adavances in City Logistics*. Langkawi, Malaysia, July 2005.

[174] A.G. Qureshi, E. Taniguchi, and T. Yamada. A microsimulation based analysis of exact solution of dynamic vehicle routing with soft time windows. *Procedia - Social and Behavioral Sciences*, 39, 2011.

[175] B. Raa and E.-H. Aghezzaf. Designing distribution patterns for long-term inventory routing with constant demand rates. *International Journal of Production Economics*, 112(1):255–263, 2008.

[176] B. Raa and E.-H. Aghezzaf. A practical solution approach for the cyclic inventory routing problem. *European Journal of Operational Research*, 192(2):429–441, 2009.

[177] C.R. Reeves. Genetic algorithms. In *Handbook of Metaheuristics - Second Edition*, International series in operations research and management science, chapter 5, pages 109–140. Springer, 2010.

[178] Y. Rochat and É. Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1(1):147–167, 1995.

[179] J.P. Rodrigue. *The Geography of Transportation Systems*. Routledge - Taylor & Francis Group, 2013. http://people.hofstra.edu/geotrans/index.html.

[180] D. Ronen. Allocation of trips to trucks operating from a single terminal. *Computers & Operations Research*, 19(5):445–451, 1992.

[181] J.L. Routhier. Urban goods transport data collection: Indicators and efficiency? a french experience. In *Städtischer Wirtschaftsverkehr - Commercial/Goods Transportation in Urban Areas - Transports Commerciaux/Marchandises en Ville*, pages 75–90. Berlin, 2012.

[182] J.L. Routhier and F. Toilier. Freturb v3, a policy oriented software of modeling urban goods movement. page 35p. Berkeley, USA, June 2007.

[183] A. Rushton, P. Croucher, and P. Baker. *The handbook of Logistic and Distribution Management – 3rd edition*. Kogan Page, 2006.

[184] W. Ruske. City logistics - solutions for urban commercial transport by cooperative operations management. pages 195–228. Omiya, Japan, 1994.

[185] S. Salhi. *The integration of routing into location-allocation and vehicle fleet composition problems*. PhD thesis, university of Lancaster, 1987.

[186] S. Salhi and R.J. Petch. A GA based heuristic for the vehicle routing problem with multiple trips. *Journal of Mathematical Modeling and Algorithms*, 6(4):591–613, 2007.

[187] M.W.P. Savelsbergh. Local search in routing problems with time windows. *Annals of Operations Research*, 4, 1985.

[188] M. Schneider, B. Sand, and A. Stenger. A note on the time travel approach for handling time windows in vehicle routing problems. *Computers & Operations Research*, 2013.

[189] E. Segalou, C. Ambrosini, and J.L. Routhier. The environmental assessment of urban goods movement. In *Logistics systems for sustainable cities - Proceedings of the 3rd International Conference on City Logistics*, pages 207–220. Elsevier, 2004.

[190] A. Shyshou, K. Fagerholt, I. Gribkovskaia, and G. Laporte. A large neighbourhood search heuristic for a periodic supply vessel planning problem arising on offshore oil and gas operations. Technical Report SINTEF S18202, 2010.

[191] A.M. Silas, J. Holguín-Veras, and S. Jara-Díaz. Optimal distribution of financial incentives to foster off-hour deliveries in urban areas. *Transportation Research Part A*, 46(8):1205–1215, 2012.

[192] Transportation Optimization Portal Sintef. `http://www.sintef.no/Projectweb/TOP/VRPTW/Solomon-benchmark/100-customers/`, november 2013.

[193] Transportation Optimization Portal Sintef. `http://www.sintef.no/Projectweb/TOP/VRPTW/Homberger-benchmark/`, november 2013.

[194] S.K. Smit and A.E. Eiben. Comparing parameter tuning methods for evolutionary algorithms. In *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, pages 399–406. 2009.

[195] D. Soler, J. Albiach, and E. Martínez. A way to optimally solve a time-dependent vehicle routing problem with time windows. *Operations Research Letters*, 37(1):37–42, 2009.

[196] M.M. Solomon. Algorithms for the vehicle routing and scheduling problem with time windows constraints. *Operations Research*, 35:254–265, 1987.

[197] É.D. Taillard. Parallel iterative search methods for vehicle routing problems. *Networks*, 23(8):661–673, 1993.

[198] É.D. Taillard, G. Laporte, and M. Gendreau. Vehicle routeing with multiple use of vehicles. *Journal of Operational Research Society*, 47(8):1065–1070, 1996.

[199] E. Taniguchi and H. Shimamoto. Intelligent transportation sustem based dynamic vehicle routing and scheduling with variable travel times. *Transportation Research Part C*, 12(3), 2004.

[200] E. Taniguchi, R.G. Thompson, T. Yamada, and R. van Duin. *City Logistic – network modelling and intelligent transport systems*. Pergamon, Amsterdam, 2001.

[201] E. Taniguchi and R.E.C.M Van Der Heijden. An evaluation methodology for the city logistics. *Transport Reviews*, 20(1):65–90, 2000.

[202] C.D. Tarantilis and C.T. Kiranoudis. Using the vehicle routing problem for the transportation of hazardous materials. *Operational Research*, 1(1):67–78, 2001.

[203] R.G. Thompson and K.P. Hassall. A collaborative urban distribution network. *Procedia - Social and Behavioral Sciences*, 39:230–240, 2012.

[204] P. Toth and D. Vigo. *The Vehicle Routing Problem*. Monographs on Discrete Mathematics and Applications. Siam, 2002.

[205] P. Toth and D. Vigo. The granular tabu search and its application to the vehicle routing problem. *INFORMS Journal of Computing*, 15(4):333–346, 2003.

[206] P. Tsirimpas, A. Tatarakis, I. Minis, and E.G. Kyriakidis. Single vehicle routing with a predefined customer sequence and multiple depot returns. *European Journal of Operational Research*, 187(2):483–495, 2008.

[207] D.V. Tung and A. Pinnoi. Vehicle routing-scheduling for waste collection in hanoi. *European Journal of Operational Research*, 125(3):449–468, 2000.

[208] A.C. Ullrich. Integrated machine scheduling and vehicle routing with time windows. *European Journal of Operational Research*, 227(1):152–165, 2013.

[209] J. Van Bell, P. Valckenaers, and D. Cattrysse. Cross-docking: State of the art. *Omega*, 40(6):827–846, 2012.

[210] M.G. Van Buer, D.L. Woodruff, and R.T. Olson. Solving the medium newspaper production/distribution problem. *European Journal of Operational Research*, 115(2):237–253, 1999.

[211] T. van Rooijen and H. Quak. Local impacts of a new urban consolidation centre – the case of binnenstadservice.nl. *Procedia - Social and Behavioral Sciences*, 2(3):5967–5979, 2010.

[212] T. Van Woensel, L. Kerbache, H. Peremans, and N. Vandaele. Vehicle routing with dynamic travel times: a queueing approach. *European Journal of Operational Research*, 186(3):990–1007, 2008.

[213] T. Vidal, T.G. Crainic, M. Gendreau, N. Lahrichi, and W. Rei. A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, 60(3):611–624, 2012.

[214] T. Vidal, T.G. Crainic, M. Gendreau, and C. Prins. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time windows. *Computers & Operations Research*, 40(1):475–489, 2013.

[215] Z. Wang, W. Liang, and X. Hu. A metaheuristic based on a pool of routes for the vehicle routing problem with multiple trips and time windows. *Journal of the Operational Research Society*, 2013.

[216] M. Wen, J. Larsen, J. Clausen, J.-F. Cordeau, and G. Laporte. Vehicle routing with cross-docking. *Journal of the Operational Research Society*, 60:1708–1718, 2009.

[217] S. Wøhlk. In *The Vehicle Routing Problem - Last Advances and New Challenges*, Operations Research Computer Science Interfaces, chapter A Decade of Capacitated Arc Routing, pages 29–48. Springer, 2008.

[218] J. Yang, P. Jaillet, and H. Mahmassani. Real-time multivehicle truckload pickup and delivery problems. *Transportation Science*, 38(2):135–148, 2004.

[219] P.C. Yellow. A computational modification to the savings method of vehicle scheduling. *Operational Research Quarterly*, 21(2):281–283, 1970.

[220] V. Zeimpekis and G.M. Giaglis. A dynamic real-time management system for incident handling in city logistics. In *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, volume 5, pages 2900–2904, 2005.

École Nationale Supérieure des Mines

de Saint-Étienne

NNT : *2014 EMSE 0737*

Diego CATTARUZZA

VEHICLE ROUTING FOR CITY LOGISTICS

Speciality: Industrial Engineering

Key words: Vehicle routing, multi-trip, city logistics, heuristic, genetic algorithm, memetic algorithm

Abstract:

Transportation of merchandise in urban areas has become an important nowadays topic. In fact, transportation is a vital activity for each city, but entail pollution, congestion, accidents. City logistics aims at optimizing the whole urban logistics and transportation process, taking into account environmental and social aspects. This thesis, that is part of the MODUM project, finds its location in this area of research. In particular, MODUM aims at studying a delivery system based on City Distribution Centers.

We first present a classification and an analysis of urban good movements and routing problems peculiar to metropolitan areas. A second survey proposes a complete collection of articles that has been done on the Multi Trip Vehicle Routing Problem (MTVRP). The MTVRP is an extension of the Vehicle Routing Problem (VRP) where vehicles are allowed to perform several trips.

We propose an efficient heuristic for the MTVRP that is, in a subsequent step, adapted to a new routing problem, the MTVRP with Time Windows and Release Dates (MTVRPTWR). It is a variant of the MTVRP where each customer is associated with a time window and each merchandise is associated with a release date that represents the instant it becomes available at the depot.

We, then, study a variant of the MTVRP where goods belong to different commodities that cannot be transported at the same time by the same vehicle. An analysis is conducted on the benefits of the multi-trip aspect in fleet dimensioning problems.

Finally we describe the complex routing problem that arises in MODUM and the simulator that is developed to evaluate the performances of the system.

École Nationale Supérieure des Mines

de Saint-Étienne

NNT : *2014 EMSE 0737*

Diego CATTARUZZA

OPTIMISATION DE TOURNÉES DE VÉHICULES POUR LA LOGISTIQUE URBAINE

Spécialité: Génie Industriel

Mots clefs: Tournées de véhicules, multi-trip, logistique urbaine, algorithme génétique, algorithme memetique

Résumé:

Le transport de marchandises dans les zones urbaines est un sujet important de nos jours. Le transport est une activité vitale pour les villes, mais implique pollution, congestion, accidents. La logistique urbaine vise à optimiser les processus logistiques et de transports urbains en tenant compte des aspects environnementaux et sociaux. Cette thèse traite de cette thématique et fait partie du projet MODUM.

MODUM vise à étudier un système de livraison basé sur des centres de distribution urbains. Nous présentons une classification et une analyse des mouvements de marchandises et des problèmes de tournées de véhicules (VRP) associés.

La deuxième partie propose une revue complète des travaux de recherche traitant des problème VRP avec routes multiples (MTVRP). Le MTVRP est une extension du VRP où les véhicules sont autorisés à effectuer plusieurs tournées. Nous proposons une heuristique pour le MTVRP qui est par la suite adaptée pour un problème plus riche, le MTVRP avec fenêtres de temps et dates de disponibilité. Il s'agit d'une variante du MTVRP où à chaque client est associée une fenêtre de temps et à chaque marchandise une date de disponibilité qui représente l'instant où elle devient disponible au dépôt.

Par la suite, nous étudions une variante du MTVRP où les marchandises sont classées par types de produits qui ne peuvent pas être transportés dans le même véhicule. Une analyse est effectuée pour montrer l'avantage des tournées multiples pour le problème de dimensionnement des flottes.

Enfin, nous décrivons le problème de tournées qui se pose dans MODUM et le simulateur qui est développé pour évaluation du système.