



HAL
open science

Optimizing the daily transport for people with disabilities

Oscar Tellez

► **To cite this version:**

Oscar Tellez. Optimizing the daily transport for people with disabilities. Other. Université de Lyon, 2019. English. NNT : 2019LYSEI059 . tel-02956227

HAL Id: tel-02956227

<https://theses.hal.science/tel-02956227>

Submitted on 2 Oct 2020

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

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



INSA

N° d'ordre

NNT : 2019LYSEI059

THESE de DOCTORAT DE L'UNIVERSITÉ DE LYON
opérée au sein de
INSA Lyon

École Doctorale N° ED512
Informatique et Mathématiques

Spécialité/ discipline de doctorat :
Génie Industriel

Soutenue publiquement le 04/09/2019, par:
Oscar Augusto TELLEZ SANCHEZ

Optimisation du transport quotidien des personnes en situation de handicap

Devant le jury composé de :

NORRE, Sylvie	Professeur Université Clermont-Auvergne	Rapporteur
JUSSIEN GUERET, Christelle	Professeur Université d'Angers	Rapporteur
DI MASCOLO, Maria (Présidente)	Directrice de recherche Lab. G-SCOP	Examinatrice
GARAIX, Thierry	Maître Assistant École des Mines St. Étienne	Examinateur

MONTEIRO, Thibaud	Professeur INSA Lyon	Directeur
VERCRAENE, Samuel	Maître de conférences INSA Lyon	Co-directeur
LEHUÉDÉ, Fabien	Professeur IMT Atlantique	Co-directeur
PÉTON, Olivier	Professeur IMT Atlantique	Co-directeur

Optimizing the daily transport for people with disabilities

By

OSCAR AUGUSTO TELLEZ SANCHEZ

DISP Laboratory, Université de Lyon, Insa Lyon

A dissertation submitted to the Université de Lyon in accordance with the requirements of the degree of PHD in Industrial Engineering from the Doctoral School Informatiques et Mathématiques ED512.

JUNE, 2019

Département FEDORA – INSA Lyon - Ecoles Doctorales – Quinquennal 2016-2020

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	<p><u>CHIMIE DE LYON</u> http://www.edchimie-lyon.fr Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage secretariat@edchimie-lyon.fr INSA : R. GOURDON</p>	<p>M. Stéphane DANIELE Institut de recherches sur la catalyse et l'environnement de Lyon IRCELYON-UMR 5256 Équipe CDFA 2 Avenue Albert EINSTEIN 69 626 Villeurbanne CEDEX directeur@edchimie-lyon.fr</p>
E.E.A.	<p><u>ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE</u> http://edeea.ec-lyon.fr Sec. : M.C. HAVGOUDOUKIAN ecole-doctorale.eea@ec-lyon.fr</p>	<p>M. Gérard SCORLETTI École Centrale de Lyon 36 Avenue Guy DE COLLONGUE 69 134 Écully Tél : 04.72.18.60.97 Fax 04.78.43.37.17 gerard.scorletti@ec-lyon.fr</p>
E2M2	<p><u>ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION</u> http://e2m2.universite-lyon.fr Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 INSA : H. CHARLES secretariat.e2m2@univ-lyon1.fr</p>	<p>M. Philippe NORMAND UMR 5557 Lab. d'Ecologie Microbienne Université Claude Bernard Lyon 1 Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69 622 Villeurbanne CEDEX philippe.normand@univ-lyon1.fr</p>
EDISS	<p><u>INTERDISCIPLINAIRE SCIENCES-SANTÉ</u> http://www.ediss-lyon.fr Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 INSA : M. LAGARDE secretariat.ediss@univ-lyon1.fr</p>	<p>Mme Emmanuelle CANET-SOULAS INSERM U1060, CarMeN lab, Univ. Lyon 1 Bâtiment IMBL 11 Avenue Jean CAPELLE INSA de Lyon 69 621 Villeurbanne Tél : 04.72.68.49.09 Fax : 04.72.68.49.16 emmanuelle.canet@univ-lyon1.fr</p>
INFOMATHS	<p><u>INFORMATIQUE ET MATHÉMATIQUES</u> http://edinfomaths.universite-lyon.fr Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 infomaths@univ-lyon1.fr</p>	<p>M. Luca ZAMBONI Bât. Braconnier 43 Boulevard du 11 novembre 1918 69 622 Villeurbanne CEDEX Tél : 04.26.23.45.52 zamboni@maths.univ-lyon1.fr</p>
Matériaux	<p><u>MATÉRIAUX DE LYON</u> http://ed34.universite-lyon.fr Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bât. Direction ed.materiaux@insa-lyon.fr</p>	<p>M. Jean-Yves BUFFIÈRE INSA de Lyon MATEIS - Bât. Saint-Exupéry 7 Avenue Jean CAPELLE 69 621 Villeurbanne CEDEX Tél : 04.72.43.71.70 Fax : 04.72.43.85.28 jean-yves.buffiere@insa-lyon.fr</p>
MEGA	<p><u>MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE</u> http://edmega.universite-lyon.fr Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bât. Direction mega@insa-lyon.fr</p>	<p>M. Jocelyn BONJOUR INSA de Lyon Laboratoire CETHIL Bâtiment Sadi-Carnot 9, rue de la Physique 69 621 Villeurbanne CEDEX jocelyn.bonjour@insa-lyon.fr</p>
ScSo	<p><u>ScSo*</u> http://ed483.univ-lyon2.fr Sec. : Véronique GUICHARD INSA : J.Y. TOUSSAINT Tél : 04.78.69.72.76 veronique.cervantes@univ-lyon2.fr</p>	<p>M. Christian MONTES Université Lyon 2 86 Rue Pasteur 69 365 Lyon CEDEX 07 christian.montes@univ-lyon2.fr</p>

Abstract

IN most developed countries, the transport for people with disabilities is subsidized with public funds. In France, the number of transported people increased by 18% from 2006 to 2008 (Laurent et al., 2012). In 2017, the cost of this transport was around 500 M€ representing 10% of the total medical transportation cost (Gonzalez et al., 2018). Between 2006 and 2014, this cost has increased by 40% (Wahl et al., 2016).

For Medico-Social Institutions (MSI), transport costs often represent the second-largest expense after that of the staff. Despite this, few MSIs have the means to effectively manage the transport to provide high-quality service and keep the costs down. With little expertise, MSIs are often led to adopt sub-optimal practices that impact costs, quality of service or the environment.

In the medico-social sector, transport is no longer considered merely a painful or non-value-added activity. It is generally integrated into the life project of People With Disabilities (PWD). Transportation allows the inclusion of the PWD to school or work life. It can be a lever for the autonomy of PWD. Transport demands in the medico-social sector are mostly regular and very often known in advance. This anticipated knowledge of needs can be used to jointly improve: the quality of service for users, and the efficiency of the resources used.

This thesis is part of the project “Numérique et Optimisation pour une Mobilité Adaptée” (NOMAd). This project aims to improve the daily transport service for people with disabilities between their homes and the MSIs. To this end, we suggest the transport pooling between several MSIs on one side and a global optimization strategy for building the annual transport planning on the other side. The construction of the annual transport planning is a complex problem that has not been completely solved in the scientific literature. Following the current practice, we can assume that the annual transport plan is composed of a pattern of identical weekly schedules.

To solve the weekly problem, we first work on the planning of the half-day transport problem. This problem is modeled as a mixed-integer program called the fleet size and mix dial-a-ride problem with multiple passenger types and with a heterogeneous fleet of reconfigurable vehicles. In this new variant of the dial-a-ride problem, en-route modifications of the vehicle’s inner configuration are allowed. The main consequence is that the vehicle capacity is defined by a set of configurations and the choice of vehicle configuration is associated with binary decision variables. Vehicle reconfiguration is a lever to efficiently reduce transport costs, but the number

of passengers and vehicle fleet setting make this problem intractable for exact solution methods. A large neighborhood search matheuristic combined with a set covering component and a reactive mechanism to automatically adjust its parameters is therefore proposed.

Once we solve the half-day transport problem, we conduct a weekly integration of solutions. However, the simple juxtaposition of half-day transport schedules give cost-efficient transport planning but a very inconsistent service time for each passenger. This situation is uncomfortable for PWD, or even unacceptable for some people with mental disabilities. These service time requirements together with traditional route planning define a new variant of the multi-period dial-a-ride problem that we denote the time consistent DARP. This problem is modeled as a route based mathematical modeling considering two objectives: the transport cost and the service time consistency.

A perfectly consistent schedule, defined for each passenger a single service time during the planning horizon. However, this transport solution is very expensive for MSIs. With a lexicographic definition of the solution time-consistency, we are able to produce a large variety of less consistent but affordable solutions that fits better to the needs of MSIs. The problem is solved with a matheuristic framework based on a master set partitioning problem and routes generated from a large neighborhood search procedure.

Currently, the pooling of transport between MSIs is not a common practice. Instead, each MSI manages its transport independently. This way of working has technical and administrative advantages in terms of optimization, contracting and negotiation. On the contrary, dealing with multiple MSIs at a time supposes a simultaneous negotiation process, and requires a specialized mechanism for pricing users individually. Despite the advantages of the current policy, there is a major economic drawback that must be weighed against its benefits. An assessment against two other policies is presented in the context of both economic and service aspects. Alternative policies consider transport pooling between several MSIs. The service quality indicator for this study is the user ride time.

The transport pooling between MSIs is both a research and managerial challenge. Currently, transport management is mainly carried out independently in each MSI. The pooling of transport increases the size of the problem and therefore its complexity. It also implies having, on the one hand, economic interest and, on the other hand, a common mechanism of effective management and communication between the two parties. The objective of this study is to evaluate the impact of transport pooling on the costs and the ride time of the users. From an optimization point of view, this solving approach is known as “clustering first route second” for large scale routing problems. A comparison is made between a scenario without transport pooling and other scenarios allowing transport pooling shows that up to 30% of transport costs can be saved.

Résumé

Dans la plupart des pays développés, le transport des personnes en situation de handicap est subventionné avec des fonds publics. En 2017, le coût de ce transport en France était d'environ 500 M€ soit 10% du coût total du transport médical (Gonzalez et al., 2018). Entre 2006 et 2014, ce coût a augmenté de 40% (Wahl et al., 2016) et le nombre de personnes transportées de 18% de 2006 à 2008 (Laurent et al., 2012). Pour les Établissements Sociaux et Médico-Sociaux (ESMS), les coûts de transport représentent souvent la deuxième dépense en importance après celle du personnel. Malgré cela, peu d'ESMS ont les moyens de gérer efficacement le transport afin de fournir un service de haute qualité et de réduire les coûts. Avec peu d'expertise, les ESMS sont souvent amenés à adopter des pratiques peu performants ayant un impact sur les coûts, la qualité de service ou l'environnement.

Dans le secteur médico-social, le transport n'est plus considéré comme une simple activité pénible ou sans valeur ajoutée. Il est généralement intégré au projet de vie des Personnes en Situation de Handicap (PSH). Le transport permet l'inclusion des PSH à l'école ou au travail, ce qui favorise leur autonomie. La demande de transport dans le secteur médico-social est généralement régulière et très souvent connue à l'avance. Cette connaissance anticipée des besoins peut être utilisée pour améliorer conjointement la qualité de service pour les utilisateurs et l'efficacité des ressources utilisées.

Dans ce contexte, cette thèse réalisée dans le cadre du projet «Numérique et Optimisation pour une Mobilité Adaptée» (NOMAd) vise à améliorer le service de transport quotidien des personnes en situation de handicap entre leur domicile et les ESMS. À cette fin, nous proposons la mutualisation du transport entre plusieurs ESMS et nous proposons une stratégie d'optimisation globale de la planification annuelle du transport.

La construction de la planification annuelle des transports est un problème complexe qui n'a pas été complètement résolu dans la littérature scientifique. Conformément à la pratique actuelle, nous pouvons supposer que le plan de transport annuel est composé d'un ensemble d'horaires hebdomadaires identiques.

Pour résoudre le problème hebdomadaire, nous travaillons d'abord à la planification d'un problème de transport d'une demi-journée. Ce problème est modélisé comme un programme à nombres entiers mixtes appelé "*the fleet size and mix dial-a-ride problem*", Dans ce problème, nous prenons en compte plusieurs types de passagers et un parc hétérogène de véhicules reconfigurables. Dans cette nouvelle variante du problème *dial-a-ride*, les modifications de la

configuration interne du véhicule sont durant le trajet sont autorisées. La conséquence principale est que la capacité du véhicule est définie par un ensemble de configurations dont le choix est associé à des variables de décision binaires. La reconfiguration des véhicules est un levier pour réduire efficacement les coûts de transport, mais le nombre de passagers et l'hétérogénéité du parc de véhicules rendent ce problème insoluble pour des méthodes de résolution exactes. Nous proposons donc une matheuristique de recherche de grand voisinage combinée à un *set partitioning problem* doté d'un mécanisme réactif pour ajuster automatiquement ses paramètres.

Une fois que nous avons résolu le problème de transport d'une demi-journée, nous procédons à une intégration hebdomadaire des solutions. Cependant, la simple juxtaposition d'horaires de transport d'une demi-journée peut permettre une planification très efficace du transport en terme de coûts, mais peut conduire à un service horaire très irrégulier pour chaque passager. Cette situation est inconfortable pour les PSH, voire inacceptable pour certaines personnes souffrant de certains handicaps mentaux. Ces exigences de régularité de service, combinées à la planification traditionnelle, définissent une nouvelle variante du problème *multi-period* DARP. Ce problème, nommé le *time-consistent* DARP prend en compte deux objectifs: le coût de transport et la régularité du temps de service.

Une planification parfaitement cohérente définit pour chaque passager une seule heure de service pendant l'horizon de planification. Cependant, cette solution de transport est très coûteuse pour les ESMS. Avec une définition lexicographique de la régularité des horaires, nous sommes en mesure de produire une grande variété de solutions moins régulières mais plus économiques et mieux adaptées aux besoins des ESMS. Le problème est résolu avec une matheuristique basée sur un *master set partitioning problem* et des routes générées à partir d'une procédure de recherche de grand voisinage.

La mutualisation du transport est à la fois un défi de recherche et un défi managérial. Actuellement, la gestion du transport est principalement réalisée intra-établissement c'est-à-dire sans mutualisation. La mutualisation du transport augmente la taille du problème et donc sa complexité. Elle suppose également d'avoir d'une part, un intérêt économique et d'autre part, un mécanisme commun de gestion et de communication efficace. L'objectif de cette étude est d'évaluer l'impact de la mutualisation sur les coûts de transport et le temps de trajet des usagers. En termes mathématiques, ce problème correspond à l'approche *clustering first route second* pour la résolution des problèmes de transport de grande taille. Sur un jeu de données issue du terrain, une comparaison est réalisée entre le transport sans mutualisation et deux scénarios avec différents niveaux de mutualisation.

Lo dedico a mi familia y mis seres queridos

*“et au loin, un écho,
Comme une braise sous la cendre
Un murmure à mis-mots,
Que mon coeur veut comprendre”
(extrait d’Anastasia)*

Acknowledgments

“malgret tout c’est un hasard avec un grand D”

Je voudrais remercier à toutes les personnes qui m’ont accompagné tout au long de cette aventure. Sans eux j’aurais certainement jamais réussi. Je pense tout d’abord à mes encadrants Samuel Vercaene, Fabien Léhuédé, Olivier Péton et mon Directeur Thibaud Monteiro qui m’ont toujours guidé et mon appris l’art de la recherche mais aussi des valeurs humaines. Je remercie également le soutien très important de ma famille en Colombie, de ma chérie Elena, de mes amis de Ste. Madelaine, mes collègues du laboratoire DISP, to my friends and English reviewers John Paul and Ka Yu (obviously any mistake is my fault!)

Table of Contents

	Page
List of Tables	xvii
List of Figures	xviii
1 Introduction	1
1.1 Context	2
1.2 The NOMAd project	3
1.3 Field study	3
1.3.1 Stakeholders	4
1.3.2 Survey: needs analysis	5
1.3.3 A multidimensional problem	8
1.4 Transportation management processes	9
1.5 Road map and contributions	12
1.5.1 Construction of the long-term transportation planning	12
1.5.2 Assessment of transport pooling policies	14
1.6 Conclusion	15
2 Literature review	17
2.1 Healthcare transport	17
2.1.1 Transport of staff, supplies, and patients & passengers	18
2.1.2 Emergency and non-emergency transport for patients & passengers	19
2.1.3 Occasional and regular transport for patients & passengers	21
2.2 Optimization problems in the non-emergency transport of patients & passengers with disabilities	22
2.2.1 Strategic problems	23
2.2.2 Tactical problems	23
2.2.3 Operational problems	24
2.2.4 Real-time problems	26
2.3 Conclusions	29

3	Half-day transport problem	31
3.1	Introduction	32
3.2	Literature review	34
3.2.1	The dial-a-ride problem	34
3.2.2	Routing a heterogeneous fleet of vehicles	35
3.2.3	Configurable vehicle capacity	35
3.2.4	Contributions with respect to the literature	36
3.3	Problem settings and mathematical model	36
3.3.1	Example	36
3.3.2	Problem settings	37
3.3.3	Mathematical model	39
3.4	Solution method	40
3.4.1	Matheuristic framework (LNS-RSCP)	41
3.4.2	LNS operators	43
3.4.3	Set covering problem (SCP)	44
3.5	Evaluation of the insertion of requests	46
3.5.1	Schedule evaluation	48
3.5.2	Capacity evaluation	50
3.6	Computational experiments	55
3.6.1	Description of instances	56
3.6.2	Evaluation of the metaheuristic components	57
3.6.3	Performance evaluation on benchmarks from the literature	59
3.6.4	Managerial insights	62
3.7	Conclusion	64
4	Weekly transport problem	67
4.1	Introduction	68
4.2	Literature review	69
4.2.1	Service time consistency models	70
4.2.2	Solution approaches for time-consistent routing problems	71
4.2.3	Contributions to the literature	72
4.3	Modeling the consistent dial-a-ride-problem (TC-DARP)	73
4.4	Solution method	76
4.4.1	The matheuristic framework (MSPP)	77
4.4.2	Solving the multi-period DARP (MP-DARP)	79
4.4.3	Mono-objective optimization procedure	83
4.4.4	Selection rules	85
4.5	Route generation: DARP with multiple time windows	87
4.5.1	Main process for the generation of new routes	87

4.5.2	Definition of time windows for the DARPmTW	88
4.5.3	Solving the DARPmTW	89
4.6	Computational experiments	92
4.6.1	Parameters settings	92
4.6.2	Description of instances	93
4.6.3	Evaluation of the metaheuristic components	93
4.6.4	Performance evaluation on benchmarks from the literature	95
4.6.5	Managerial insights about time-consistency and transportation costs	98
4.6.6	Economic impact route time shift	100
4.7	Conclusions	101
5	Assessment of transport pooling policies	103
5.1	Context	103
5.2	The clustering problem	105
5.2.1	Proximity measures	105
5.2.2	Clustering policies	107
5.3	Solving Approach	109
5.4	Input data	110
5.5	Computational experiments	111
5.5.1	Comparison proximity measures	111
5.5.2	Comparison of MSI clustering policies (C-MSI)	113
5.5.3	Comparison of user clustering policies (C-USER)	114
5.5.4	Global comparison	115
5.5.5	Fleet utilization	118
5.6	Conclusions	119
6	Conclusions and perspectives	121
6.1	Conclusions	121
6.2	Perspectives	123
A	Appendix Chapter 3	125
A.1	Parameters used for ALNS-based metaheuristics	125
A.2	Detailed results comparison of LNS-based algorithms	126
A.3	Detailed results benchmark Cordeau and Laporte (2003)	127
A.4	Detailed results benchmark Qu and Bard (2013)	128
B	Appendix Chapter 4	131
B.0.1	Construction of $\mathcal{TW}(\mathcal{H}, u)$	131
B.0.2	Parameters LSN-SCP	132
B.0.3	Parameters MSPP tests	132

TABLE OF CONTENTS

B.0.4 Scheduling algorithm DARP	133
B.1 TC-VRP	134
C Appendix Chapter 5	137
C.0.1 Parameters LNS-SCP	137
Bibliography	139

List of Tables

1.1	Geographical dispersion of users	6
2.1	Emergency vs non-emergency transport	20
2.2	Non-emergency transport	22
2.3	Literature review non-emergency transport of patients & passengers	28
3.1	Matrix representation of S^k for the vehicle presented in Figure 3.1.	54
3.2	Parameters (all defined in Section 3.4).	56
3.3	Vehicle types – costs	57
3.4	Vehicle types – configurations	57
3.5	Performance comparison of LNS-based heuristics for our instances (see A.2 for detailed results)	58
3.6	Benchmark classic DARP Cordeau and Laporte (2003) (Detailed results in Section A.3)	61
3.7	Benchmark Qu and Bard (2013) (detailed results in A.4)	62
3.8	Savings due to en-route configuration	63
3.9	Characteristics of vehicle types for fixed cost analysis	64
3.10	Costs of vehicle types for fixed cost analysis	65
4.1	Sets	74
4.2	Data	74
4.3	Variables	74
4.4	Overview of cases for the definition of \mathcal{TW}_u^t	89
4.5	Parameters MSPP (all defined in Sections 4.3-4.5).	92
4.6	Avg Gaps of MSPP variants for solutions with a C_{\max} of 3, 2 and 1 time-classes.	94
4.7	Avg Gap(%) with respect to the best hypervolume (5 runs)	95
4.8	Benchmark RconVRP instances from Feillet et al. (2014)	97
4.9	Results aggregated by percentage of users requests on Feillet et al. (2014) benchmark	98
4.10	Results aggregated by instance size on Feillet et al. (2014) benchmark	99
4.11	Economic implications of allowing a later departure of routes in solutions (Best solutions on 5 runs)	102
5.1	Global characteristics of the test data set	110

5.2	Vehicle list	112
5.3	Comparison MSI clustering policy	113
5.4	Comparison users clustering policy	114
5.5	Global comparison	117
5.6	Vehicle’s utilization	118
A.1	Parameters for the ALNS as in Masson et al. (2014)	125
A.2	Performance comparison of LNS-based heuristics	126
A.3	Results on the DARP instances Cordeau and Laporte (2003)	127
A.4	Benchmark Qu and Bard (2013) instance “A” scenario (iii) without groups	128
A.5	Benchmark Qu and Bard (2013) instance “B” scenario (iii) without groups	128
A.6	Benchmark Qu and Bard (2013) instances “B” ignoring walker requirements.	129
B.1	Parameters LNS-SCP (Tellez et al., 2018b).	132
B.2	Benchmark Feillet et al. (2014)	134
B.3	Benchmark Feillet et al. (2014)	135
C.1	Parameters LNS-SCP algorithm	137

List of Figures

1.1	The four stakeholders in the transportation for PWD	4
1.2	Transportation from the MSI perspective	6
1.3	Transportation from the user’s perspective	7
1.4	User’s preferences	7
1.5	Example of a reconfigurable vehicle (https://www.handynamic.fr/)	9
1.6	Transportation management processes	11
1.7	Structuring a typical week	12
1.8	Illustration of vehicle routes	13
1.9	Global approach to improve time-consistency	14
2.1	Healthcare transport classification	18
3.1	Vehicle with three configurations (http://handynamic.fr)	33
3.2	Comparison of routes with and without vehicle capacity reconfiguration	37
3.3	Insertion of request r into route ω	47

3.4	Graphical representation of a route load	51
3.5	Capacity of the vehicle presented in Figure 3.1.	53
3.6	Impact SCP component (Instance I01-80)	59
3.7	Reactive SCP (Instance I13-280)	60
3.8	Fixed cost impact of reconfigurable vehicles (Instance I13-280)	64
4.1	Solution with $C_{\max} = 2$ time-classes	70
4.2	Solution with $C_{\max} = 5$ time-classes	70
4.3	Pareto	77
4.4	Pareto with the two optimization procedures of one iteration (Algorithm 6 lines 7 and 8).	79
4.5	Example of projection and complementary routes	82
4.6	Example of 3 successive calls of the sequential selection rule with $N = 6$	86
4.7	Construction of $\mathcal{W}(\mathcal{H}, u)$	88
4.8	Comparison Pareto front instance C01_60	96
4.9	Pareto front C01_60	99
4.10	Pareto front C10_160	99
4.11	Pareto front C12_280	100
4.12	Scaled Pareto fronts	101
5.1	CURRENT clustering policy	105
5.2	Proximity between points (left) and users (right)	106
5.3	Clustering MSIs	108
5.4	Clustering of users (C-USER)	109
5.5	Solving approach	110
5.6	Maximum ride time function	111
5.7	Comparison of proximity measures	113
5.8	Comparison clustering policies	116
5.9	Assignment of MSI to clusters	116

Introduction

From 2010, healthcare transportation in France has become one of the top ten priorities of the risk management plan due to the increasing cost of these transports. For social and Medico-Social Institutions (MSI), this cost represents the second biggest expense after that of the wages. In this context, this thesis as part of the NOMAd project aims to improve the daily transport service for people with disabilities between their home to MSIs. To this end, we suggest the transport pooling between several MSIs on one side and a global optimization strategy on the other side. This strategy makes possible to group and optimize routes on a given geographical area. The challenge is then to improve economic performance while maintaining social and environmental goals. A decision aiding tool for transport optimization is proposed for this purpose.

Keywords: decision support systems, transport pooling, passenger transport, optimization, healthcare

Part of this chapter have been published in a conference paper and an extended version submitted to a journal:

Tellez, O., Daguet, L., Lehuédé, F., Monteiro, T., Montoya, G. O., Péton, O., Vercraene, S., 2019. A stakeholder oriented approach to the optimization of transports of people with disabilities. Supply Chain Forum: An International Journal (Submitted)

Tellez, O., Daguet, L., Lehuédé, F., Monteiro, T., Montoya, G. O., Péton, O., Vercraene, S., 27-29 August 2018a. Étude pour l'analyse et l'optimisation du transport des personnes en situation de handicap. In: Conférence Francophone en Gestion et Ingénierie des Systèmes Hospitaliers (GISEH). Geneva, Switzerland

1.1 Context

The medico-social sector

In 2010, medical transportation became one of the top ten priorities of the risk management plan of the French insurance scheme due to the increasing cost of these transports ANAP (2013). In France, the «transport sanitaire» includes both the medical transportation and the medico-social transportation. This sector serves both elderly and people with disabilities. The report “Establishments and services for disabled people - Offers and needs, financing modalities”, published by the General Inspectorate of Social Affairs & General Inspection of Finance estimates the transportation cost for all medico-social institutions at 10% of the total cost of medical transportation (Laurent et al., 2012). Thus, in 2017, the medico-social transportation cost represents 500 million euros, at the national level, for all medico-social institutions (Gonzalez et al., 2018). There are few details on the evolution of this cost. Between 2006 and 2014, this cost has increased by 40% (Wahl et al., 2016). There are few statistics on the evolution of this cost.

Medico-Social Institutions

For Medico-Social Institutions (MSI), transport costs often represent the second largest expenditure after that of the staff (ANAP, 2016b). Despite this, few MSIs have the skills and means to effectively manage the transport, provide high quality service and keep the costs down (ANAP, 2016b). With little expertise, MSIs are often led to adopt sub-optimal practices that impact costs, quality of service or the environment. Unlike medical transportation, medico-social transportation is generally integrated into the life project of People With Disabilities (PWD) (ANAP, 2016c). Transportation is no longer considered merely a painful or non-value-added activity. It allows, the inclusion of PWD to the school or work life. It can be a lever for the autonomy these people (ANAP, 2016b).

Transportation demands in the medico-social sector are mostly regular and very often known in advance. This anticipated knowledge of needs can be used to jointly improve: the punctuality, quality of service for users, and the efficiency of the resources mobilized. The success of such a project requires the mobilization of all stakeholders in order to consider the different concerns of the parties involved. We must indeed meet the needs of people with disabilities and family carers, accompanying persons and mobility centers, MSIs and professional carers.

The following sections are organized as follows: Section 1.2 presents the NOMAd project. Then, Section 1.3 describes the field study in three parts: the stakeholders, a needs analysis drawn from a survey conducted in the fall of 2017 and a summary of the special features of the transport for PWD. Section 1.4 illustrates where this thesis is positioned within the general transportation process. Section 1.5 presents the road map and contributions of this thesis. Finally, the conclusions of this chapter are presented in Section 1.6.

1.2 The NOMAd project

The project called “Numérique et Optimisation pour une Mobilité Adaptée¹” (NOMAd) is a 3-year project supported by the European Union through FEDER Funds. Deployed in the Auvergne Rhône-Alpes region, NOMAd aims to develop a web service to facilitate exchanges among all the stakeholders of adapted transport: transporters, families of PWD and MSIs. The aim is to build the transport solution by jointly integrating the points of view of the different actors (see <https://nomad DISP-lab.fr/>). Three objectives are targeted:

- *Optimization of the transportation costs* by reducing the number of kilometers traveled and the number of vehicles needed.
- *Improvement of the quality of service* for passengers and establishments by reducing travel times and provide time-consistent schedules.
- *Reduce the environmental impact* by reducing CO2 emissions linked to the number of kilometers traveled.

These objectives are interlinked and have been considered conjointly. As proposed in Zhu et al. (2018), to be sustainable, the environmental impact reduction has to be associated with a cost improvement.

This project is carried out by a group of researchers from the DISP² laboratory in collaboration with researchers from LS2N³ laboratory and two strategic partners of the medico-social sector:

Ressourcial: a cooperative structure specialized in information systems for the medico-social sector. Created as an association, it is intended to share management practices and IT tools to non-profit MSIs.

GIHP Service Adapté: a transportation company specialized in the management and realization of paratransit services. GIHP intervenes particularly in the agglomeration of Lyon. To give an order of magnitude, this structure carries up to 1500 people per day. One of the benefits of GIHP is that its stakeholders are the MSIs, which implies that making profit is not the only objective nor priority.

1.3 Field study

During the interaction with different stakeholders involved in the transportation for PWD we could understand the needs of the sector. In this section we present first the role of stakeholders, then a needs analysis study drawn from a survey conducted in the fall of 2017 and finally a summary of the special features found in the transportation for PWD.

¹In english “Digital and Optimization for Adapted Mobility”

²<https://disp-lab.fr/>

³<https://www.ls2n.fr/>

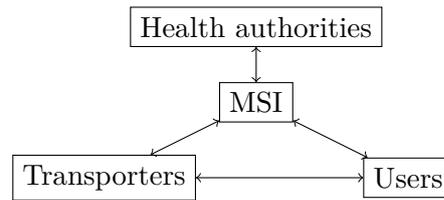


Figure 1.1: The four stakeholders in the transportation for PWD

1.3.1 Stakeholders

In the process of transport for PWD there are multiple interdependent stakeholders assuring the reliability, quality and efficiency of the transportation service. As shown by Chakraborty (2018), a well-design interdependence can bring positive impact on the level of commitment between stakeholders.

Stakeholders can be classified into 4 categories: Health authorities, MSIs, Transporters and Users as shown in Figure 1.1. Arrows represent the direct interaction between stakeholders. Each category can include the coordination of multiple stakeholders.

Health authorities

Health authorities play the role of regulators and very often finance the transport itself. In France, most of MSIs are financed with public funds. This funds come from different sources, named: the National Solidarity Fund for Autonomy (CNSA) and health insurance (Assurance Maladie), and in a smaller proportions the State and Department councils.

Medico Social Institutions (MSIs)

MSIs accompany PWD daily in the construction of a life project that will allow PWD develop their autonomy. Some PWDs may stay in boarding house the whole week, but the majority come back home daily. Each MSI receive a global budget to finance all its expenses, including transportation. Thus, there is an incentive to optimize costs without compromising the service quality. The transportation activity can be so cumbersome for MSIs that outsourcing is often seen as a promising solution. However, with an external transporter the coordination can be difficult and the quality of service may be compromised.

Transporters

The transportation activity can be internally organized by the MSI themselves or outsource to a specialized provider. In both cases, the transporter must follow strict regulations regarding safety measures, driving competences and vehicle specifications. The two main tasks for a

transporter are: the construction of the annual transport planning and the daily operation of the transportation itself taking into account unforeseen events.

The transportation planning is a complex task that requires decision support tools. However, the available solutions in the market are scarce and do not completely adapt to the needs of the medico-social sector. Currently, complex tasks like route optimization are performed manually, as in GIHP, based on field experience only. The project NOMAd is meant to fill these needs on in comprehensive manner.

Users

Users are PWD and their families who are very often the spokesperson. Families facilitate the transport of their relatives and stay in direct contact with MSI and the driver. In a certain way, families verify that the transportation is done with the expected quality.

1.3.2 Survey: needs analysis

In 2017, we conducted a survey with 30 managers of MSI (12 urban, 10 suburban and 8 rural) and 35 users (33 parents and 2 PSW) from the Auvergne Rhône-Alpes Region in France. This survey provides us an overview of the current situation of the adapted transportation from two different angles, from the users perspective and from the service provider's perspective. The service providers are MSIs that can outsource the transportation activity. Nearly half of the MSIs surveyed outsource the transport.

Service provider's perspective

Figure 1.2 shows how MSIs perceive the quality of their transport and the resources they deploy in the transportation management. On one side, 77% of MSIs think they provide a good quality transport (yes + rather yes). This perception shows the importance given to the transportation activity. On the other side, this attention is confirmed with the resources deployed in the transportation management: 68% of MSIs put a great effort in the management of transport (significant + high levels) as shown in the graph on the right.

For MSIs, the transportation activity is not just a logistics service, but it is integral part of the life project of PWD. The main resource required by the transportation activity is the staff necessary to the reception and departure; the realization of the transport itself, if it is not outsourced; and the staff for transportation management, gathering needs, planning routes and managing interactions with family caregivers on a daily basis. The last activities are often perceived as highly time-consuming and without real added value.

MSIs were also asked about the approximate distance of users to MSIs. Table 1.1 summarizes this distance in kilometers and minutes. Half of users live more than 30 km away from their MSIs and make more than 1 hour of ride time. Moreover, there is high dispersion of demands reflected

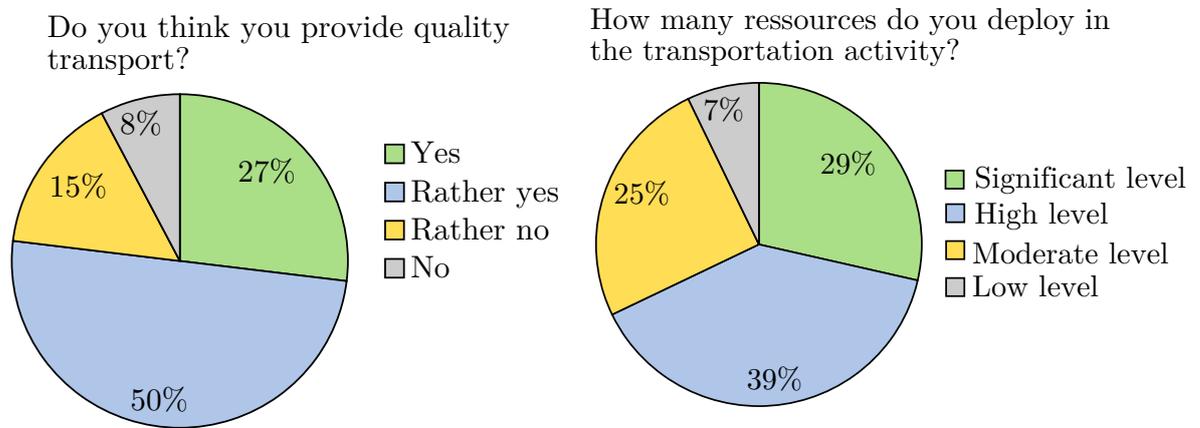


Figure 1.2: Transportation from the MSI perspective

in a standard deviation of 46 km and 43 minutes. Note that these values are approximately estimated and sometimes overestimated by some administrators. This condition reinforces the interest to consider the transport pooling among several MSIs. This strategy is barely implemented in current practices. According to Rouquet and Vauché (2015), this type of pooling is known as District pooling, and leads to a more collective and stronger supply chain.

	Distance (km)	Time (min)
Median	30	60
Mean	48	74
Standard deviation	46	43

Table 1.1: Geographical dispersion of users

Users' perspective

Figure 1.3 shows the users view point regarding service quality on the left side and how users perceive their current ride time on the right side. Note however, that most of the responses come from parents as a spokesperson of PWDs. The perception of the service quality is very good 94% ("yes" + "rather yes"). However, there is a 6% of users completely unsatisfied. This situation is reflected in the graph of the ride side with 15% of users unsatisfied with their current ride time ("no" + "rather no"). This result shows us that one way to improve the quality of service is by reducing long ride times.

Users were asked to give a level of importance to different improvement directions. Figure 1.4 presents the results sorted from the most to the least important aspect. The most important aspect for users (parents) is to stay informed about modifications such as driver or scheduling changes (69% very important). Users give more importance to the service punctuality than to

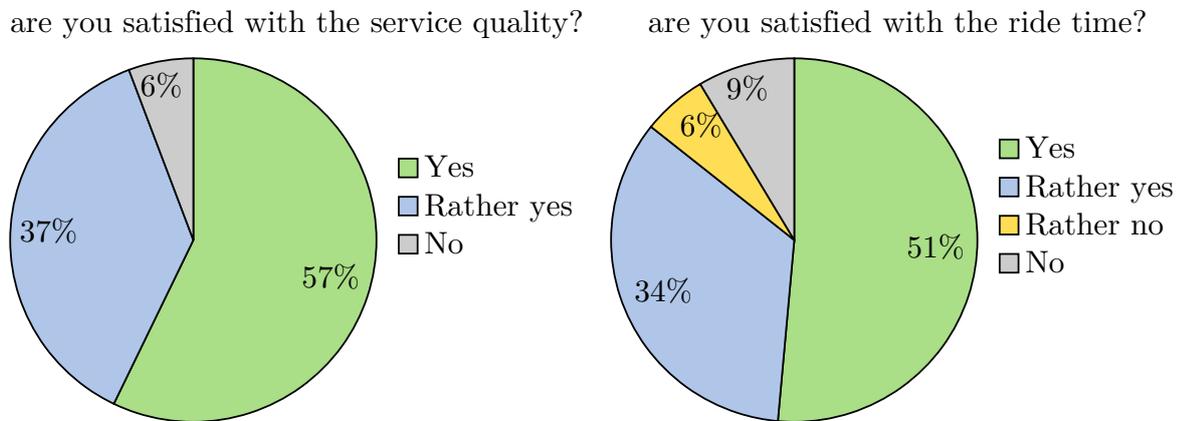


Figure 1.3: Transportation from the user's perspective

have prior information about the vehicle arrival. Time-consistency, which is having the same service time every day, seems to be a bit more important than driver consistency which is having the same driver every day.

	Very important	Important	Moderate importance	Low importance
Stay informed about unexpected events	69%	29%	3%	0%
Service punctuality	43%	49%	9%	0%
Time consistency	34%	40%	17%	9%
Driver consistency	31%	31%	23%	14%
Information about vehicle arrival	14%	29%	46%	11%
Make suggestions	11%	71%	11%	6%

Figure 1.4: User's preferences

Open questions

In the survey to MSIs, we also address open questions regarding difficulties and possible actions to improve the transportation system. Regarding the main difficulties found in the transportation activity, managers of MSI point out the elements already seen above, ride times and transportation costs. They mention several other topics of concern:

- First, the need to take into account the specificity of this type of transport by adapting it to the PWD's needs. The necessity to manage behavioral problems in PWD like potential crises during transport.
- Secondly, the subject of communication and coordination especially for the management of unforeseen events, changes and delays. The coordination among 3 or more stakeholders and the need of real-time information about the transportation is a real concern for managers.

- Finally, managers evoke the human dimension of this transport. The need of a considerable number of staff in order to ensure the service quality and the safety. The specific driver training with respect to the qualifications and stability.

In regard of the actions to improve transportation, we found that 16% of the MSIs had integrated transport-related objectives into their long-term contract with the health authorities (in France known as CPOM). MSI leaders mention the following objectives:

- To transform the transportation into a lever of autonomy for PWDs.
- To reduce operating costs, particularly those related to vehicle maintenance.
- To reduce ride times by the transport pooling among multiple MSIs.
- To adequately treat all the transport demands in the territory.
- To transform the transportation activity into an enabler of the PWD life project rather than a hurdle to his/her personal life project.

This field survey allowed us to define the three main objectives of the NOMAd project. It is about controlling costs, the quality of service and the ability to communicate and coordinate actions among multiple MSIs.

1.3.3 A multidimensional problem

The management and optimization of the transportation for PWD is a complex task. It requires taking into account a set of variables and features related to people, vehicles and the organizational scheme.

Beneficiaries

Beneficiaries are the PWD transported. They can be in a wheelchair or in a seat. PWD have a certain degree of autonomy but some have to be accompanied by a third party. From a quality of service perspective, the time dimension is very important. Users have availability constraints that must be respected. These constraints are expressed in the form of time windows with an earliest and latest arrival time. A maximum ride time of transport for each person is taken into account. Furthermore, in the design of routes a consistent schedule must be taken into consideration. Indeed, despite the variability of demand during the week, it is better to give a single time of service in the morning (afternoon) when they are present several days during the week.

Vehicles

Vehicles used in the transport for PWD are specifically designed. Each vehicle can receive a limited capacity for people in wheelchair and/or seated. Additionally, one vehicle may have different layouts that can be configured before the departure or during the routing. These configurations are made with retractable seats, for example, to swap a wheelchair space with a conventional seat. Figure 1.5 shows four different layouts of a reconfigurable vehicle. This vehicle can have spaces for seats, wheelchairs and electric wheelchairs(in black).

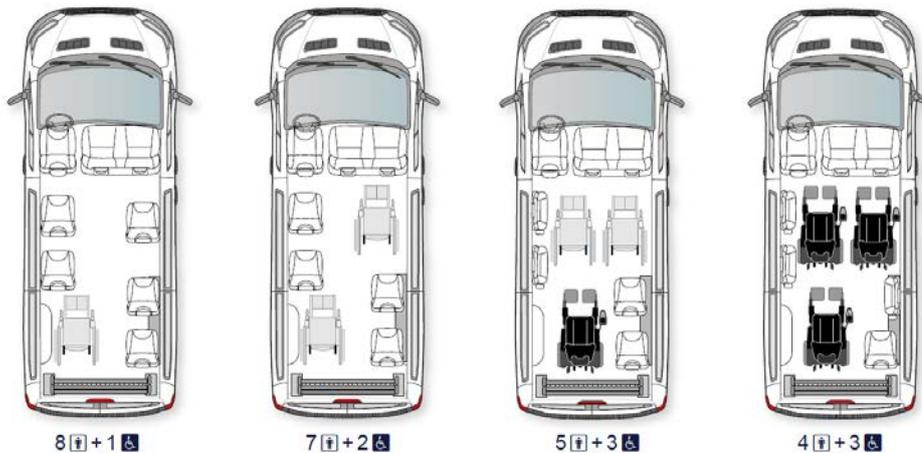


Figure 1.5: Example of a reconfigurable vehicle (<https://www.handynamic.fr/>)

Co-organization

Stakeholders are characterized for having different, even antagonistic, objectives. A transport management based solely on a financial criterion will tend to reduce the number of vehicles mobilized and to offer rather long routes, which is a sub-optimal policy from quality point of view. Conversely, an organization of a transport seeking only to offer the best quality of service will tend to increase the number of vehicles and to offer direct-route transport for each person. Thus, it seems necessary to find a compromise solution. A multi-criteria approach is therefore required to take into account together cost reduction, environmental impact and quality of service criteria.

1.4 Transportation management processes

There are multiple processes involved in the management of transport for PWD. Figure 1.6 shows the Business Process Modeling (BPM) of the main transportation activities. In it, we model the interaction among the MSI, the transporter and users in order to perform 3 main

processes. (i) *The transport mutualization* to determine the opportunities of transport pooling between MSIs;

(ii) *the long-term planning processes* defining the annual transportation plan of each user and logistics need like the fleet; (iii) *the short-term planning processes* to manage the day-to-day modifications of the transportation plan; and, (iv) *the transport execution process* covering the daily iterations between stakeholders and the follow up of routes. The scientific results of the following section respond to the need of decision support tool for the long-term planning. It is bounded with a dotted square in the Figure 1.6.

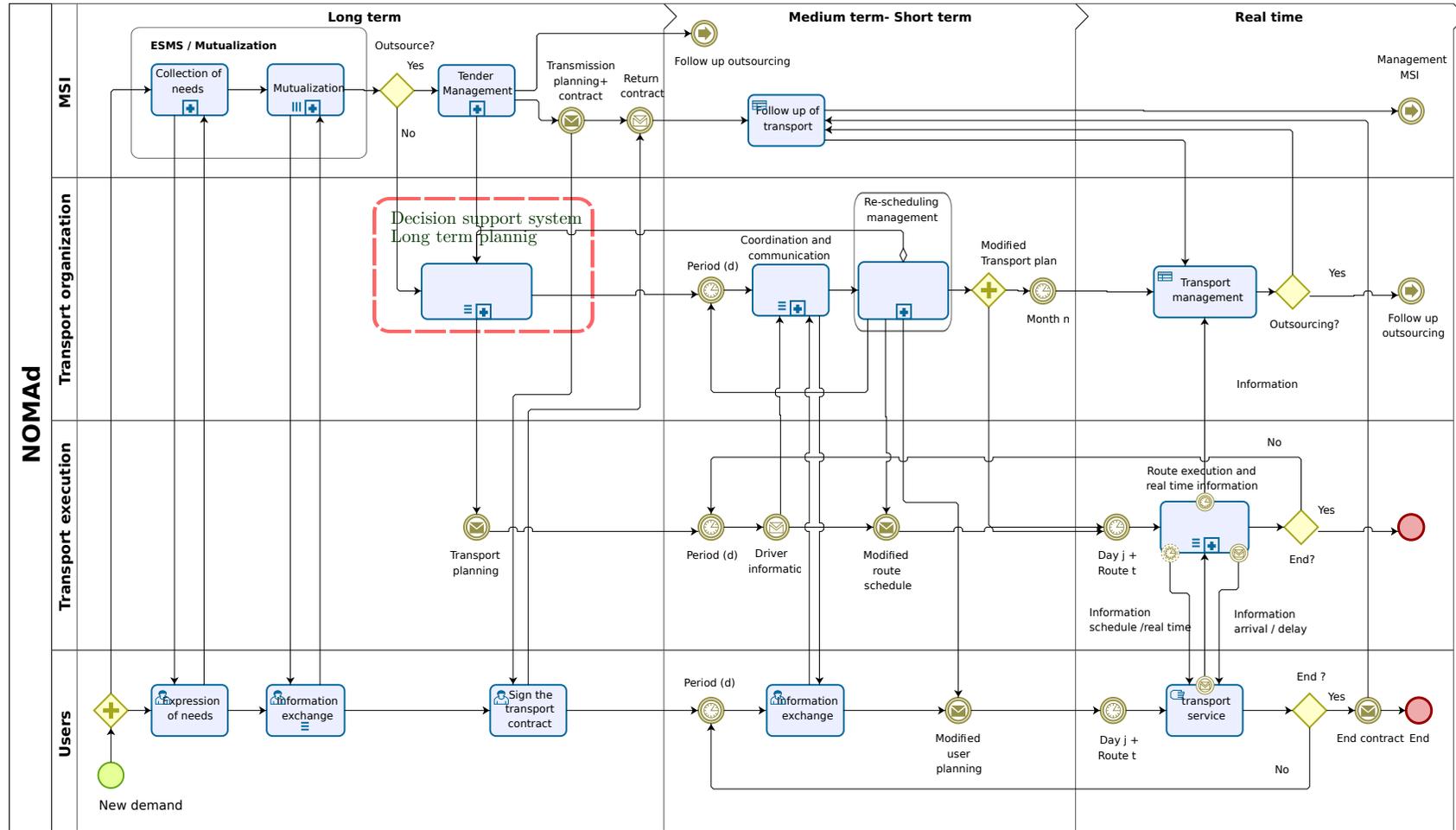


Figure 1.6: Transportation management processes

1.5 Road map and contributions

This thesis is structured in 5 chapters. The first four chapters are focused on the construction of the long-term transportation planning while the last one is dedicated to the assessment of transport pooling policies.

1.5.1 Construction of the long-term transportation planning

The construction of the long-term transportation planning is a complex problem that has not been completely solved in scientific literature. A literature review about the transportation problems in the health sector is presented in Chapter 2. The long-term planning is defined for one academic year. In accordance with practice, we assume that this transport plan is constructed from a pattern of identical weekly transportation plans.

Defining a weekly schedule is a complex task that needs to be decomposed into several sub-problems. First, we work in the planning of each sub-problem called the half-day transport problem. Secondly, we focus on building the weekly schedule from the sub-problems that compose it. Figure 1.7 illustrates the process. The rectangle M_1 corresponds to one sub-problem related to the half-day problem. The dotted rectangle specifies the perimeter of the weekly morning or afternoon schedule. We therefore assume the problem of the morning and the afternoon can be solved independently.

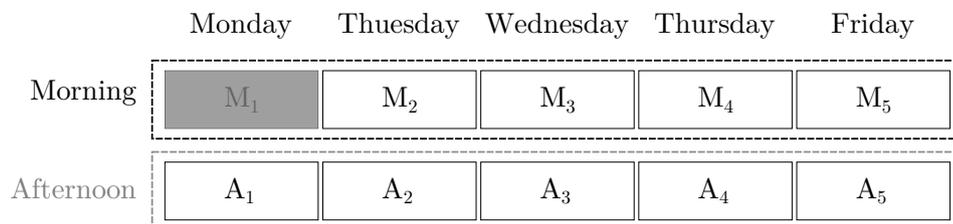


Figure 1.7: Structuring a typical week

The half-day transport problem

Defining the half-day schedule consists in solving an extension of the so called Dial-a-Ride transportation Problem (DARP). In the DARP each transported person is characterized by having a different pickup and delivery location, two time windows and a maximum ride time. Vehicle capacity constraints are also integrated. The problem of on-demand transport for PWDs is already covered in the scientific literature (Toth and Vigo, 1997; Parragh, 2011; Lehuédé et al., 2014). However, our problem considers specific transport elements such as the typology of transported people (wheelchairs or seated) and the typology of adapted vehicles that can be reconfigured en-route.

The transport optimization of the half-day problem is based on a cost objective that incorporates vehicle depreciation, hourly costs (eg driver wages), and mileage costs. The respect of pickup and delivery time windows and maximum ride time is expressed as timing constraints. Figure 1.8 illustrates two morning routes for transporting PWS to MSIs. Both routes have benefited of the transport pooling between the two MSIs. This problem is known as the fleet size and mix dial a ride problem with reconfigurable vehicle capacity (FSM-DARP-RC) which is a contribution of this thesis and that will be presented in Chapter 3.

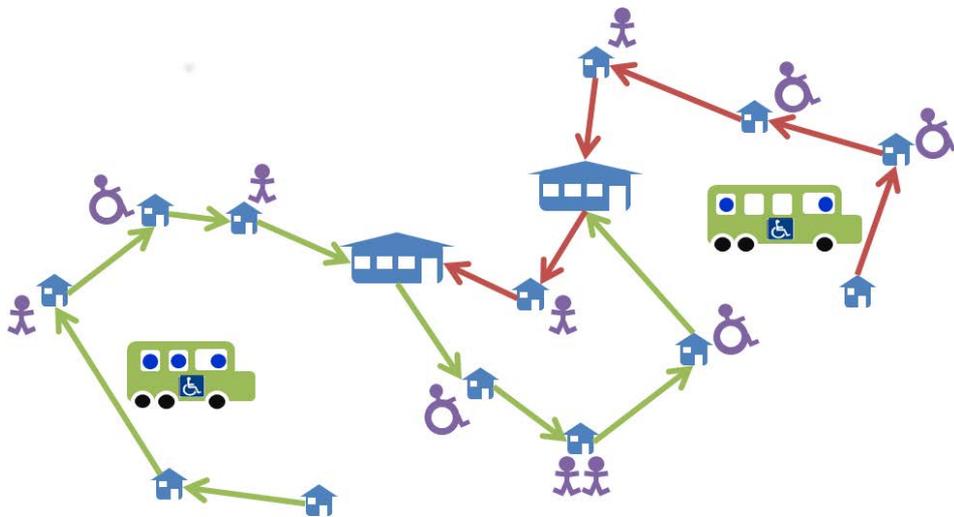


Figure 1.8: Illustration of vehicle routes

To solve this problem we proposed the matheuristic LNS-RSCP. This matheuristic combines a Large Neighbourhood Search (LNS) procedure with a Set Covering component to find high-quality solutions at a reasonable time. This matheuristic integrates innovative components related to the vehicle selection and the capacity check procedures able to determine if a given reconfigurable vehicle can perform a given route with a maximum number of reconfigurations.

Weekly transport problem

Once we solved the half-day transport problem, we conduct a weekly solving. Indeed, not PWD need to be transported every day. The simple juxtaposition of half-day transportation schedules can give a very cost-efficient transport planning. However, in this plan, users may have a different service time every day. This situation is uncomfortable for PWD, or even unacceptable for some people with mental disabilities. Thus, from a global approach this weekly schedule aims to optimize both transport costs and time-consistency throughout the week.

The problem of time-consistency has already been treated in the scientific literature for a transportation problem close to ours but simpler called “Time-consistent Vehicle Routing Problem”. It is characterized by a fleet of homogeneous vehicles, a common destination place,

and no service time constraints (Feillet et al., 2014). The time-consistency for one user is defined by the number of different time slots during the week. In this approach, one time slot can hold several service times within 10 minutes to each other. If one user has a different time slot each day, he has therefore 5 time-classes. A perfectly consistent transportation plan is characterized by a single time-class for each PWD. This condition is very convenient for the user but very expensive for MSIs. Considering a consistency objective in addition to the cost objective to build the weekly schedule is another contribution of this thesis that is presented in Chapter 4. This problem is denoted the time-consistent DARP.

Figure 1.9 illustrates the proposed solving approach for the time-consistent DARP. We begin by solving the sub-problem of each half-day described in Section 1.5.1. The union of these solutions is the starting point for the construction of the weekly transportation plan. This first plan is economically efficient, but highly inconsistent. During this first step, we collect the routes obtained in the different iterations to constitute a pool of routes useful for the improvement of time-consistency. Then, using an optimization MILP solver, a time-consistent DARP is solved using a different subset of routes from the pool at each iteration. This process is repeated until the desired pattern is achieved. A Pareto front is constructed using an epsilon-constraint approach. Each new solution of the front is found by relaxing the cost of the previous solution by an epsilon quantity to gradually improve the time-consistency.

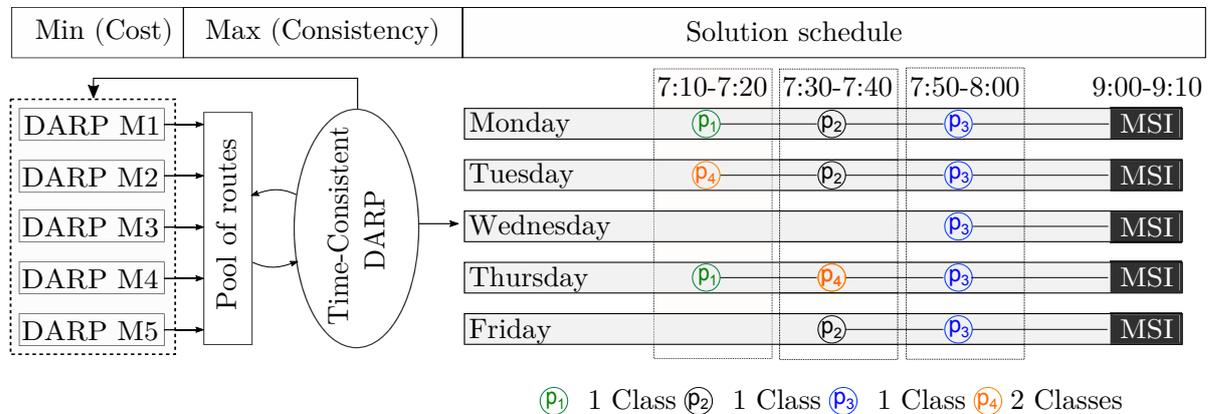


Figure 1.9: Global approach to improve time-consistency

1.5.2 Assessment of transport pooling policies

The transport pooling between MSIs is both a research and a managerial challenge. Currently, the transport operation is managed independently for each MSI. In contrast, the management of multiple MSI at a time in addition to increasing the size of the problem (complexity), presupposes among other things: to have an economic interest to be shared and to have a common mechanism of management and communication between partners.

At the strategic level, the study of Chapter 5 aims to evaluate the marginal gain of inter-institutional transport pooling. Alternative policies with different pooling levels are proposed. The management objective of this study is to provide the elements to open up a serious discussion between managers of MSIs in order to go further in the development of this strategy.

At the operational level, this problem corresponds to a clustering approach for solving large scale routing problems. One of the challenges in the clustering step is to define a relevant measure to determine the proximity between user demands. Different metrics from the literature will be tested in order to identify which one works better in each way of clustering.

1.6 Conclusion

The transportation management and optimization for PWD is a complex task. It requires taking into account a set of variables and particularities related to people, vehicles and organizational schemes. For MSIs, transportation management is a highly demanding task and very often requires a lot of human resources to ensure the expected quality and safety. Knowing that this transport is characterized by a strong geographical dispersion of users, there is a real economic interest in optimizing and pooling this service. This must be done without degrading the quality of service. To this end, we propose a comprehensive approach for developing a transport planning that is both cost-effective and time-consistent for each PWD.

There are five chapters in this thesis corresponding to five major contributions. In this chapter, we have presented a characterization of the transport sector for PWD, the needs in transportation and the global approach for optimizing the long-term transportation planning. Chapter 2 presents a literature review of the non-emergency transport of patients and PWD in the context of healthcare transport. Chapter 3 focused the modeling and solving approach of the half-day transport problem. Chapter 4 presents the modeling and solving approach of the weekly transport problem. Finally, Chapter 5 addresses the question of transport pooling between several MSIs. A comparison between different pooling policies is presented in regard to transport costs and service quality.

Literature review

The transport for people with disabilities is commonly studied outside of a medical transport context. However, in practice, it is usually part of the public medical transport or home care services. Healthcare transport is usually related to emergency medical transport. However, a large percentage of patient transport is of non-urgent nature (Hains et al., 2011). In this chapter, we present a demand-based classification for healthcare transport services. A literature review about the optimization problems in the context of non-emergency transport of patients & passengers is carried out with a focus on the decision problems found at the strategic, tactical, operational and real-time levels.

Keywords: healthcare, non-emergency transport, patients transport, optimization, dial-a-ride problem

2.1 Healthcare transport

In France, the transport for People With Disabilities (PWD) is part of the so called «transport sanitaire». According to French healthcare care system, it concerns “*any transport of a sick person, injured or parturient, for reasons of care or diagnosis on prescription or in case of medical emergency, carried out by means of terrestrial, aerial or maritime transport*” (ANAP, 2013). More generally, health transport includes the transport of patients, passengers with disabilities, medical-related supplies and professionals in the case of home care services. In this chapter, patients and passengers with special transport needs will be referred to as P&P.

Healthcare transport can be divided into emergency and non-emergency transport. The first case is when the life of a person is in danger and the second one is, for example, the transport of patients between hospitals, the daily transport of PWD and home care. Non-emergency cases

manage both short term patients with temporal illness (occasional) and long term patients with chronic illness or disability (regular). Figure 2.1 presents a global classification of healthcare transport based on the nature of the demand. The following sections present the characteristics of each category. The daily transport of PWD studied in this thesis is part of the non-emergency regular transport of patients as shown in gray.

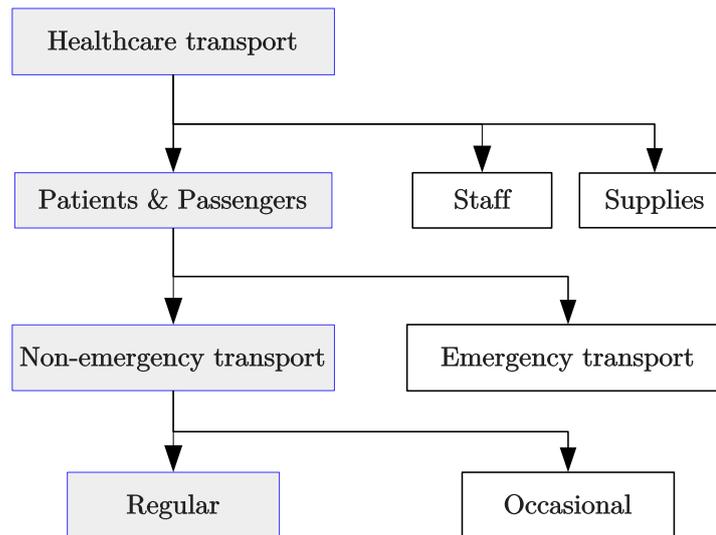


Figure 2.1: Healthcare transport classification

2.1.1 Transport of staff, supplies, and patients & passengers

Healthcare transport continuously moves staff, medical supplies, and P&P all along the supply chain. Although the different type of flows is interrelated, each type of flow is so different that needs to be studied independently.

Staff: the transport of professionals to the patient’s residence place for support services is known as home care. This service is provided for patients (i.e. elderly, disabled) needing assistance for daily living or managing health issues. There are three types of home care services: (i) non-medical care to support everyday activities such as bathing, dressing, household, shopping, etc.; (ii) long term nursing care to support patients with chronic illness, injury or disability. (iii) short term care or home health care which is physician-directed care designed to help a patient prevent or recover from an illness, injury, or hospital stay. In-home care services, a caregiver takes care of multiple patients during the day. The sequence of patient visits can be optimized to reduce transport costs. Moreover, home care routing should consider the patient’s preferences and the nurse working shift. Recent developments in the optimization of home care routing and scheduling are presented in Fikar and Hirsch (2017).

Supplies: Hospitals manage a complex distribution network of supplies such as food for patients, sterilization, patient's specimens, pharmaceuticals, and medical records. On a daily basis, vehicles, supply carts, personnel move between storage areas and points of care to continuously supply the demand of healthcare facilities (Landry and Philippe, 2004). From a logistics point of view, supplies need being transported with the right quality at the right time and place. Medicine shortages and improper use of pharmaceuticals can not only lead to financial losses but also have a significant impact on patients (Uthayakumar and Priyan, 2013). The coordination of stakeholders around inventory management is a key component for attaining these objectives effectively (Kelle et al., 2012).

Patients & Passengers (P&P): the transport of patients in hospitals, commonly known as intrahospital transport, include medical transfers between healthcare facilities specialized in diagnostics or medical treatment. Patients have fixed appointments, such as x-ray, ultrasonic, blood testing or surgery. For medical reasons some patients are not able to go on their own, and need to be escorted by trained staff (Turan et al., 2011). Hence, the transport for going and return needs to be scheduled for these patients. In campus-based hospitals, intrahospital transport is provided by appropriate vehicles, generally, ambulances, which several patients often share (Hanne et al., 2009). Some PWD living at home or in boarding schools need specialized transport for moving to different places such as school centers or workplaces. They usually go every day from home to Medico Social Institutions (MSI) in the morning and go back home in the afternoon (Qu and Bard, 2013).

The following section defines the types of transport for P&P in a general context before detailing in the non-emergency transport of P&P which is the focus of this chapter.

2.1.2 Emergency and non-emergency transport for patients & passengers

Healthcare transport of P&P differentiates from both emergency and non-emergency cases. The first case is when the life of a person is in danger and the second one is, for example, the transport of patients between hospitals, the daily transport of PWD or home care. Over the past years, much attention has focused on the emergency transport of acute and critical-care patients (Hains et al., 2011). However, a large percentage of patient transport are of a non-urgent nature (Robinson et al., 2009; Hains et al., 2011).

Emergency transport in the healthcare context is commonly known as emergency medical transport. Emergency transport is usually referred to the transport of patients whose conditions are life-threatening or time-critical (Huggins and Shugg, 2015). It is mainly performed by ambulances after emergency calls due to an accident or a malfunction of the body. Emergencies are highly unpredictable in magnitude or location. The response time must be as short as possible because the person's life is in danger. There are two emergency management systems,

the Franco-German and the Anglo-American systems. In the Franco-German system, most patients are treated on the scene because care providers are medical doctors supported by paramedics (Dick, 2003). In the Anglo-American system, however, very few patients are treated on-scene as care providers are paramedics with medical oversight. The Anglo-American model, ambulances should be located in strategic places to be able to arrive quickly to the incident site. This model is known as “scoop and run” because the aim is to rapidly bring patients to the hospital with less pre-hospital interventions (Al-Shaqsi, 2010). In both cases, the expected level of service is very high and therefore very expensive. Thus, the transport cost is more of a budget constraint than a goal by itself. A review of the recent developments in the optimization of emergency medical services can be found in Saghafian et al. (2015); Bélanger et al. (2019).

Non-emergency transport However, a wider view of non-urgent transport concerns also the daily transport for PWD and seniors to and from specialized centers. Reports from several countries show that non-urgent transport is continuously growing at significant levels (ANAP, 2016b; Hains et al., 2011). Unlike emergency transport, non-emergency cases can be scheduled at a later time depending on the needs and resources’ availability. In some contexts, demands are known in advance. This anticipated knowledge can be used to optimize transport costs by pooling demands and increasing vehicle utilization. This is at the cost of a longer response and ride times. In other context, demands arrive in an on-going fashion and dispatchers should propose quickly a schedule or reject the request if there is not available schedules with the current status of the system.

Table 2.1 summarizes the differences between emergency and non-emergency transport services.

Characteristics	Emergency	Non-demand
<i>Transport</i>	Immediate	On-availability
<i>Demand</i>	Unknown	Unknown or known in advance
<i>Transportation cost</i>	High	Medium-low
Fleet utilization	Low	Medium-high
Operating cost	High	Medium-low
<i>Service quality</i>	High	Medium-low
Response time	Short	Short-long
Ride times	Short	Medium-long

Table 2.1: Emergency vs non-emergency transport

The non-emergency transport for P&P covers a wide variety of applications from the leisure transport for PWD to the long term transport of chronically ill patients. Non-emergency transport can be divided into subcategories according to the frequency of demand. The following section presents the characteristics of non-emergency occasional and regular transport.

2.1.3 Occasional and regular transport for patients & passengers

Non-emergency services can differ depending on the frequency of demands. The transport of a patient with temporal illness is called “occasional” while the transport for patients with long term care is called “regular”. Occasional transport concerns non-recurrent P&P while regular transport concerns P&P with frequent and periodic demands.

Occasional transport refers to the transport of non-recurring patients. It is usually the case of punctual medical appointments for patients or the transport of extra-curricular activities for PWD. On a wider scope, this transport type is known as Transport On Demand (TOD). Demands are called “static” if they are known before the scheduling phase or “dynamic” if demands are scheduled as they arrive (Pillac et al., 2013). Static cases allow higher cost optimizations but usually take longer solving time. Dynamic solutions rather look for quick and cheap feasible schedules based on the current status of the system. However, real-life TOD systems manage both static and dynamic cases at a time.

TOD systems are characterized by having three often conflicting objectives: maximizing the number of requests served, minimizing operating costs and minimizing user inconvenience (Cordeau et al., 2007). In the context of patient transport, these objectives have different names. In healthcare transport, three common objectives are the number of patients served, transport costs and individual/collective preferences. Sometimes one objective can prevail over the others depending on the application. Rasmussen et al. (2012) for example, in an application to the Danish home care service for seniors and disabled citizens, consider as a primary objective the number of served patients, the visit preferences as secondary and lastly the operating costs. For PWD, preferences are usually expressed in terms of the desired pickup and delivery time (Jaw et al., 1986).

Regular transport concerns recurrent demands of patients during a planning horizon. Most of PWD are transported daily to medico-social institutions or twice a week for those in boarding houses. Patients with a long-term illness such as dialysis, tracheotomy or ventilator care that is treated on a regular basis at home or in the hospital. The planning horizon for regular transport is long, the school year for PWD. With demand information well in advance, transporters can optimize costs by pooling resources. However, regular transport often assumes additional objectives such as staff-consistency and time-consistency.

Staff-consistency in a medical context refers to the fact that patients expect to be followed up by the same caregiver (Allaoua, 2014). In the transport of PWD it is known as driver-consistency (Braekers and Kovacs, 2016). PWD with autism, for example, hardly supports the change of driver. At most, they tolerate a reduced pool of drivers acknowledged beforehand. Time-consistency refers to having a similar service time every time. For regular patients is highly desirable to have consistent visit schedules. For children with disabilities, families are

not ready to accept highly inconsistent schedules during the week (Feillet et al., 2014). Some PWD cannot tolerate irregular time schedules.

Table 2.2 summarizes the main characteristics of occasional and regular transport services. Note, however, in practice we can find hybrid transport systems that combine aspects from both regular and occasional transport. In the following section, we present optimization problems from the scientific literature in the context of non-emergency transport of P&P. Transport problems were classified according to the decision level.

	Occasional transport	Regular transport
Frequent objectives	Number of served patients Transportation costs Individual/collective preferences	Transportation costs Time-consistency Staff-consistency
Planning horizon	Short and medium term (i.e. hours, days)	Long-term (i.e. months, year)
Operating cost	Medium-high	Low-medium
Examples	Punctual medical appointments Extra-curricular activities PWD	Patients with long-term treatments Daily transport for PWD

Table 2.2: Non-emergency transport

2.2 Optimization problems in the non-emergency transport of patients & passengers with disabilities

Different types of problems are found in the optimization of non-urgent transport of P&P. Some problems deal with the whole transport system while other problems only a part of it. For example, the selection of the vehicle fleet can greatly influence the transport costs in the long-term. However, the re-scheduling of transport routes concern a small group of patients during a short time period. Healthcare systems differ from production systems in having constraints and objectives related to the service quality of patients. There are four decision problems in the transport optimization of patients: strategic problems concerning long-term decisions usually at the scale of years; tactical problems for medium-term decisions at the scale of months; operational problems for short-term decisions at the scale of days; and real-time problems concerning dynamic decisions. Table 2.3 shows the main characteristics found in applications for the literature review of non-emergency transport of P&P. The following parts describe the research problems found at each decision level.

2.2.1 Strategic problems

Strategic problems concern long-term decisions such as depot location, fleet size or opening hours for MSI. Healthcare institutions locate depots next to facilities in order to reduce response times that are critical in urgent cases. Depots are then located at the same time as health care facilities. Facility location problems in a non-emergency context are very often solved independently of the routing design (Ahmadi-Javid et al., 2017). However, some applications consider simultaneously location and routing to improve the service access in applications for developing countries (Doerner et al., 2007; ur Rahman and Smith, 2000).

The vehicle fleet for P&P is highly varied due to a large number of possibilities of combining wheelchair, seats or stretchers. Hence, the choice of vehicles and the fleet dimensioning is a real concern for transporters. Fu and Ishkhanov (2004) proposed a method to estimate the fleet size and mix for a given application. A case study for the weekday paratransit service in Canada is presented considering low-demand of 460 trips and high-demand of 682 trips scenarios. This study confirms the intuition that the larger the vehicles, the higher the average vehicle productivity and the smaller the required fleet size. Moreover, under high-demand applications is more advantageous to use larger vehicles. Clearly, there exists a critical point beyond which additional capacity would not improve the system performance.

In the paratransit context, the opening hours of MSIs impact the possibilities of transport pooling among multiple MSIs. Defining opening hours is out of reach of transporters. However, it is to the interest of both MSIs and transporters to propose alternative hours that can bring new opportunities to facilitate transport pooling and reduce costs. To the best of our knowledge, this transport problem is not yet treated in the research literature. Demand characteristics, regular or occasional, are not explicitly addressed in the strategic issues. However, for occasional transport, we tend to prefer flexible vehicles to fixed capacity buses to adapt to changing demands.

2.2.2 Tactical problems

Tactical decisions consider medium-term choices about the routes design and service-oriented considerations such as time-consistency and driver-consistency for regular P&P. In scientific literature, the routing problem that better adapts to the characteristics of paratransit is the Dial-A-Ride Problem (DARP). Unlike classic routing problems, DARPs include constraints and objectives related to the services offered to passengers (Parragh et al., 2012). Tactical decisions mostly concern multi-period transport problems (i.e multiple days for a week).

The concepts of time-consistency and driver-consistency have been presented in the previous section. The relationship between time-consistency and transport costs is studied in a vehicle routing problem (VRP) context which is a DARP without time constraints and with a single delivery point (MSI). The majority of contributions on this field are developed for the transport of goods (Groër et al., 2009; Kovacs et al., 2015a). However, Feillet et al. (2014) proposed a new

consistency measure adapted to the transport of people. This measure is defined for each user and is based on time slots of fixed width (i.e. 10 minutes). Thus, service times within 10 minutes to each other will be considered in the same class. The user time-consistency is therefore equal to the number of different time-classes. This problem is called the time-consistent VRP. To the best of our knowledge, the time-consistency aspect has not been investigated in a DARP context.

The impact of driver-consistency on transport costs is examined in Braekers et al. (2014) for the transport of PWDs in Belgium. The simultaneous relationship between driver-consistency, time-consistency, and transport costs has only been studied for the transport of goods in Kovacs et al. (2015b). There is a research gap about the relationship between time-consistency and transportation costs in the transport of patients. This thesis aims to fulfill this gap with a new problem denoted the time-consistent DARP. The problem is detailed in Chapter 4. Tactical problems are almost exclusive to regular transport systems. Occasional transport has considerations at the operation or strategic level.

2.2.3 Operational problems

Operational problems concern the design of vehicle routes for a single time period (i.e one day). In the occasional transport of patients, new requests are generally collected during the day, so that the next day's itineraries are scheduled at the end of the day. The majority of applications in the literature concern operational routing problems considering different transport features.

Toth and Vigo (1997) solved a DARP with heterogeneous users (seat and wheelchair) for the paratransit service in the city of Bologna. This problem was later extended to include heterogeneous vehicles (Borndörfer et al., 1999; Parragh, 2011) and multiple depots (Braekers et al., 2014; Detti et al., 2017). Borndörfer et al. (1999) and Braekers et al. (2014) studied a paratransit service in Germany and Belgium respectively while Parragh (2011) a broader transport service including the transport of patients in stretchers for the Austrian Red Cross. The DARP with heterogeneous users and vehicles is called Heterogeneous DARP (H-DARP).

It is quite common in paratransit vehicles to have different inner configurations of seats and wheelchair spaces to be able to adapt to the different needs of demand. Configurations may be done at depots (Rekiek et al., 2006; Qu and Bard, 2013) or en-route (**Tellez et al., 2018b**). En-route reconfigurations are only possible for easy layout modifications like (un)folding seats without significant changeover time. Higher flexibility of vehicles favors greater utilization of the vehicles and transportation pooling. The DARP with en-route reconfigurations is a contribution of this thesis and will be presented in Chapter 3 (Tellez et al., 2018b).

The DARP assumes a full driver availability. However, in practice, drivers are subject to regulatory rules. The DARP with driver-related constraints is studied in Xiang et al. (2006a); Parragh et al. (2012); Zhang et al. (2015). Zhang et al. (2015) studied the patient transportation service of the Hong Kong Hospital Authority providing ambulance transport services for disabled

and elderly patients from one healthcare location to another. This problem is modeled as a multi-trip dial-a-ride problem which requires designing several routes for each ambulance in order to maximize driver availability. Some patients present preferences about drivers or incompatibilities with other patients for reasons of health or behavior (Detti et al., 2017).

Some healthcare applications consider service quality not as a constraint but as an objective in itself. Paquette et al. (2012) proposed some quality measures for evaluating the service of dial-a-ride systems. And, for a case study in the paratransit transport of a Canadian city the authors combine objectives related to costs, waiting time and user ride time using dynamic weights for generating non-dominant solutions (Paquette et al., 2013). Lehu  d   et al. (2014) proposed a multi-criteria approach for combining cost and patient-oriented objectives within a preference model based on the choquet integral. For an application in the transport of patients in Belgium, Molenbruch et al. (2017) proposed a bi-objective modeling to evaluate the trade-off between the operational efficiency and the service quality measured by total ride time.

The Aeromedical Airlift Wing of the United States Air Force is responsible for transporting personnel in need of specialized medical treatment to and from military hospitals. Non-emergency patients may be transferred before the actual consultation date for increasing vehicle utilization and reduce transport costs. This problem is modeled as a route based pickup and delivery problem with the objective of minimizing the number of nights at the hospital before consultation (Ruland, 1999). Other applications consider service oriented criteria into a single weighted objective function using suitable coefficients (Fu, 1999, 2002; Ruland, 1999; Aldaihani and Dessouky, 2003; Detti et al., 2017).

Door-to-door transport systems may be combined with other application-related problems. Aldaihani and Dessouky (2003) studied the integration of the transport on-demand with fixed bus lines in the transport for disabled or elderly individuals at Lancaster, California. The aim of the integration is to reduce the vehicle miles of the transport on-demand which is more expensive. Dikas and Minis (2014), on the contrary, considered the case in which buses operating in public transport routes may diverge from its nominal path to pick-up passengers and drop them off at their destination. This service is particularly relevant for people with limited mobility that can not easily reach bus stops. Two objectives in a lexicographic order were considered: first the maximization of requests served; and then the total ride time in order to select the solutions with lower ride times for the rest of passengers. Coppi et al. (2013) modeled the healthcare timetable of non-urgent patients for a hospital in the city of Tuscany (Italy), in such a way that the transportation costs for patients going to hospitals is minimized.

Most routing problems applications assume deterministic and time-unvarying travel times. However, traffic congestion in an urban environment can significantly modify these times Fu and Teply (1999). Fu (1999) studied the disabled adult transportation system in the city of Edmonton (Canada) under dynamic and stochastic travel times. Travel times may differ at different hours of the day (dynamic) and can be altered by external events like traffic (stochastic).

His study showed that there are several negatives consequences from ignoring the dynamic and stochastic variation of travel times such as a high percentage of trips that may not be served during their desired time windows and a decline in vehicle productivity due to the dynamic aspect (Fu, 1999).

The previous problems conceive solutions for occasional transport services. Regular transport problems are modeled with additional patient requirements into a DARP system. Fleischmann et al. (2009) for example consider a desired pick-up and drop-off service times beside the DARP time windows. The objective of this problem is to minimize the transport cost as well as deviations to the desired service time. Some patients may be absented one day for different reasons. In those cases, transport planners should adjust vehicle routes so that the inconvenience to the rest of passengers is minimized. To our knowledge, this problem has not been studied in the literature. A closer problem in the transport of goods seeks to minimize planning discrepancies with the original schedule in relation to service times. This problem is known as the vehicle re-scheduling problem (Spliet et al., 2014).

2.2.4 Real-time problems

Real-time problems treat decisions mainly about accepting new transport requests, canceling or reprogramming existing ones. Very often, new transport requests arrive with the desired service for the same day time, while vehicles are already in service with operational schedules. The dispatcher usually has to give an immediate answer providing a pickup and drop-off times if the transport is feasible. Routing problems with on-line decisions are usually known as dynamic problems. In these problems, real-time technological support is needed to inform vehicles en-route about schedule modifications (Pillac et al., 2013).

Most of the paratransit applications combine a dynamic version of the DARP (Dyn-DARP) dealing with on-line requests with classic planning dealing with off-line requests from the day before (Madsen et al., 1995; Fu, 1999; Xiang et al., 2006b; Schilde et al., 2011). However, patient transfers between hospitals are mainly dynamic as on-line requests account for around 90% of cases (Hanne et al., 2009; Beaudry et al., 2010). Dyn-DARPs are based on fast heuristics to quickly find a suitable schedule of the incoming request. Mo et al. (2018) studied the option of providing a 20% discount for passengers who accept a 15-min earlier pick-up time or a later drop-off time. An application in Hong Kong paratransit service shows that this policy increases around 15% the number of people served.

Unlike operational problems, Dyn-DARPs include, in almost any application, objectives combining cost and quality criteria. Indeed, dispatchers look for affordable solutions satisfying patient's preferences. Madsen et al. (1995) proposed an insertion heuristic that permits a flexible way of weighting different goals for the paratransit service in the city of Copenhagen. Depending on the given weights, the solution reflects the user's preferences. In a German hospital, Hanne et al. (2009) considered a weighted objective function measuring criteria in a single time scale:

patient inconvenience measured as the deviations to the latest desired time (total lateness); fleet operating cost (total earliness); fleet utilization (driving time); and patient satisfaction (total ride time of patients).

Scheduling in paratransit services might be influenced by traveling time fluctuation or external events such as traffic congestion. This problem, known as the dynamic dial-a-ride problem under time-dependent and stochastic events, is studied in Fu and Teply (1999) for the paratransit service in the city of Edmonton (Canada). Then, it was extended to include other stochastic events, such as new requests, absences, vehicle breakdowns, and cancellations (Schilde et al., 2011). Results showed that dynamic problems under stochastic events provide more reliable and realistic transport schedules to customers.

Transport requests can sometimes be anticipated with the help of historic data. A patient regularly transported from home to a hospital for dialysis, consumes a predictable amount of time before going back home. An application of the daily operations in the Austrian Red Cross studied the impact of adding prior information about inbound requests. With a certain probability, each outbound request causes a corresponding inbound request on the same day. The result is a Dynamic and Stochastic DARP (Dyn-S-DARP) showing that models with stochastic information can outperform myopic dynamic models under certain circumstances (Schilde et al., 2011). Real-time problems are mainly occasional. Similarly, to operational problems, regular applications would imply on-line planning modifications that minimize deviations to the original schedule.

Problem	Works	Type	Objectives		Demand				P&P	Vehicles			Time				
			Cost	Service quality	Regular	Occasional	Static	Dynamic	Deterministic	Stochastic	Heterogeneous	Fleet size	Heterogeneous	Configurable	Reconfigurable	Max ride time	Time windows
Strategic																	
Depot location	Doerner et al. (2007)	Facility location problems	X	X	x	x	X	X									
Fleet dimensioning	Fu and Ishkhanov (2004)	Fleet size and mix DARP	X			X	X	X	X	X	X	X		X	X		
Tactical																	
Consistency	Braekers and Kovacs (2016)	DARP driver consistent	X	X	X		X	X	X		X			X	X		
	Feillet et al. (2014)	VRP time consistent	X	X	X		X	X									
	Chapter 4	DARP time-consistent	X	X	X		X	X	X	X				X	X		
Operational																	
Vehicle routing	Aldaihani and Dessouky (2003)	PDP with bus lines	X	X		X	X	X						X	X		
	Borndörfer et al. (1999)	H-DARP	X			X	X	-	X	X		X		X	X		
	Braekers et al. (2014)	H-DARP	X			X	X		X	X		X		X	X		
	Coppi et al. (2013)	H-DARP with scheduling	X			X	X		X	X		X		X	X		
	Deti et al. (2017)	H-DARP with compatibility c.	X	X		X	X		X		X			X	X		
	Dikas and Minis (2014)	Flexible bus lines		X		X	X		X					X	X		X
	Fleischmann et al. (2009)	DARP patient preferences	X	X	X	X	X							X	X		
	Fu (1999)	H-DARP time-dependent-S	X	X		X	X		X		X			X	X		X
	Fu (2002)	DARP time-dependent-S	X	X		X	X							X	X		X
	Lehuédé et al. (2014)	DARP Multi-criteria	X	X		X	X		X					X	X		
	Molenbruch et al. (2017)	DARP Bi-objective	X	X		X	X		X					X	X		
	Paquette et al. (2013)	H-DARP Multi-criteria	X	X		X	X		X		X			X	X		
	Parragh (2011)	H-DARP	X			X	X		X	X		X		X	X		
	Parragh et al. (2012)	H-DARP with driver c.	X			X	X		X	X		X		X	X		
	Qu and Bard (2015, 2013)	H-DARP-C	X			X	X		X	X		X	X		X	X	
	Rekiek et al. (2006)	H-DARP-C	X			X	X		X	X		X	X		X	X	
	Ruland (1999)	PDP patient preferences	x	X		X	X		X	X		X			X		
	Toth and Vigo (1997);	DARP with user types	X			X	X		X						X	X	
	Xiang et al. (2006a)	DARP with driver c.	x			X	X		X			X			X	X	
	Zhang et al. (2015)	VRP multi-trips	X			X	X		X		X				X	X	
Chapter 3 (Tellez et al., 2018b)	FSM-DARP-RC	X			X	X		X	X	X	X	X	X	X	X		
Real time																	
Online vehicle routing	Beaudry et al. (2010)	Dyn-DARP	X	X		X	-	X	X	X		X	X		X		
	Fu and Tepy (1999)	Dyn-DARP time-dependent-S	X	X		X	X	X	X	X		X		X	X		X
	Hanne et al. (2009)	Dyn-DARP	X	X		X	-	X	X	X		X	X		X		
	Madsen et al. (1995)	Dyn-DARP	X	X		X	X		X	X		X		X	X		
	Mo et al. (2018)	DARP with discounts	X			X	X		X					X	X		
	Schilde et al. (2011)	Dyn-S-DARP	X	X		X	X	X	X	X				X	X		
	Xiang et al. (2008)	Dyn-DARP time-dependent-S	X			X		X		X		X		X	X		X

Table 2.3: Scientific contributions in the domain of non-emergency transport of patients & passengers.

DARP: Dial-a-Ride Problem, PDP: Pickup-And-Delivery Problem, VRP: Vehicle Routing Problem, H-DARP: Heterogeneous DARP, Dyn-DARP: Dynamic DARP, Dyn-S-DARP: Dynamic and Stochastic DARP, c: constraints, FSM: Fleet size and mix.

2.3 Conclusions

In healthcare transport, there are different operating modes according to the nature of the demand. The two main transport types are emergency and non-emergency transport. A lot of attention has been focused on the evolution of emergency medical systems. However, a large percentage of patient transport is non-urgent. Non-emergency transport concerns a wide range of applications in-home care, transport for PWD and intra-hospital transport.

Depending on the frequency of demands of recurrent P&P, non-emergency transport can be occasional or regular. Both transport types differ to each other in their objectives and characteristics. This thesis examines long-term transport planning for people with disabilities that, according to this classification, falls into the category of *non-emergency regular transport*.

Contributions in the non-emergency transport of patients can impact the system's performance at a strategic, tactical, operational or real-time level. There are numerous contributions in this area, especially at the operational and real-time levels.

Reviewed applications at the strategic level are very few and do not distinguish regular and occasional P&P. Regular transport problems are mostly studied at the tactic level. However, the needs of time-consistency in a DARP context, which is a characteristic of regular transport systems, has not been studied in the literature. The definition and solution of this problem is a contribution of this thesis that will be studied in Chapter 4.

At the operational level, en-route reconfiguration of vehicles is a novelty contribution of this thesis which will be presented in Chapter 3.

Finally, we point out that although the decisions about the opening hours of MSIs are out of reach of transporters. It is to the interest of both MSIs and transporters to negotiate these hours that can lead to a possible transport pooling among MSIs. Transport pooling is not a common practice in the medico-social sector. However, there is an economic intensive to share resources. The impact of transport pooling in the daily transport of PWD will be studied in Chapter 5.

Half-day transport problem: The fleet size and mix dial-a-ride problem with reconfigurable vehicle capacity

This chapter introduces a fleet size and mix dial-a-ride problem with multiple passenger types and a heterogeneous fleet of reconfigurable vehicles. In this new variant of the dial-a-ride problem, en-route modifications of the vehicle's inner configuration are allowed. The main consequence is that the vehicle capacity is defined by a set of configurations and the choice of vehicle configuration is associated with binary decision variables. The problem is modeled as a mixed-integer program derived from the model of the heterogeneous dial-a-ride problem. Vehicle reconfiguration is a lever to efficiently reduce transportation costs, but the number of passengers and vehicle fleet setting make this problem intractable for exact solution methods. A large neighborhood search metaheuristic combined with a set covering component with a reactive mechanism to automatically adjust its parameters is therefore proposed. The resulting framework is evaluated against benchmarks from the literature, used for similar routing problems. It is also applied to a real case, in the context of the transportation of disabled children from their home to medical centers in the city of Lyon, France.

Keywords: Dial-A-Ride Problem, Fleet Size and Mix Problem, reconfigurable vehicles, heterogeneous fleet, large neighborhood search, set-covering, feasibility check.

This chapter have been published in the journal chapter:

Tellez, O., Vercaene, S., Lehuédé, F., Péton, O., Monteiro, T., 2018b. The fleet size and mix dial-a-ride problem with reconfigurable vehicle capacity. *Transportation Research Part C*:

3.1 Introduction

Solving the Dial-A-Ride Problem (DARP) consists in designing vehicle routes in order to fulfill transportation requests scattered throughout a geographic area. The objective is to minimize costs while satisfying transportation requests. The DARP is a special case of the Pickup and Delivery Problem (PDP) in which constraints and objectives related to the quality of services offered to passengers are integrated (the reader may refer to Parragh et al. (2008) for a recent survey on PDP). The main applications concern door-to-door transportation of people, particularly elderly or disabled people.

In this chapter, we propose a generalization of the heterogeneous PDP with configurable vehicle capacity introduced by Qu and Bard (2013). This problem includes both heterogeneous fleets of vehicles and heterogeneous users. We consider several types of users who do not occupy the same space in the vehicle (e.g. passengers using seats or wheelchairs). The vehicle capacity depends on the chosen *configuration*. A *configuration* is characterized by a multidimensional capacity vector indicating the maximum number of users of each type allowed in the vehicle. Figure 3.1 presents three configurations of a vehicle with two types of users (passengers using seats or wheelchairs). The configuration has a capacity of 1 wheelchair and 7 seats, including 5 folding seats (the driver's seat is not available for passengers). In the second configuration, one folding seat has been lifted and replaced by a wheelchair space. This configuration has a capacity of 2 wheelchairs and 6 seats. Adding one more wheelchair requires two additional seats to be folded. This is represented by the third configuration, which has a capacity of 3 wheelchairs and 4 seats.

Vehicle reconfiguration is as simple as folding or unfolding a foldable seat. This operation is performed manually by the driver, when the vehicle is parked. It takes only a few seconds, while service times take several minutes (including the time necessary to park the vehicle, install the car ramp, help passengers get in/off the vehicle, etc.). Although reconfigurable vehicles lead to higher vehicle investments, neither extra time nor cost is associated with the operation of en-route reconfiguration.

Note that there is no linear correspondence between the number of seats and wheelchairs. Moreover, the total number of passengers can be either 7 or 8, depending on the chosen configuration.

Qu and Bard (2013) integrate the choice of the initial vehicle configurations as a decision variable in a PDP dedicated to the transport of people with reduced mobility. In such services, it is actually possible to switch en-route from one configuration to another to serve additional transportation requests. As a result, we propose a generalization to the problem presented by Qu and Bard (2013), by assuming the possibility of en-route reconfiguration. This extension

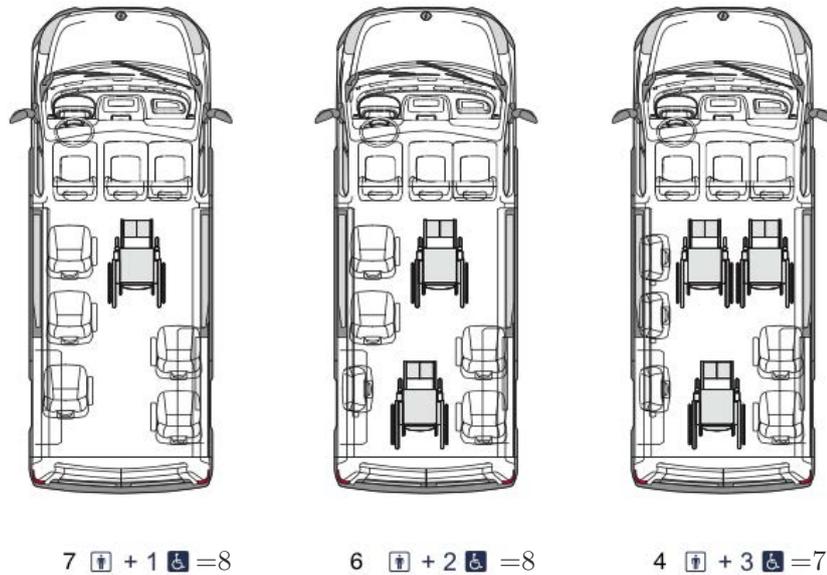


Figure 3.1: Vehicle with three configurations (<http://handynamic.fr>)

is called the DARP with Re-Configurable vehicle capacity (DARP-RC). The present study considers an arbitrarily large fleet of heterogeneous vehicles that has to be set: the problem addressed here falls within the category of fleet size and mix problems. We therefore call it the Fleet Size and Mix DARP with reconfigurable vehicle capacity (FSM-DARP-RC).

This work is motivated by the daily transport of people with disabilities in the city of Lyon operated by the GIHP Company¹. Every day, GIHP transports around 500 children from and to Medico-Social Institutions (MSI). GIHP works for 60 MSIs with about 180 adapted vehicles. From Mondays to Fridays, users are picked up at home in the morning and driven to their MSI. In the afternoon, they are transported back home. For MSIs, transportation is often considered the second-biggest expense after wages. Optimizing transport is therefore a priority. In this chapter, without loss of generality, we present the case of morning trips. Every year, the company defines the fleet mix as well as the vehicles' itineraries. In addition, routing planners re-assess decisions daily, if required. All decisions are currently made without vehicle routing software. In order to design routes, route planners have to use simplifying assumptions, e.g. considering each MSI separately and ignoring the possibility of en-route vehicle reconfiguration. These assumptions lead to an over-constrained problem and consequently, to sub-optimal solutions. One aim of this chapter is to solve the problem of having to stop at several MSIs on the same route. Using reconfigurable vehicles furthermore increases the possibility to pool routes between MSIs. Another aim of this chapter is to assess the impact of en-route reconfiguration.

The chapter is organized as follows: Section 3.2 reviews related problems. Section 3.3 presents the problem settings and the mathematical formulation for the FSM-DARP-RC. Section 3.4

¹www.gihp-sa.com

is dedicated to the solution method. In particular, we detail the Large Neighborhood Search (LNS) operators and the set covering component. The capacity check and the minimization of the number of reconfigurations are detailed in Section 3.5. We report extensive computational results and management insights in Section 3.6. Finally, Section 3.7 gives the conclusions and prospects.

3.2 Literature review

In this section, we first review some related work on DARP. We then present papers regarding the design and the routing of a fleet of heterogeneous vehicles with a special focus on DARP applications. Finally, we discuss how configuration and reconfiguration have been considered in the Vehicle Routing Problem (VRP) literature.

3.2.1 The dial-a-ride problem

The DARP is related to the optimization of a *multi-occupancy, door-to-door transport service for people* (Doerner and Salazar-González, 2014). Most commonly, this problem consists in designing the routes of a homogeneous fleet of vehicles to satisfy a set of transportation requests. The objective is to minimize the sum of route costs, satisfying constraints on vehicle capacity, time windows at origins and destinations of requests, and limits on the ride time of each passenger. Due to the combinatorial nature of the problem, exact methods (see for instance Ropke et al. (2007); Ropke and Cordeau (2009); Braekers et al. (2014)) are still limited in their ability to solve large size instances. Accordingly, several metaheuristics have been proposed to solve the DARP and its variants. In particular, Cordeau and Laporte (2003) developed a Tabu Search algorithm and proposed a set of instances that are still used to benchmark new algorithms. The most successful heuristics that have been applied to these instances in the last ten years are the Variable Neighborhood Search (VNS) of Parragh et al. (2010), the Hybrid LNS (HLNS) of Parragh and Schmid (2013), the Adaptive LNS (ALNS) of Masson et al. (2014), the deterministic annealing algorithm of Braekers et al. (2014), the evolutionary local search of Chassaing et al. (2016) and the ALNS of Gschwind and Drexel (2016).

One of the most challenging parts of DARP heuristics is the scheduling algorithm. It determines the feasibility of routes with respect to temporal constraints and, in some case, parts of the objective function such as the routes' duration. Several efficient algorithms have thus been designed to satisfy the most common DARP constraints with the best possible complexity (see e.g. Hunsaker and Savelsbergh (2002); Tang et al. (2010); Firat and Woeginger (2011); Gschwind, Timo (2015)). The latest contribution in this area is Gschwind and Drexel (2016), which proposes an incremental algorithm to evaluate, in constant time, the feasibility of request insertions in routes. The *minimization of route duration* is considered in Cordeau and Laporte

(2003). This algorithm has been used and extended in many papers such as Parragh et al. (2010) and Braekers et al. (2014).

3.2.2 Routing a heterogeneous fleet of vehicles

In the FSM-DARP-RC, a heterogeneous fleet of vehicles has to be constituted to transport several types of users. In the VRP context, the routing of a heterogeneous fleet of vehicles was recently surveyed in Koç et al. (2016). Problems of this class fall into two main variants: (i) the Fleet Size and Mix VRP (FSM-VRP) introduced by Golden et al. (1984), which considers an unlimited fleet of various vehicle types and costs for using vehicles. Solving the problem amounts to determining the fleet of vehicles and designing routes simultaneously; (ii) the Heterogeneous VRP (HVRP) introduced by Taillard (1999), which considers a given fleet of various types of vehicles. In such problems, vehicle types can induce differences both in constraints (e.g. capacity) or objectives (e.g. distance related or fixed related costs).

Regarding people transportation problems, the combined transportation of seated and wheelchair passengers by a heterogeneous fleet of vehicles has been studied in Toth and Vigo (1997). This problem falls into the class of Heterogeneous Dial-A-Ride-Problems (HDARP) introduced later by Parragh (2011). In the latter chapter, the authors propose a branch-and-cut algorithm which optimally solves instances with up to 40 requests and a hybrid method which combines branch-and-price and VNS. The HDARP was also thoroughly investigated in Braekers et al. (2014). The authors propose a branch-and-cut algorithm and a deterministic annealing metaheuristic to solve a multi-depot version of this problem. To our knowledge, variants of the FSM-DARP have not yet been covered by the literature.

3.2.3 Configurable vehicle capacity

Finally, the FSM-DARP-RC extends the notion of vehicle configuration, which implies having multiple capacity options for some vehicles. To our knowledge, this notion was first introduced by Qu and Bard (2013) who define the heterogeneous PDP with configurable vehicle capacity. The problem could actually be called a DARP since it integrates considerations related to passenger transportation such as ride time minimization. These authors consider a limited and heterogeneous fleet of configurable vehicles that serve transportation requests coming from a heterogeneous set of passengers. In addition to routing decisions, one configuration must be chosen for each vehicle. Hence, this problem differs from the FSM-DARP-RC since a vehicle is assumed to keep the same configurations for its entire route. The authors propose a multi-start ALNS. The assignment of configurations for a given set of routes is done in a feasibility check either by a heuristic or by solving a general assignment problem when the heuristic fails. A branch-and-price algorithm for this problem was also proposed in Qu and Bard (2015) for instances with up to 30 requests.

Choosing a vehicle configuration can be compared to determining compartment sizes in the flexible multi-compartment VRP described in Derigs et al. (2011). In this chapter, compartment sizes are determined together with a set of routes for vehicles with flexible compartments. The goal is to deliver various types of goods, given that some goods cannot be transported in the same compartment. Compartments can be *continuously flexible* as presented in Koch et al. (2016); Derigs et al. (2011), or *discretely flexible* if compartment options are defined beforehand as in Henke et al. (2015). The difference with our problem is twofold: first, in the FSM-DARP-RC, the total capacity changes from one configuration to another, in a non-linear way, whereas in multi-compartment problems, the sum of compartment sizes is assumed to remain constant. Second, we consider a PDP. Hence, a vehicle's load can increase and decrease along its route and en-route reconfiguration may be advantageous. In a VRP, a single configuration is needed since all goods are delivered at once. To our knowledge, the PDP with flexible compartment sizes has not been studied.

3.2.4 Contributions with respect to the literature

Regarding the literature, the main contributions of this chapter are the following. First, we introduce the FSM-DARP-RC and propose a mathematical model for this problem. Second, we propose a matheuristic framework that combines a LNS with a reactive set covering component to solve this problem. This matheuristic is proven to be competitive on benchmark instances for the DARP and the heterogeneous PDP with configurable vehicle capacity. This matheuristic integrates innovative components related to the use of reconfigurable vehicles: (i) the *vehicle selection procedure*, which efficiently determines the minimum cost of a reconfigurable vehicle that is feasible for a given route, and (ii) the *reconfigurable vehicle capacity test*, which determines if a given vehicle can perform a given route together with its minimum number of reconfigurations. We also show that, when it is combined with a set covering component in a matheuristic, the ALNS framework introduced in Pisinger and Ropke (2007) can be simplified. Finally, we present some managerial insights based on our experiments on the real instances provided by the GIHP. These instances will be made available to the community.

3.3 Problem settings and mathematical model

In this section, we start by providing an example of en-route reconfiguration. Then in Section 3.3.2 the FSM-DARP-RC problem is formally described, and is modeled in Section 3.3.3.

3.3.1 Example

The example represented in Figure 3.2 illustrates the potential benefits of vehicle reconfiguration. We consider a single vehicle which starts and terminates its routes at depot D . This vehicle has two possible configurations, denoted as c_1 and c_2 respectively. Configuration c_1 consists of

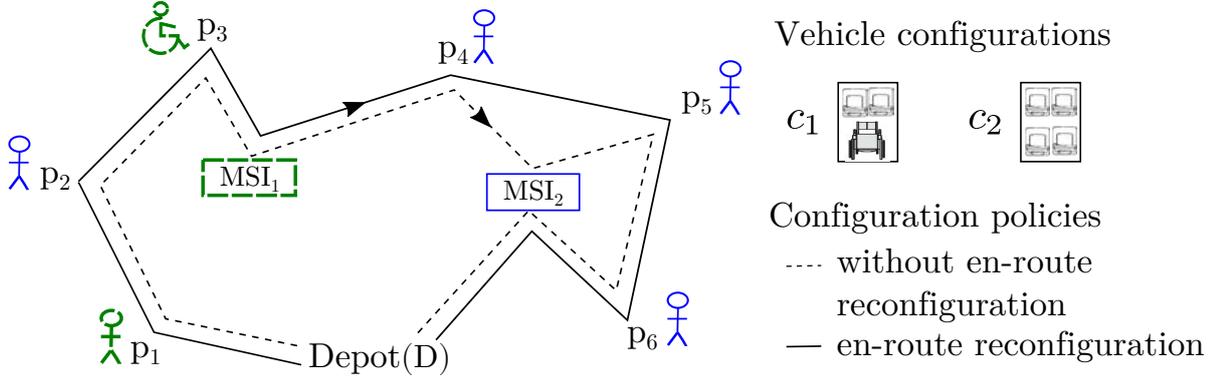


Figure 3.2: Comparison of routes with and without vehicle capacity reconfiguration

2 seats and 1 wheelchair space. Configuration c_2 consists of 4 seats only. In this example, six users are transported to two MSIs. Users 1 and 3 are picked up at p_1 and p_3 respectively, and get off at MSI_1 . Users 2, 4, 5 and 6 are picked up at p_2 , p_4 , p_5 and p_6 respectively, and get off at MSI_2 . User 3 uses a wheelchair.

Without an en-route reconfiguration policy, the only solution is to leave the depot D with configuration c_1 , which is the only one that includes a wheelchair space. The optimal solution for this policy is represented with dashed lines in Figure 3.2.

However, with en-route reconfiguration policy, the vehicle can be configured with configuration c_2 at the depot D and reconfigured with configuration c_1 after visiting MSI_1 . This solution is represented with solid lines in Figure 3.2. This strategy reduces the length of the route by the value of the detour $\Delta_{p_4, MSI_2} + \Delta_{MSI_2, p_5} - \Delta_{p_4, p_5}$, where $\Delta_{i,j}$ represents the value of the shortest path from i to j .

3.3.2 Problem settings

The FSM-DARP-RC is modeled on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$. The set \mathcal{V} of nodes contains the set \mathcal{O}^+ of starting depots, the set \mathcal{O}^- of arrival depots, the set \mathcal{P} of pickup locations and the set \mathcal{D} of delivery locations. We consider a large (potentially infinite) heterogeneous fleet of vehicles. We denote by \mathcal{K} the set of vehicle types. Without loss of generality, we consider in this section that there is a single vehicle of each type, so that \mathcal{K} also represents the set of vehicles. Each vehicle $k \in \mathcal{K}$ has a starting depot $o_k^+ \in \mathcal{O}^+$ and an ending depot $o_k^- \in \mathcal{O}^-$. In practice, o_k^+ and o_k^- often correspond to the same physical location. The set \mathcal{A} contains three categories of arcs: Arcs (o_k^+, i) where $k \in \mathcal{K}, i \in \mathcal{P}$. Arcs (i, j) where $i, j \in \mathcal{P} \cup \mathcal{D}, i \neq j$. Arcs (i, o_k^-) where $k \in \mathcal{K}, i \in \mathcal{D}$. Every arc (i, j) represents the fastest path from nodes i to j . It is associated with a travel time t_{ij} and a distance Δ_{ij} .

The set of transport requests is denoted by \mathcal{R} . These requests concern passengers who may require various types of spaces in vehicles (e.g. seat, wheelchair). By extension, the type of

space required by a passenger in a vehicle is called a user type and the set of user types is denoted by \mathcal{U} . Each request $r \in \mathcal{R}$ is characterized by a pickup node $p_r \in \mathcal{P}$, a delivery node $d_r \in \mathcal{D}$, a maximum ride time \bar{T}_r and a quantity $q_{r,u}$ of users of each type to be transported from p_r to d_r . Note that even though each request is modeled by pair of pickup and delivery vertices, in practice, different nodes can refer to the same geographical location. We define the following notations to represent the load variation at the pickup node and the delivery node of each request respectively: $\phi_{p_r,u} = q_{r,u}$ and $\phi_{d_r,u} = -q_{r,u}$. This modeling implies that one request can include many users at a time if they share the same transport requirements (pickup and delivery locations, time windows and maximum ride time). For instance, if a passenger with a wheelchair and another passenger traveling on a regular seat have the same transport requirements, they can be considered in the same request r with load $q_{r,0} = 1$ and $q_{r,1} = 1$ thus sharing the same vehicle.

Each node $i \in \mathcal{V}$ is associated with a service duration s_i and a time window $[a_i, b_i]$. At depots $o \in \mathcal{O}^+ \cup \mathcal{O}^-$, we assume that load variations are null for all user types, vehicles are empty and the service time s_o is null. Each vehicle $k \in \mathcal{K}$ is associated with a fixed cost f^k , a cost per kilometer γ^k and a set $\mathcal{C}^k = \{1, \dots, \bar{c}^k\}$ of possible configuration indexes. The set of configurations is defined by a set of capacity vectors $\mathcal{Q}^k = \{\mathbf{Q}^{k1}, \dots, \mathbf{Q}^{k\bar{c}^k}\}$ with $\mathbf{Q}^{kc} = (Q_1^{kc}, \dots, Q_{|\mathcal{U}|}^{kc})$, where Q_u^{kc} represents the maximal number of users of type $u \in \mathcal{U}$ that can be carried by vehicle $k \in \mathcal{K}$ in configuration $c \in \mathcal{C}^k$.

Example 1. Let us consider the vehicle $k \in \mathcal{K}$ represented in Figure 3.1. We have $\mathcal{C}_k = \{1, 2, 3\}$ and $\mathcal{Q}^k = \{\mathbf{Q}^{k1}, \mathbf{Q}^{k2}, \mathbf{Q}^{k3}\} = \{(4, 3), (6, 2), (7, 1)\}$.

The sequence of nodes visited by a vehicle forms a *route*. A route is characterized by a vehicle $k \in \mathcal{K}$ (and its associated depots $o_k^+ \in \mathcal{O}^+$ and $o_k^- \in \mathcal{O}^-$) and a sequence of pickup and delivery nodes. The maximal allowed route duration is denoted by \bar{T} . The GIHP's experience shows that reconfigurable vehicles allow fast and easy reconfigurations. So, in this work, for a given vehicle $k \in \mathcal{K}$, switching from a configuration $c \in \mathcal{C}^k$ to another configuration $c' \in \mathcal{C}^k$ can be done at no cost and within negligible time. However, to limit the inconvenience for the driver, related to this task, we introduce the parameter \bar{R} , which denotes the maximum number of reconfigurations allowed within a route. For each driving hour, there is a cost α related to the driver wages.

The FSM-DARP-RC consists in selecting a set of vehicles, designing their pickup and delivery routes, determining the time of service at each node and selecting the configuration used on each arc of the solution. The problem solutions must satisfy the vehicle capacity for all selected configurations, the maximum number of reconfigurations for each route, the time window constraint for each node and the maximum ride time constraint of each request.

As a generalization of the vehicle routing problem, the FSM-DARP-RC is NP-hard.

3.3.3 Mathematical model

Let us introduce the following variables:

- x_{ij}^{kc} is a binary variable which is equal to 1 if vehicle $k \in \mathcal{K}$ uses arc $(i, j) \in \mathcal{A}$ with configuration $c \in \mathcal{C}_k$, and 0 otherwise,
- z_i^k is a binary variable which is equal to 1 if vehicle $k \in \mathcal{K}$ is reconfigured at node $i \in \mathcal{P} \cup \mathcal{D}$, and 0 otherwise.
- $l_{i,u}^k$ is an integer variable representing the number of users of type $u \in \mathcal{U}$ in vehicle $k \in \mathcal{K}$ after visiting node $i \in \mathcal{V}$,
- w_i^k is a continuous variable representing the time of service of vehicle $k \in \mathcal{K}$ at node $i \in \mathcal{V}$.

The FSM-DARP-RC can then be formulated with the following mixed integer program:

$$\min \sum_{k \in \mathcal{K}} f^k \sum_{i \in \mathcal{P}} \sum_{c \in \mathcal{C}^k} x_{o_k^+ i}^{kc} + \alpha \sum_{k \in \mathcal{K}} (w_{o_k^-}^k - w_{o_k^+}^k) + \sum_{k \in \mathcal{K}} \sum_{c \in \mathcal{C}^k} \sum_{(i,j) \in \mathcal{A}} \gamma^k \Delta_{ij} x_{ij}^{kc} \quad (3.1)$$

s.t.

$$\sum_{c \in \mathcal{C}^k} \sum_{(p_r, j) \in \mathcal{A}} x_{p_r j}^{kc} - \sum_{c \in \mathcal{C}^k} \sum_{(j, d_r) \in \mathcal{A}} x_{j d_r}^{kc} = 0 \quad \forall r \in \mathcal{R}, k \in \mathcal{K} \quad (3.2)$$

$$\sum_{c \in \mathcal{C}^k} \sum_{k \in \mathcal{K}} \sum_{(j, p_r) \in \mathcal{A}} x_{j p_r}^{kc} = 1 \quad \forall r \in \mathcal{R} \quad (3.3)$$

$$\sum_{c \in \mathcal{C}^k} \sum_{(j, i) \in \mathcal{A}} x_{j i}^{kc} - \sum_{c \in \mathcal{C}^k} \sum_{(i, j) \in \mathcal{A}} x_{i j}^{kc} = 0 \quad \forall i \in \mathcal{P} \cup \mathcal{D}, k \in \mathcal{K} \quad (3.4)$$

$$\sum_{c \in \mathcal{C}^k} \sum_{i \in \mathcal{P}} x_{o_k^+ i}^{kc} - \sum_{c \in \mathcal{C}^k} \sum_{i \in \mathcal{D}} x_{i o_k^-}^{kc} = 0 \quad \forall k \in \mathcal{K} \quad (3.5)$$

$$w_j^k \geq w_i^k + t_{ij} + s_i - M_{ij} (1 - \sum_{c \in \mathcal{C}^k} x_{ij}^{kc}) \quad \forall (i, j) \in \mathcal{A}, k \in \mathcal{K} \quad (3.6)$$

$$a_i \leq w_i^k \leq b_i \quad \forall i \in \mathcal{V}, k \in \mathcal{K} \quad (3.7)$$

$$w_{p_r}^k + s_{p_r} + t_{p_r d_r} \leq w_{d_r}^k \quad \forall r \in \mathcal{R}, k \in \mathcal{K} \quad (3.8)$$

$$l_{j,u}^k \geq l_{i,u}^k + \phi_{ju} - \bar{Q}_u^k (1 - \sum_{c \in \mathcal{C}^k} x_{ij}^{kc}) \quad \forall (i, j) \in \mathcal{A}, u \in \mathcal{U}, k \in \mathcal{K} \quad (3.9)$$

$$l_{i,u}^k \leq \sum_{c \in \mathcal{C}^k} \sum_{(i,j) \in \mathcal{A}} Q_u^{kc} x_{ij}^{kc} \quad \forall i \in \mathcal{P}, u \in \mathcal{U}, k \in \mathcal{K} \quad (3.10)$$

$$\sum_{c' \in \mathcal{C}^k / \{c\}} \sum_{(j,i) \in \mathcal{A}} x_{j i}^{k c'} + \sum_{(i,j) \in \mathcal{A}} x_{i j}^{kc} \leq 1 + z_i^k \quad \forall i \in \mathcal{P} \cup \mathcal{D}, k \in \mathcal{K}, c \in \mathcal{C}^k \quad (3.11)$$

$$\sum_{i \in \mathcal{P} \cup \mathcal{D}} z_i^k \leq \bar{R} \quad \forall k \in \mathcal{K} \quad (3.12)$$

$$w_{d_r}^k - w_{p_r}^k - s_{p_r} \leq \bar{T}_r \quad \forall r \in \mathcal{R}, k \in \mathcal{K} \quad (3.13)$$

$$w_{o_k^-}^k - w_{o_k^+}^k \leq \bar{T} \quad \forall k \in \mathcal{K} \quad (3.14)$$

$$x_{ij}^{kc} \in \{0, 1\} \quad \forall (i, j) \in \mathcal{A}, k \in \mathcal{K} \quad (3.15)$$

$$l_{i,u}^k \in \mathbb{Z}^+ \quad \forall i \in \mathcal{V}, u \in \mathcal{U}, k \in \mathcal{K} \quad (3.16)$$

$$w_i^k \in \mathbb{R}^+ \quad \forall i \in \mathcal{V}, k \in \mathcal{K} \quad (3.17)$$

$$z_i^k \in \{0, 1\} \quad \forall i \in \mathcal{P} \cup \mathcal{D}, k \in \mathcal{K}. \quad (3.18)$$

The objective function (3.1) to be minimized is the total transportation cost, including fixed costs associated with each selected vehicle, time-related costs proportional to the total route duration (which may include some waiting time), and distance-related costs proportional to the total distance traveled. Constraints (3.2) and (3.3) ensure that every transportation request is satisfied and carried out by a single vehicle. Constraints (3.4) are flow conservation constraints at pickup and delivery nodes. Constraints (3.5) state that any vehicle $k \in \mathcal{K}$ leaving the node o_k^+ must end-up at the node o_k^- .

Constraints (3.6) set arrival time variables. If arc $(i, j) \in \mathcal{A}$ is used by a vehicle k , then the arrival time at node j is greater than the arrival time at node i plus the service duration in i and the transportation time from i to j . The big-M can be estimated as $M_{ij} = b_i + s_i + t_{ij}$. Constraints (3.7) set time windows for variables w_i^k . Constraints (3.8) assure that for each request $r \in \mathcal{R}$ in each route the pickup node is visited before its corresponding delivery node.

Constraints (3.9) propagate the load along the route in each type of user each time the arc $(i, j) \in \mathcal{A}$ is crossed by a vehicle k . The constraint linearization requires to define a big-M value denoted as $\bar{Q}_u^k = \max_{c \in \mathcal{C}^k} \{Q_u^{kc}\}$, the maximum capacity of type $u \in \mathcal{U}$ among all configurations.

Constraints (3.10) correspond to the vehicle capacity for each type of user. Note that, as the quantity of users cannot increase during deliveries, those constraints can be defined for pickup nodes only.

Constraints (3.11) and (3.12) concern the reconfiguration of vehicles. In constraints (3.11), the variable z_i^k is set at value 1 when the vehicle $k \in \mathcal{K}$ is reconfigured at node i . Constraints (3.12) limit the number of reconfigurations to be less than \bar{R} for each vehicle. Constraints (3.13) limit the total ride time (including waiting and service times) of request $r \in \mathcal{R}$ to be less than \bar{T}_r . Constraints (3.14) states that the duration of each route should be less than \bar{T} . Finally, the last constraints define the decision variables.

3.4 Solution method

In this section, we describe the solution method that we propose to solve the FSM-DARP-RC. It relies on a combination of two components: a Large Neighborhood Search (LNS) and the solution of a Set Covering Problem (SCP) using a layer with a Reactive adjustment of SCP parameters (denoted RSCP). The framework is denoted LNS-RSCP. This section is structured as follows: Section 3.4.1 presents the general framework of LNS-RSCP. LNS removal and repair operators are detailed in Section 3.4.2 and the RSCP is presented in Section 3.4.3.

3.4.1 Matheuristic framework (LNS-RSCP)

LNS was first proposed by Shaw (1998) in a constraint programming context and introduced under the name *ruin and recreate* in Schrimpf et al. (2000). In LNS, the current solution is improved by following an iterative process of destroying it (i.e. removing a part of it) and repairing it. This process is repeated until a stopping criterion is reached. In our case, the stopping criterion is either a maximum number of iterations or a maximum computational time.

The potential of LNS for solving a large variety of vehicle routing problems was revealed by Ropke and Pisinger who proposed an Adaptive version of LNS, known as ALNS, consisting of multiple search operators adaptively selected according to their past performance (see e.g. Ropke and Pisinger (2006) and Pisinger and Ropke (2007)). LNS was successfully applied to many variants of vehicle routing problems. In particular, it remains one of the best-known approaches to solve PDPs.

The RSCP component acts as a long-term memory. Given that routes that appear together in optimal or near-optimal solutions can be generated at distinct LNS iterations, it aims at reassembling these routes. Let us consider two routes ω and ω' , with respective costs $\Pi(\omega)$ and $\Pi(\omega')$. Route ω is said to *dominate* ω' if both routes visit the same nodes and $\Pi(\omega) \leq \Pi(\omega')$. All non-dominated routes collected through LNS iterations are stored in a set $\Omega = \{1, \dots, |\Omega|\}$ called the *pool of routes*. The RSCP component is based on a route-based formulation of the FSM-DARP-RC built from this pool. An SCP is solved with an MILP solver at regular intervals. The number of iterations between each SCP solution is adjusted in a reactive way, as described below. The solver is given a time limit to solve this problem.

The LNS-RSCP framework is described in Algorithm 1.

The algorithm requires a set of repair and removal operators denoted as Σ^+, Σ^- respectively. The initial solution is created using a repair operator, denoted as $\sigma_{init}^+ \in \Sigma^+$, from a totally destroyed solution. At every iteration, the proportion of requests to be removed, called Φ , is randomly determined according to a continuous uniform distribution on the interval between parameters Φ^- and Φ^+ , where $0 < \Phi^- < \Phi^+ < 1$ (line 7). A removal operator $\sigma^- \in \Sigma^-$ and a repair operator $\sigma^+ \in \Sigma^+$ are selected according to a discrete uniform distribution in line 8. The operator σ^- removes $\max\{1, \lfloor \Phi \cdot |\mathcal{R}| \rfloor\}$ requests from the *current solution* s (line 9).

These requests are placed in a set of unsatisfied requests called Request Bank (\mathcal{B}). From the request bank \mathcal{B} , they are reinserted into the partially destroyed solution using the repair operator σ^+ (line 10). If the solution cannot be completely repaired, its total cost is increased by the value *penalty* for each unsatisfied request. Once the solution s' is repaired, it can be either accepted or rejected as the next *current solution* using an acceptance criterion (line 11). We use the *record-to-record* acceptance criterion proposed by Dueck (1993): if $objective(s') \leq (1 + \chi) \cdot objective(s^*)$, then s' is accepted as the new *current solution* where χ is a small positive value. We refer to Santini et al. (2016) for an extensive comparison of the acceptance criteria that support this choice. Similarly, the *best solution* s^* is updated every time a new *temporary solution* s'

Algorithm 1: The LNS-RSCP framework

Input: Σ^- : set of destroy operators, Σ^+ : set of repair operators,
 η : initial number of iterations between two solutions of a SCP,
 t_{limit} : initial time limit for SCP.
Output: Best solution found s^* .

- 1 Build initial solution s using repair operator $\sigma_{init}^+ \in \Sigma^+$
- 2 Pool of routes: $\Omega \leftarrow \emptyset$
- 3 Request bank: $\mathcal{B} \leftarrow \emptyset$
- 4 $iter \leftarrow 0$
- 5 **while** *Termination criterion not met* **do**

/* LNS component */

 - 6 $s' \leftarrow s$
 - 7 **Destroy quantity:** Randomly select a proportion $\Phi \sim Uniform(\Phi^-, \Phi^+)$ of requests to remove
 - 8 **Operator selection:** Randomly select a destroy operator $\sigma^- \in \Sigma^-$ and a repair operator $\sigma^+ \in \Sigma^+$
 - 9 **Destroy:** Apply σ^- to remove $\max\{1, \lfloor \Phi \cdot |\mathcal{R}| \rfloor\}$ requests from s' and place them into the request bank \mathcal{B}
 - 10 **Repair:** Apply σ^+ to reinsert requests from \mathcal{B} into s'
 - 11 **if** *Acceptance criterion is met* **then**
 - 12 $s \leftarrow s'$
 - 13 **if** *Cost of s' is better than the cost of s^** **then**
 - 14 $s^* \leftarrow s'$

/* RSCP component */

 - 15 Update Ω with the non-dominated routes of s'
 - 16 **if** *iter modulo $\eta = 0$ (every η iterations)* **then**
 - 17 $s \leftarrow \text{Solve SCP}(\Omega, s^*, t_{limit})$
 - 18 **if** *the SCP is not solved to proven optimality* **then**
 - 19 Reinitialize Ω with the routes of s^*
 - 20 *Every two consecutive failures to optimally solve the SCP: update parameter*
 - 20 η
 - 21 Remove duplicate requests in s
 - 22 $s^* \leftarrow s$
 - 23 $iter \leftarrow iter + 1$
- 24 **return** s^*

cheaper than s^* is found (lines 13-14). Finally, lines 15 to 22 describe the RSCP component of the algorithm. Every η iteration (line 16), SCP is solved with a MILP solver to find the best combination of routes generated during the previous iterations (line 15). A parameter t_{limit} is used to limit the time allocated to the solver. When the SCP cannot be solved to optimality within the time limit, the pool of routes Ω is cleared and reinitialized with the routes of the best-known solution s^* (line 19). The reactive layer of the RSCP readjusts parameter η every time the solver fails to optimally solve the SCP twice consecutively (line 20).

3.4.2 LNS operators

The sets of removal and repair operators are major components of the LNS method. Hence, many LNS operators have been developed. Pisinger and Ropke (2007) proposed a list of 7 destroy and 2 repair operators to solve a large variety of vehicle routing problems efficiently. In practice, LNS operators should enable both intensification and diversification, but the exact list of implemented operators varies substantially from one implementation to another. Several papers (see e.g., Christiaens and Vanden Berghe (2016)) have shown that implementing only a few good operators is enough to get very good solutions. Additional operators may slightly improve the quality of the solutions. Following this principle, our implementation of LNS uses only a few operators in order to keep the framework as simple as possible, without sacrificing the solution quality.

Removal operators

These operators determine the set of requests to be removed from the current solution according to a given criterion. The following operators have been used:

- *Random removal*: This operator removes a proportion Φ of randomly selected requests from the current solution.
- *Historical node-pair removal*: This operator consists in removing a proportion Φ of requests from the current solution that were “better placed” in previous solutions. The LNS algorithm calculates a score for each arc $(i, j) \in \mathcal{A}$. The initial score of each arc is set to infinity. It is then updated at each iteration with the value of the best solution found so far that includes arc (i, j) . The removal heuristic calculates the cost of a request $r \in \mathcal{R}$ from pickup node i to delivery node j by summing the score of the arcs that are incident to i and j in the current solution. It then removes requests according to a probabilistic rule used in Ropke and Pisinger (2006): Consider the list \mathcal{R} of requests sorted by non-increasing costs. The operator iteratively removes the request at position $\xi^p \times |\mathcal{R}|$, where $0 \leq \xi < 1$ is a random number and $p \in \mathbb{Z}$ is a deterministic parameter. This probabilistic choice gives higher probability to the requests with the highest cost to be removed.

We also implemented the following operators, that were not kept in the final version of the LNS: *worst removal*, which removes the request responsible for the longest detour in the solution; *distance-related removal* which removes the nearest nodes, based on a distance-related indicator; and *time-related removal*, which removes requests that are similar from a time point of view. For a full description of these operators, we refer to Pisinger and Ropke (2007).

Repair operators

Removed requests are stored in the request bank \mathcal{B} . Repair operators are intended to take them out of the request bank and to re-insert them into the current partial solution. We implemented the two most common repair operators: *best insertion* and *k-regret insertion*.

- *Best insertion*: At each iteration, the best insertion position is calculated for each request $r \in \mathcal{B}$, in each route of the current solution. The request with the minimal insertion cost is then inserted at its best position. This process is repeated until the request bank is empty or no more feasible insertion exists.
- *K-regret*: This operator generalizes Best insertion with more looking-ahead information. At each iteration, the best insertion position of each request $r \in \mathcal{B}$ is calculated for each route of the current solution. Let Δf_r^j designate the insertion cost of a request $r \in \mathcal{R}$ in its j^{th} best route at its best position. Δf_r^1 denotes the insertion cost (min additional cost) of inserting request $r \in \mathcal{R}$ in its best route, Δf_r^2 is the insertion cost for the same request in its 2nd best route, etc. The request r^* selected for insertion at its best position is: $r^* = \arg \max_{r \in \mathcal{B}} \left(\sum_{j=2}^k \Delta f_r^j - \Delta f_r^1 \right)$. The heuristic stops when the request bank is empty or when no more requests can be inserted. In this chapter, we consider *K-regret* heuristics with values of *K* between 2 and 4 and regard them as 3 independent operators.

3.4.3 Set covering problem (SCP)

The use of heuristic approaches for generating routes to solve a set covering or set partitioning formulations of the VRP was first proposed by Foster and Ryan (1976): several routes, called *petals* are first generated and a set partitioning problem is then solved to build a VRP solution. This approach was then extended to multiple applications. In particular, it has been combined with local search methods in Rochat and Taillard (1995) and Subramanian et al. (2013) for the VRP, and has been combined with LNS-based heuristics in Parragh and Schmid (2013) and Gschwind and Drexel (2016) for the DARP. Note that, whereas Gschwind and Drexel (2016) solve a single SCP at the very end of their algorithm, the matheuristic of Parragh and Schmid (2013) uses reduced costs from the SCP solution to guide a variable neighborhood search that can possibly generate new routes. In the survey of Archetti and Speranza (2014) for VRP matheuristics, this approach is classified under the name of *column generation based heuristics*.

The name is naturally chosen because of the similarity with set partitioning formulations in branch-and-price algorithms. More precisely, this method falls into the class of *restricted master heuristics* according to the same survey.

The proposed approach is based on the framework of Grangier et al. (2017) for a VRP with cross-docking. According to this framework, a solver is called every η iterations to solve the SCP on a pool Ω containing all non-dominated routes found by LNS in the previous iterations. When the number of routes in Ω makes the SCP intractable for the solver within a given time limit t_{limit} , Ω is cleared and reinitialized with the routes of the best solution found so far.

In the following sections, we detail the mathematical formulation of the SCP and the pool management process. We also develop a reactive version denoted RSCP where parameter η is automatically adjusted from one SCP solving to another.

Formulation of the SCP

Note that $\Omega = \{1, \dots, |\Omega|\}$ is the set of routes collected through LNS iterations, and $\pi_\omega \in \mathbb{R}^+$ denotes the cost of route $\omega \in \Omega$. Let $\rho_{r\omega}$ be a parameter equal to 1 if request $r \in \mathcal{R}$ is served by route $\omega \in \Omega$, and 0 otherwise.

The SCP aims at building a new solution by selecting a subset of independent routes in Ω . The SCP model uses binary variables y_ω , that are set at value 1 if route $\omega \in \Omega$ is part of the solution and 0 otherwise. The SCP can be defined by the following integer linear program.

$$\min \sum_{\omega \in \Omega} \pi_\omega y_\omega \quad (3.19)$$

s.t.

$$\sum_{\omega \in \Omega} \rho_{r\omega} y_\omega \geq 1 \quad \forall r \in \mathcal{R} \quad (3.20)$$

$$y_\omega \in \{0, 1\} \quad \forall \omega \in \Omega. \quad (3.21)$$

SCP solving

As introduced before, the SCP is solved by a MILP solver with a run time limited to t_{limit} . To improve the solver performance and guarantee that a feasible solution is returned, the solver is initialized with the current routes of s^* using a so-called *warm-start* function.

Given that constraints (3.20) of the SCP model allow request duplication, a request may appear in more than one route in the optimal solution of the SCP. In this case the request is left only in the route that yields the best cost (line 21, Algorithm 1).

Management of the pool of routes Ω

At each iteration of the LNS-SCP, the non-dominated routes in the current solution are added to the pool of routes Ω (line 15, Algorithm 1), possibly dominating some routes in the pool. Hence, the size of the pool generally increases after each iteration. Eventually, the pool becomes so large that the MILP solver cannot optimally solve the SCP within the time t_{limit} . When this happens, the pool Ω is cleared and reinitialized with the routes of s^* . This strategy used in Grangier et al. (2017) allows for the route pool to be maintained at a reasonable size. Of course, this strategy heavily depends on the solving time left to the MILP solver.

Reactive adjustment of SCP parameters (RSCP)

The combination of the run time t_{limit} and the frequency η determine the overall computational effort spent in solving the SCP. Thus, one issue in tuning the SCP component is to define a common policy to set the values of parameters t_{limit} and η in all instances.

In our numerical experiments we observe that, for a given value of η , the run time necessary to optimally solve the SCP is considerably longer on larger instances. Moreover, this time is longer at the beginning of the LNS execution than at the end. This has already been observed by Subramanian et al. (2013) in a VRP context. The authors developed a reactive strategy to limit the number of routes in the SCP. Their approach requires setting 5 parameters. Moreover, a threshold mechanism eliminates bad solutions even if they contain good routes. Thus, we propose a reactive layer RSCP that allows a simpler and automatic adjustment of parameter η , reducing its value geometrically such that $\eta \leftarrow \eta/\psi$ (Algorithm 1, line 20). This mechanism uses a single real parameter $\psi > 1$ and assures that every dominant route will be considered at least once in a SCP solving. Unlike the re-initialization of Ω which is done when the solver cannot solve the SCP optimally, the automatic adjustment of η is performed only when the solver cannot solve the SCP optimally twice consecutively.

3.5 Evaluation of the insertion of requests

In each iteration of the framework LNS-SCP (Algorithm 1), the repair operator checks the potential insertions of all unplanned requests in all routes at all positions. This represents a large number of insertion attempts. In practice, for every unplanned request, only the best insertion is performed. Repair operators therefore evaluate the feasibility and the performance of a very large number of unnecessary insertions, which has a major impact on the performance of the SCP-LNS algorithm. This section describes the core algorithms involved in the insertion of unplanned requests.

Let ω denote a feasible route with N nodes. Without loss of generality, let us denote by $1, \dots, N$ the set of nodes visited by this route, where nodes 1 and N are the initial and the final depot of this route. Inserting request $r \in \mathcal{R}$ consists in inserting pickup node $p_r \in \mathcal{P}$

between nodes $i \in \omega$ and $i + 1 \in \omega$ and inserting delivery node $d_r \in \mathcal{D}$ between nodes $j \in \omega$ and $j + 1 \in \omega$. We call ω' the route resulting from this request insertion, as represented in Figure 3.3

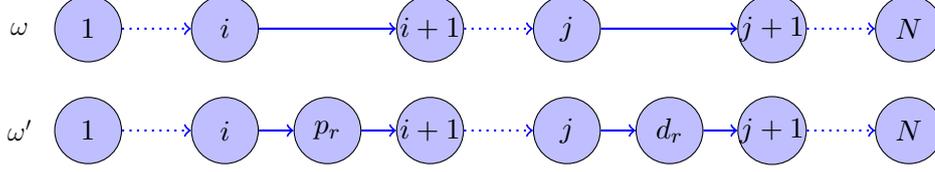


Figure 3.3: Insertion of request r into route ω

The feasibility evaluation of ω' has two parts: The *Capacity evaluation* checks that at least one vehicle type can serve all requests in ω' with at most \bar{R} reconfigurations. The *Schedule evaluation* computes the minimal route duration and checks if there is a departure time for route ω' that complies with all users time windows and maximum ride times.

Let us denote by $time(\omega')$ the minimal duration of route ω' , $\mathcal{K}(\omega')$ the set of vehicle types compatible with route ω' , $k(\omega') \in \mathcal{K}(\omega')$ the cheapest vehicle type compatible with route ω' , and $dist(\omega')$ the total length of route ω' . Note that $dist(\omega')$ can be computed in constant time with an incremental approach:

$$dist(\omega') = dist(\omega) + \begin{cases} \Delta_{i,p_r} + \Delta_{p_r,i+1} - \Delta_{i,i+1} + \Delta_{j,d_r} + \Delta_{d_r,j+1} - \Delta_{j,j+1} & \text{if } i < j, \\ \Delta_{i,p_r} + \Delta_{p_r,d_r} + \Delta_{d_r,i+1} - \Delta_{i,i+1} & \text{if } i = j. \end{cases}$$

Remember that α and γ^k represent the unitary costs related to the route duration and length respectively. Then, if route ω' is feasible regarding the *Capacity evaluation* and the *Schedule evaluation*, the minimal cost $\Pi(\omega')$ of route ω' is determined by

$$\begin{aligned} \Pi(\omega') &= \alpha \, time(\omega') + \min_{k \in \mathcal{K}(\omega')} (f^k + \gamma^k dist(\omega')), \\ &= \alpha \, time(\omega') + f^{k(\omega')} + \gamma^{k(\omega')} dist(\omega'). \end{aligned} \quad (3.22)$$

Note that in our application, all vehicles travel at the same speed. Hence, the time feasibility and the duration of a route can be evaluated without knowing the vehicle that travels that route.

Algorithm 2 shows how the *Schedule evaluation* and the *Capacity evaluation* algorithms are organized. Since these algorithms are mutually independent, they can be executed in any order. Although this order may impact computational times, there is no general rule to define which one should be placed first because the performance always depends on the considered data. *Schedule evaluation* (line 11) is performed first because our instances are mostly constrained by time windows and ride times. Accordingly, the *Capacity evaluation* (line 4) is run only on the resulting feasible routes. The algorithm returns the cost of an insertion, and the value -1 if the insertion is not possible.

Sections 3.5.1 and 3.5.2 detail the *Schedule evaluation* and the *Capacity evaluation* respectively.

Algorithm 2: Evaluation of a route ω'

Input: Routes ω and ω'
Output: Total insertion cost (-1 if infeasible)

```

1  $time(\omega') \leftarrow ScheduleEvaluation(\omega')$           /* Algorithm 11 */ ;
2 if  $time(\omega') > -1$  then
3    $k(\omega') \leftarrow CapacityEvaluation(\omega, \omega')$       /* Algorithm 4 */ ;
4   if  $k(\omega') > -1$  then
5      $\Pi(\omega') = \alpha time(\omega') + f^{k(\omega')} + \gamma^{k(\omega')} dist(\omega')$ ;
6     return  $\Pi(\omega')$ 
7 return -1

```

3.5.1 Schedule evaluation

The scheduling of a route first consists in determining if the user time windows and maximum ride time are respected on the route. Second, it determines a minimal duration schedule. The *Schedule evaluation* is the most time-consuming operation in the algorithm due to the large number of evaluations performed each time a request insertion is evaluated.

The route scheduling for our problem is performed by Algorithm 11. It is based on Tang et al. (2010) as presented in Gschwind, Timo (2015). We add forward time slack calculations from Cordeau and Laporte (2003) in order to determine the minimum route duration. Note that some simple classical and necessary conditions are checked before running this algorithm. These are well summarized in Braekers et al. (2014).

Let us consider a route $\omega = \{1, \dots, N\}$. For each node $i \in \omega$, Algorithm 11 determines the beginning of service w_i in such a way that route duration is minimized. This algorithm has three phases. In the first phase, the earliest possible schedule is computed for every visited node (lines 5 to 9). At the same time, we calculate the forward time slack of node 1, denoted by F_1 , and the total waiting time H on the route. The second phase minimizes the route duration by shifting the first node w_1 by F_1 units. Consequently, the beginning of service in all other nodes $i = 2, \dots, N$ is updated (lines 11 to 13). Once the route duration is minimized, the maximum route duration constraint is verified in line 14.

The third phase checks ride-time constraints for every request (lines 16 to 27), starting from the end of the route. For any pickup node i , we denote as $r \in \mathcal{R}$ the request to which it belongs. Consequently, d_r is the corresponding delivery node and i is equal to p_r . The ride-time associated with this request is then $w_{d_r} - w_{p_r} + s_r$. If the value $(w_{d_r} - w_{p_r} + s_r) - \bar{T}_i$ is positive, then the max ride-time constraint associated with pickup p_r is violated. In this case, the beginning of service w_{p_r} is shifted by the value of this violation (line 25). This may lead to infeasibility on time windows, which is checked in lines 26 to 29. Finally, since the beginning of service w_{p_r} may have been changed, the ride-time constraint for request r is checked again in line 30. If no infeasibility is detected, the minimal route duration $w_N - w_1$ is returned (line 29).

Algorithm 3: Schedule evaluation

Input: Route $\omega = \{1, \dots, N\}$.
Output: Minimal duration of route ω (-1 if infeasible)

```

1  $w_1 \leftarrow a_1$  /* Beginning of the service */
2  $H \leftarrow 0$  /* Total waiting time on the route */
3  $F_1 \leftarrow b_1 - w_1$  /* Forward time slack at node 1 */
4
  /* Phase 1: set up nodes at the earliest start */
5 for  $i = 2, \dots, N$  do
6    $w_i \leftarrow \max\{a_i; w_{i-1} + s_{i-1} + t_{i-1,i}\}$ 
7   if  $w_i > b_i$  then return -1
8    $H \leftarrow H + \max\{0; a_i - (w_{i-1} + t_{i-1,i} + s_{i-1})\}$ 
9    $F_1 \leftarrow \min\{F_1; H + \max\{0; b_i - w_i\}\}$ 
10
  /* Phase 2: optimize route duration */
11  $w_1 \leftarrow w_1 + F_1$ 
12 for  $i = 2, \dots, N$  do
13    $w_i \leftarrow \max\{w_{i-1} + s_{i-1} + t_{i-1,i}; a_i\}$ 
  /* Check route duration constraint */
14 if  $(w_N - w_1) > \bar{T}$  then return -1
15
  /* Phase 3: check ride time constraints */
16 for  $i = N - 2, \dots, 1$  do
17   if  $i \in \mathcal{P}$  then
18      $r \leftarrow$  request of pickup  $i$  /* Implies  $i = p_r$  */
19      $\delta \leftarrow (w_{d_r} - w_{p_r} + s_r) - \bar{T}_r$ 
20     if  $(\delta > 0)$  then
21        $w_{p_r} \leftarrow w_{p_r} + \delta$ 
22       if  $w_{p_r} > b_{p_r}$  then return -1
23       for  $j = p_r + 1, \dots, N$  do
24          $w_j \leftarrow \max\{a_j; w_{j-1} + s_{j-1} + t_{j-1,k}\}$ 
25         if  $w_j > b_j$  then return -1
26       if  $\bar{T}_r - (w_{d_r} - w_{p_r} + s_r) < 0$  then
27         return -1
28
29 return  $w_N - w_1$ 

```

3.5.2 Capacity evaluation

The objective of this section is to complete the evaluation of a route by determining the cheapest vehicle type that satisfies capacity constraints as well as the constraint on the maximal number of reconfigurations. Section 3.5.2.1 presents the procedure that assigns a vehicle to a given route. Section 3.5.2.2 details the subroutine that checks if a given reconfigurable vehicle is feasible for a given route.

3.5.2.1 Vehicle type selection

We consider the resulting route ω' after the insertion of a request $r \in \mathcal{R}$ in route ω . A first observation is that the set of vehicle types that can perform route ω' is included in the set of vehicles that can perform route ω : $\mathcal{K}(\omega') \subset \mathcal{K}(\omega)$. Thus, the *Capacity evaluation* of ω' can be limited to the vehicle types in $\mathcal{K}(\omega)$.

A second observation is that it is sufficient to check the capacity only over a smaller subset of key nodes of the route. This leads to the following definitions:

Definition 1. We call the *load profile* of a route the list of vectors representing its load at every node of this route. The load profile for route ω' is $\{\mathbf{l}_1, \dots, \mathbf{l}_N\}$, where $\mathbf{l}_i = \{l_{i,u}, i \in 1, \dots, N; u \in \mathcal{U}\}$, and $l_{i,u}$ represents the load of user type $u \in \mathcal{U}$ at node $i \in \omega'$.

Load $l_{i,u}$ results from the accumulation of load variations of the route from node 1 to node i , i.e. $l_{i,u} = \sum_{j=1}^i \phi_{j,u}$.

Definition 2. The set of pickup nodes of a route that are immediately followed by a delivery node is called the *kernel* of this route. The restriction of the load profile $\{\mathbf{l}_1, \dots, \mathbf{l}_N\}$ to the nodes of its kernel is denoted by \mathcal{L} .

Theorem 1. A vehicle type $k \in \mathcal{K}$ can be assigned to a route ω' if, and only if

- each load $\mathbf{l} \in \mathcal{L}$ is compatible with at least one configuration of vehicle type k ,
- the number of reconfigurations required to carry every load $\mathbf{l} \in \mathcal{L}$ is less than or equal to the maximum number of allowed reconfigurations \bar{R} .

Proof. If i is the delivery node of some request $r \in \mathcal{R}$. Then, $l_{i,u} \leq l_{i-1,u}$ for all user types $u \in \mathcal{U}$. Thus, if the capacity is satisfied at node $i - 1$, it is also satisfied at node i . If i is a pickup node followed by another pickup node, then $l_{i,u} \leq l_{i+1,u}$ and the capacity at i is satisfied if it is satisfied at $i + 1$.

Regarding the number of reconfigurations, it is trivial to note that reducing the number of loads to test can not increase the number of reconfigurations. However, we shall prove that reducing the number of test to the set of pickups followed by a delivery node do not reduce the number of reconfigurations. First, note that it is not necessary to reconfigure before a

diminution of load (i.e. delivery). Reconfigurations can therefore be done just before a pickup node. Second, let consider a feasible route for a given vehicle. The route has two consecutive pickups and the first one requires a new configuration among a set of feasible configurations S_1 . Because the load is increasing, the second pickup requires a configuration from the subset S_2 such that $S_2 \subset S_1$. Finally, given that the configuration for the first pickup should be chosen from S_2 to minimize the number of reconfigurations, we can therefore limit the test to the second pickup and by extension to every pickup followed by a delivery. ■

By definition, for each node i in a route, there exist a node j in the route kernel that has a greater load (ie such that $\forall u \in \mathcal{U} : l_{i,u} \leq l_{j,u}$). Let k be a vehicle type. If there exists a configuration $c \in \mathcal{C}^k$ such that $\forall u \in \mathcal{U} : Q_u^{kc} \geq l_{j,u}$, then the load \mathbf{l}_i is also compatible with configuration c . As a result, checking the vehicle capacity in every node of the route kernel is equivalent to check the vehicle feasibility of the entire route. Then, in order to save computation time, the capacity evaluation of route ω' can be restricted to its kernel.

Figure 3.4 presents the successive loads of a vehicle on a route with two types of users (seats and wheelchairs). In Figure 3.4, pickup operations are illustrated by arcs pointing to the right or to the top and deliveries are represented by arcs pointing to the left or to the bottom. Although the route has 7 nodes, a sufficient *Capacity evaluation* of the route requires only a capacity check in the nodes of the kernel, this is nodes 3 and 5.

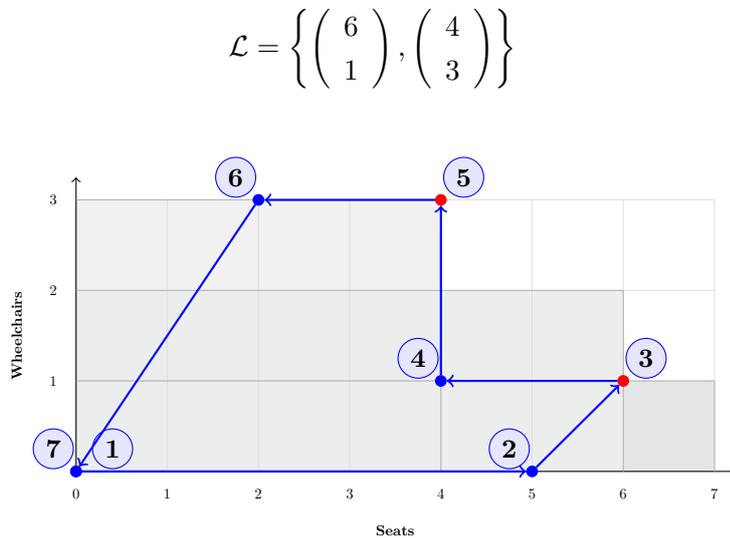


Figure 3.4: Graphical representation of a route load

Algorithm 4 describes the *Capacity evaluation* procedure. It first evaluates the fixed and distance-related costs for all vehicle types (lines 2–3). The route load profile is computed in lines 5–10. In line 10, a vertex are appended to the route kernel if it is a pickup, immediately followed by a delivery vertex. Checking if a vehicle type k is feasible for \mathcal{L} is done on line 13.

Considering that k can be reconfigured and that several types of users are considered, this evaluation is non-trivial and is detailed in Section 3.5.2.2.

Algorithm 4: Capacity evaluation

Input: Routes ω and ω' , list $\mathcal{K}(\omega)$ of vehicle types compatible with ω , distance $dist(\omega')$
Output: Cheapest vehicle type for the route ω' (-1 if infeasible)

```

1
  /* Estimate partial route costs */
2 forall  $k \in \mathcal{K}(\omega)$  do
3    $\lambda^k = f^k + dist(\omega')v^k$ 
4
  /* Compute the load profile for the kernel of the route */
5  $\mathcal{L} = \emptyset$ 
6  $\mathbf{l}_1 = \mathbf{0}$ 
7 for  $i \leftarrow 2, \dots, N$  do
8    $\mathbf{l}_i = \mathbf{l}_{i-1} + (\phi_{j,u})_{u \in \mathcal{U}}$ 
9   if  $i$  belongs to the kernel of  $\omega'$  then
10     $\mathcal{L} \leftarrow \mathcal{L} \cup \{\mathbf{l}_i\}$ 
11
  /* Find a feasible vehicle type */
12 for  $k \leftarrow \mathcal{K}(\omega)$  in non-decreasing order of costs  $\lambda^k$  do
13   if  $k$  can carry  $\mathcal{L}$  then /* Algorithm 5 */
14    return  $k$ 
15 return  $-1$ 

```

Proposition 1. *The Capacity evaluation algorithm returns $k(\omega')$, the cheapest vehicle type for the route ω' , or the value -1 if no vehicle can perform route ω' .*

Proof. From equation (3.22),

$$\Pi(\omega') = \alpha \text{time}(\omega') + \min_{k \in \mathcal{K}(\omega)} \lambda^k,$$

and

$$k(\omega') = \arg \min_{k \in \mathcal{K}(\omega)} \lambda^k.$$

Given that vehicles type $k \in \mathcal{K}(\omega)$ are tested one by one in non-decreasing order of values λ^k , as soon as ω' is feasible for a given $k \in \mathcal{K}(\omega')$, we have $k = k(\omega')$. ■

3.5.2.2 Vehicle type feasibility

In this section, we focus on solving the question: given a vehicle type $k \in \mathcal{K}$ and a load profile \mathcal{L} , can this vehicle carry this load? (see line 13, Algorithm 4)

Recall that \mathcal{Q}^k is the set of all configuration vectors for a vehicle type $k \in \mathcal{K}$, represented by the $|\mathcal{C}^k| \times |U|$ matrix:

$$\mathcal{Q}^k = \left\{ \begin{array}{c} \mathbf{Q}^{k1} \\ \dots \\ \mathbf{Q}^{k|\mathcal{C}^k|} \end{array} \right\} = \left\{ \begin{array}{ccc} Q_1^{k1} & \dots & Q_{|U|}^{k1} \\ \dots & \dots & \dots \\ Q_1^{k|\mathcal{C}^k|} & \dots & Q_{|U|}^{k|\mathcal{C}^k|} \end{array} \right\},$$

where \mathbf{Q}^{kc} represents the c^{th} configuration ($c \in \mathcal{C}^k$) of vehicle k and Q_u^{kc} the capacity of vehicle type k in configuration c for user $u \in U$.

Example 2. Let us consider the vehicle type k described in Figure 3.1. Figure 3.5 represents \mathbf{Q}^{k1} , \mathbf{Q}^{k2} , \mathbf{Q}^{k3} .

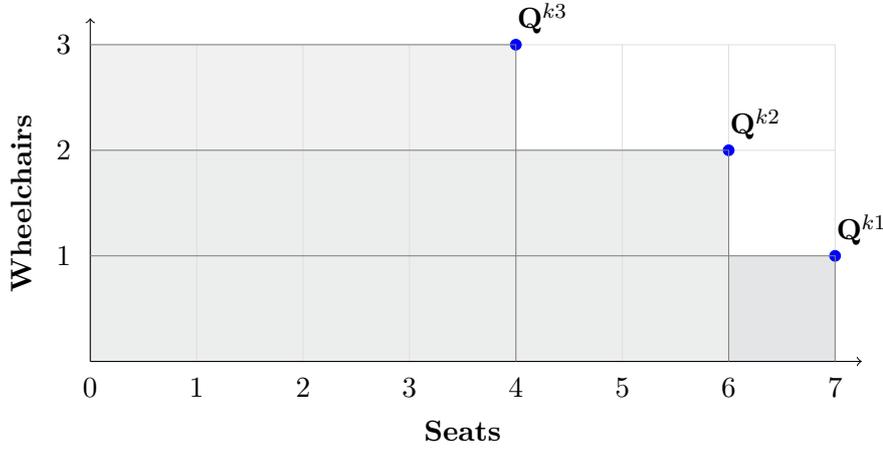


Figure 3.5: Capacity of the vehicle presented in Figure 3.1.

According to these notations, we can state that a route ω is feasible for a given vehicle type k if: (i) for each load \mathbf{l} in the route kernel \mathcal{L} , there is a capacity vector in \mathcal{Q}^k that is greater than \mathbf{l} ; and (ii) if the minimum number of reconfigurations of k to operate \mathcal{L} is less than \bar{R} .

The efficiency of these two checks is increased by computing *a priori* the list of feasible configurations for each possible load vector and for each vehicle type. The envelope set of possible load vectors is defined by

$$\mathcal{T}^k = \{0, \dots, \max_{c \in \mathcal{C}^k} Q_1^{kc}\} \times \dots \times \{0, \dots, \max_{c \in \mathcal{C}^k} Q_{|U|}^{kc}\}.$$

Mathematically, we define $S^k(\mathbf{l})$ as the function that maps the set of feasible configurations of vehicle type k for a given load \mathbf{l} .

$$S^k : \mathcal{T}^k \rightarrow \mathbb{P}(\mathcal{C}^k),$$

where $\mathbb{P}(\cdot)$ is the power-set function of set \mathcal{C}^k , i.e. the set of all subsets of \mathcal{C}^k , including the empty set and \mathcal{C}^k itself.

The function $S^k(\cdot)$ can be pre-processed and written as a matrix with $|\mathcal{U}|$ dimensions. Each vector of feasible configuration in this matrix is encoded using a *bitset*. This guarantees a limited size in memory and a $O(1)$ access to the list of configurations able to carry any element of \mathcal{T}^k .

Table 3.1 details the value of function $S^k(\cdot)$ in the case described by Example 2 and Figure 3.5. For each possible load value, this table returns the list of configurations that are compatible with this load. For example, for a load of 4 seats and 2 wheelchairs, configurations 1 and 2 are feasible. For a load of 6 seats and 3 wheelchairs, there is no feasible configuration.

		Seats							
		0	1	2	3	4	5	6	7
Wheelchairs	3	{1}	{1}	{1}	{1}	{1}	\emptyset	\emptyset	\emptyset
	2	{1, 2}	{1, 2}	{1, 2}	{1, 2}	{1, 2}	{2}	{2}	\emptyset
	1	{1, 2, 3}	{1, 2, 3}	{1, 2, 3}	{1, 2, 3}	{1, 2, 3}	{2, 3}	{2, 3}	{3}
	0	{1, 2, 3}	{1, 2, 3}	{1, 2, 3}	{1, 2, 3}	{1, 2, 3}	{2, 3}	{2, 3}	{3}

Table 3.1: Matrix representation of S^k for the vehicle presented in Figure 3.1.

Algorithm 5 presents the procedure for evaluating the feasibility of vehicle k in route ω' . The algorithm initializes the number nr of necessary reconfigurations to 0 (line 1) and the set $ListConfig$ with all admissible configurations (line 2). It then iteratively goes through the nodes of the entire route kernel \mathcal{L} . If the current node is not compatible with any vehicle configuration (line 4), vehicle k is infeasible and the value -1 is returned. Otherwise, the list $ListConfig$ is updated with the set of feasible configurations common to this node and $ListConfig$ (line 6). If the load of the current node \mathbf{l} requires a configuration not previously stored in $ListConfig$, a reconfiguration is needed. Thus, nr is increased by 1 and the $ListConfig$ is initialized with the set of feasible configurations of the current node (line 11). If nr exceeds the maximum value allowed, \bar{R} , vehicle k is infeasible and the value -1 is returned. Finally, if every load in the nodes of the kernel are compatible with vehicle k the number of reconfigurations nr is returned.

Proposition 2. *Algorithm 5 finds the minimal number of reconfigurations.*

Proof. We define the graph $G = (V, A)$ in which nodes $v_1, \dots, v_{|\mathcal{L}|}$ represent the ordered set of elements of the kernel. We define an arc between any pair of nodes which can be connected without reconfiguring the vehicle. So, if an arc (v_i, v_j) exists, then all arcs of the form $(v_{i'}, v_{j'})$, where $i \leq i' < j' \leq j$, also exist.

With all arcs weighted by 1, finding the minimum number of reconfigurations is equivalent to finding a shortest path between v_1 and $v_{|\mathcal{L}|}$.

Given a node $v_i \in V$, we define the *farthest* neighbor of v_i as the node v_j , such that the arc (v_i, v_j) exists and the arc (v_i, v_{j+1}) does not exist. Algorithm 5 starts from v_1 and iteratively looks for the farthest neighbor of the current node. Let $v_1, v_{opt(2)}, \dots, v_{|\mathcal{L}|}$ be an optimal path from v_1 to $v_{|\mathcal{L}|}$. We shall prove by contradiction that if some node v_i is the farthest neighbor of v_1 , then it belongs to an optimal path.

Algorithm 5: Feasibility of vehicle type k and number of necessary reconfigurations

Input: route ω' , with the load profile \mathcal{L} of its kernel
Input: vehicle k , with preprocessed values for $S^k(\cdot)$ and \mathcal{T}^k
Output: number nr of necessary reconfigurations

```

1  $nr \leftarrow 0$ 
2  $ListConfig \leftarrow \mathcal{C}^k$ 
3 forall  $l \in \mathcal{L}$  do
    |
    | /* capacity test */
    | if  $l \notin \mathcal{T}^k$  or  $S^k(l) = \emptyset$  then
    |   | return  $-1$ 
    |
    | /* number of reconfigurations test */
    |  $ListConfig \leftarrow ListConfig \cap S^k(l)$ 
    | if  $ListConfig = \emptyset$  then
    |   |  $nr \leftarrow nr + 1$ 
    |   | if  $nr > \bar{R}$  then
    |   |   | return  $-1$ 
    |   |   |  $ListConfig \leftarrow S^k(l)$ 
    |
12 return  $nr$ 

```

Assume that (v_1, v_i) is not in the shortest path from v_1 to $v_{|L|}$. Because $(v_1, v_j) \notin A, \forall j > i$, it is not possible to go from l_1 to any successor of v_i with a weight of 1. So, $v_{opt(2)} < i$ and $v_{opt(3)} > i$. By construction, if arc $(v_{opt(2)}, v_{opt(3)})$ exists and has weight 1, then arc $(v_i, v_{opt(3)})$ also exists and has the same weight. Thus, the cost of path $v_1 \rightarrow v_i \rightarrow v_{opt(3)}$ is 2. Hence, (v_1, v_i) is also an optimal solution. ■

3.6 Computational experiments

The matheuristic described in Section 3.4.1 was coded in C++ and run on a CPU Intel Xeon E5-1620 v3 @3.5Ghz. The SCP was solved with CPLEX 12.6 running on a single thread. The matheuristic was evaluated using real and benchmark data. The parameters shown in Table 3.2 provide the best average performance for the optimization problems solved in this chapter. After experimentation, we found that: K -regret operators with $K > 4$ did not improve the solution quality and two removal operators (historical removal and random removal) are sufficient when the SCP component is active. Note that values for parameters η , t_{limit} and ψ , can be automatically modified in the reactive version RSCP for different instance sizes (see Section 3.4.3).

Regarding the sequencing of feasibility evaluation, performing the *Schedule evaluation* (Algorithm 11) before the *Capacity evaluation* (Algorithm 2) can reduce computation time up to 50%. This reduction is mostly explained by the fact our data set (derived from real

$\chi = 5\%$	record-to-record acceptance percentage.
$penalty = 10\,000$	penalty cost for incomplete solutions.
$\Phi^- = 10\%$	minimal proportion of removed request used by removal operators.
$\Phi^+ = 45\%$	maximal proportion of removed request used by removal operators.
$p = 6$	roulette wheel parameter for the historical node-pair operator.
$\sigma_{init}^+ = 4\text{-regret}$	repair operator for building the initial solution
$\eta = 1\,000$	launch frequency of the SCP.
$t_{limit} = 3\text{ sec}$	imposed time limit for the SCP.
$\psi = 1.25$	RSCP coefficient to recompute the launch frequency of the SCP.

Table 3.2: Parameters (all defined in Section 3.4).

data) is constrained more in time than in capacity. In other words, long pickup legs are seldom feasible because of time windows and maximum ride times. In addition, sorting vehicle types in non-increasing order of costs, before testing the vehicles capacity, can reduce computation time by up to 10% (Algorithm 4).

This section is structured as follows: First, we introduce the instances used to evaluate the algorithms. In Section 3.6.2 we present the experiments that determine the choice and calibration of the main components of the proposed matheuristic. The proposed algorithms are compared to state-of-the-art algorithms on benchmarks from Cordeau and Laporte (2003) and Qu and Bard (2013), which correspond to particular cases of our problem. Finally, we provide managerial insights, mainly regarding vehicle fleet aspects.

3.6.1 Description of instances

The real data comes from the GIHP Company². An instance set of 14 instances is made from a bank of 576 requests decomposed into smaller subsets of three sizes. There are 8 small instances containing from 60 to 80 requests, 4 medium-size instances containing from 120 to 160 requests, and 2 large instances containing from 280 to 295 requests. We consider 2 different user types: users occupying seats and users with wheelchairs. Each type of user occupies dedicated spaces in vehicles. Travel times and distances are obtained from the Open Source Routing Machine³ (OSRM) by Luxen and Vetter (2011). Common characteristics for the instances are: 1) maximum ride times are defined according to direct travel time (t_{p_r,d_r}) by the formula $RT = 15 \times \lceil (t_{p_r,d_r} + 15)/15 \rceil$; 2) time windows at medico social institutions are 30 minutes wide; and 3) the service time for users using seats is 2 minutes at the pickup location and 1 minute at the delivery location. For users in a wheelchair, it is 5 minutes at pickup locations and 2 minutes at delivery locations.

The characteristics of vehicles considered in experiments are summarized in Table 3.4. There are four vehicle types, including one that is not configurable (V_0). Vehicles V_1 and V_2 have two

²These instances will be made available on www.vrp-rep.org on acceptance of this chapter.

³<http://project-osrm.org/>

possible configurations, while vehicles V_3 have three configurations. Costs for each vehicle type can be found in Table 3.4. Note that in terms of capacity, vehicle V_0 is a restricted case of vehicle V_3 . It is nevertheless an interesting choice as its fixed cost is considerably lower (€50 against €63). The fixed cost corresponds to the estimated vehicle ownership cost per day. The time-related cost corresponds to the driver’s wages, and the distance-related cost corresponds to the fuel consumption and vehicle use.

Vehicle type	Fixed cost	Time-related cost	Distance-related cost
V_0	€50	€23.81 /h	€0.17 /km
V_1	€79	€23.81 /h	€0.21 /km
V_2	€36	€23.81 /h	€0.12 /km
V_3	€63	€23.81 /h	€0.17 /km

Table 3.3: Vehicle types – costs

place Table 3.4 about here.

Type	Configuration 1		Configuration 2		Configuration 3	
	Seats	Wheelchairs	Seats	Wheelchairs	Seats	Wheelchairs
V_0	4	3	-	-	-	-
V_1	3	5	4	4	-	-
V_2	2	1	4	0	-	-
V_3	4	3	6	2	7	1

Table 3.4: Vehicle types – configurations

3.6.2 Evaluation of the metaheuristic components

Table 3.5 compares several LNS settings in order to evaluate the main components of the proposed metaheuristic. These experiments are performed on the presented real instances with the four vehicles described in Table 3.4.

The first column specifies the different LNS variants considered in this experiment. 5 runs of the full instance-set are computed for each variant. A time limit was set for each instance according to its size. For small instances (60-80 request), this time is 16 minutes. For medium-size instances (120-160 request) and large instances (280-295 request) the time is 40 and 100 minutes respectively. The following columns show average values over the full instance set. Please refer to A.2 for detailed results. Columns “Avg” correspond to the average cost over 5 runs of the instance set. The “Best” column corresponds to the average cost of the best run of each instance. The Gap for every instance is computed as $(value - BKS)/BKS \times 100$. The “Nb BKS” column is the number of best known solutions found by the heuristic (there are 14 instances in the instance set).

The first variant is a classic implementation of LNS which implements operators k-regret, best insertion, random removal, historical node-pair removal, worse removal, time-related removal and distance related removal, as described in Pisinger and Ropke (2007). This variant is denoted by LNS(5) because it implements 5 destroy operators. The second metaheuristic ALNS(5) adds-up the adaptive layer of Ropke and Pisinger (2006) to the former LNS implementation. The third metaheuristic ALNS(5)-SCP integrates the SCP component described in Section 3.4.3. The fourth variant, denoted LNS(2)-SCP, considers 2 destroy operators only (historical node-pair removal and random removal) and no adaptive layer. The fifth variant, denoted ALNS(5)-RSCP, is an extension of ALNS(5)-SCP integrating the reactive layer described in Section 3.4.3. The sixth variant, denoted LNS(5)-RSCP, does not include the adaptive layer. Finally, LNS(2)-RSCP extends the LNS(2)-SCP by including the reactive layer of Section 3.4.3. General parameters for all metaheuristics can be found in Table 4.5. Additional parameters for ALNS can be found in Table A.1

The first observation is that the SCP component brings significant improvement in the solution quality. In addition, it can be observed that the LNS can be considerably simplified when the SCP component is used. Indeed, the metaheuristic with two destroy operators performs as well as the versions with five destroy operators. Comparing LNS(2)-SCP with LNS(2)-RSCP, it can be seen that a second subsequent improvement is obtained when the SCP includes the reactive layer to adapt its parameters during the search. According to this set of experiments, LNS(2)-RSCP configuration seems to outperform the other variants, either in performance or in simplicity.

Variant	Avg Cost		Avg Gap		Nb BKS (14)
	Best	Avg	Best	Avg	
LNS(5)	2,348.51	2,357.59	3.60%	3.94%	3
ALNS(5)	2,349.06	2,369.08	3.61%	4.46%	2
ALNS(5)-SCP	2,289.81	2,300.03	1.34%	1.64%	7
LNS(2)-SCP	2,289.74	2,297.23	1.27%	1.52%	9
ALNS(5)-RSCP	2,235.25	2,242.52	0.13%	0.38%	9
LNS(5)-RSCP	2,235.36	2,241.60	0.12%	0.31%	12
LNS(2)-RSCP	2,236.46	2,243.91	0.13%	0.39%	10

Table 3.5: Performance comparison of LNS-based heuristics for our instances (see A.2 for detailed results)

Figure 3.6 shows the cost evolution considering the best run of four metaheuristics in instance I01-80. The other 3 variants are not shown because their performance is almost that of LNS(2)-RSP. Two separate groups of algorithms can be identified, corresponding to metaheuristics with and without an SCP component. The SCP increases not only solution quality, but also convergence speed.

Figure 3.7 shows the cost evolution on a large size instance I13-280. The variants without

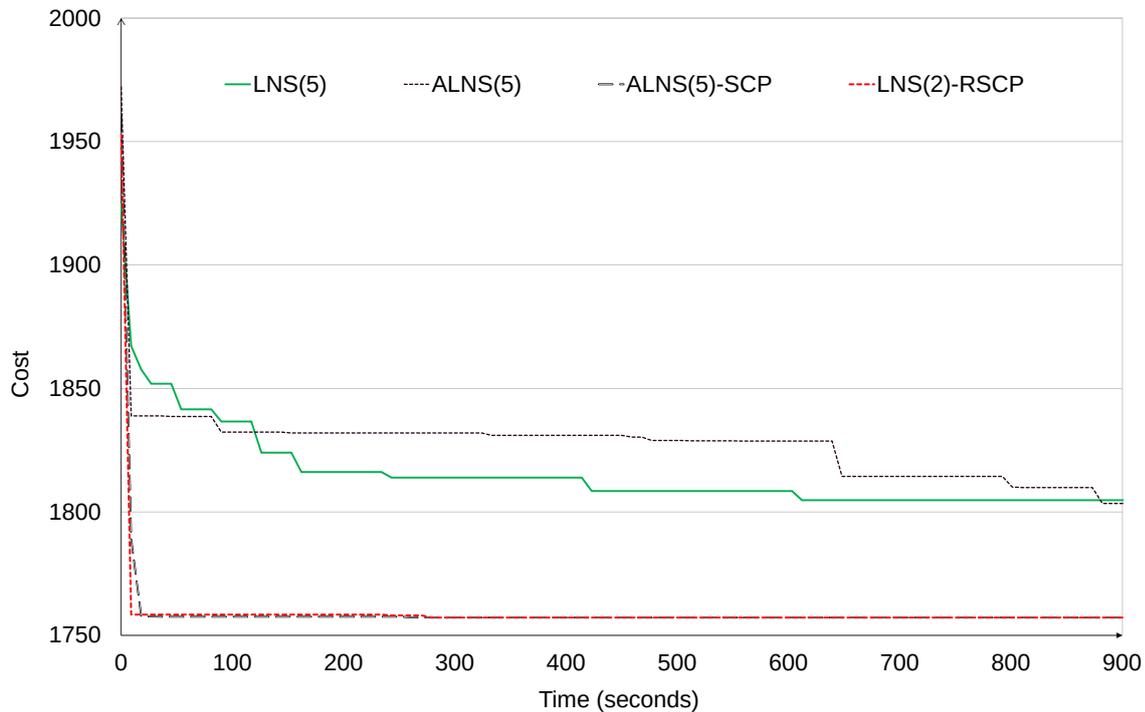


Figure 3.6: Impact SCP component (Instance I01-80)

the reactive layer fail to improve the solution significantly. This is because the SCP parameters are not adapted to this instances and the SCP is probably never solved optimally. Variants with the reactive layer perform better since the number of iterations between two calls to the SCP is reduced. This approach proves efficient in large instance as shown in the figure.

3.6.3 Performance evaluation on benchmarks from the literature

To evaluate the performance of LNS(2)-RSCP, we apply it to two sub-problems of the FSM-DARP-RC on benchmark instances. The first benchmark is made of the DARP instances of Cordeau and Laporte (2003) for which many elaborate methods were designed. In a second step, LNS(2)-RSCP is compared to the algorithm of Qu and Bard (2013) that was designed to solve a heterogeneous DARP with configurable vehicle capacity. This benchmark considers a limited fleet of vehicles that can be configured at the depot.

For the sake of conciseness, given our objective to solve large instances, we compare our algorithm to state-of-the-art metaheuristics. All these metaheuristics can easily and efficiently solve instances for which optimal solutions are known (refer to Gschwind and Drexel (2016) for the DARP and Qu and Bard (2015) for the HDARP-C).

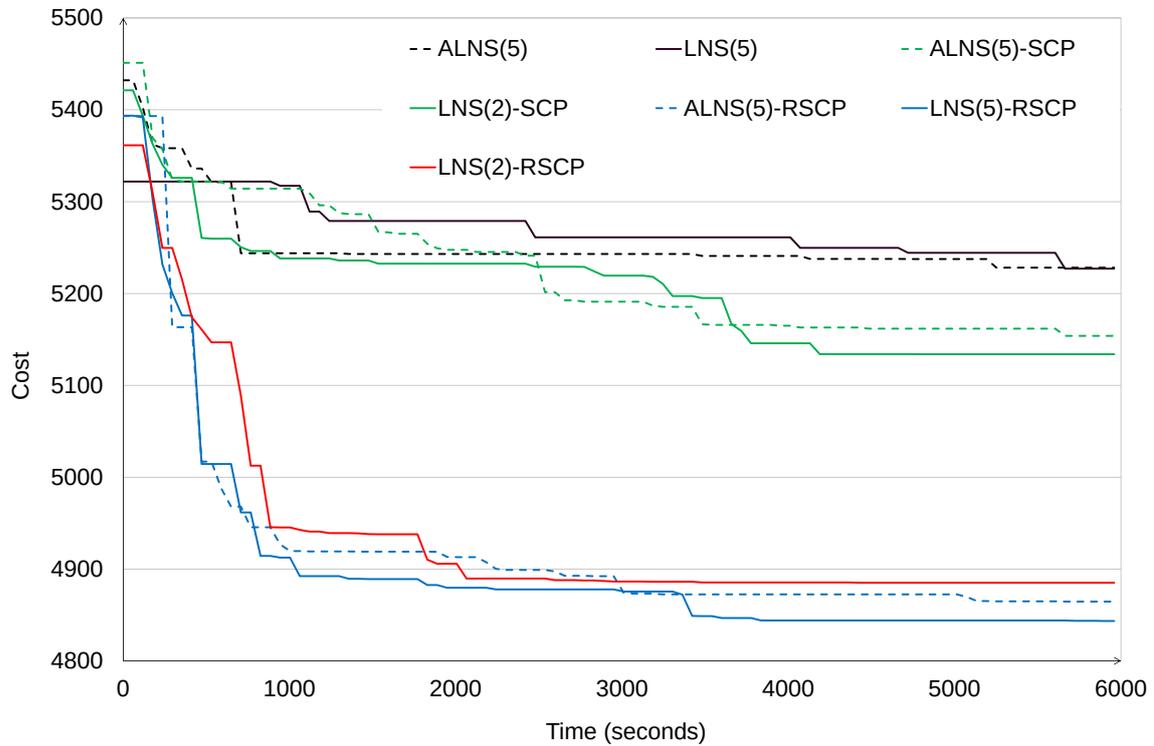


Figure 3.7: Reactive SCP (Instance I13-280)

3.6.3.1 Performance evaluation on the DARP

The instances of Cordeau and Laporte (2003) are still a reference point to evaluate the efficiency of algorithms on the DARP. Table 3.6 compares the results of the LNS(2)–RSCP, after 50,000 Iterations, to the latest heuristics of the literature that have been designed specifically to solve instances of Cordeau and Laporte (2003). Detailed results of each instance can be found in A.3. Gap values are computed in relation to the best known solutions (BKS) found in Gschwind and Drexler (2016) and have been found either by the cited meta-heuristics, or during the parameter tuning of Gschwind and Drexler (2016). Looking at the average gap (Avg Gap) we note that our metaheuristic competes with those three dedicated methods. LNS(2)–RSCP has the second-best average gap of 0.72% with 8 solutions among the best known (Nb. BKS), just after the ALNS of Gschwind and Drexler (2016) with 0.5% Avg Gap and 9 BKS. An interesting remark is that the reactive layer of the SCP component was barely activated since the initial values of $\eta = 1000$ and $t_{limit} = 3$ secs were sufficient for these instance sizes. The other two methods are the threshold acceptance (TA) based local search of Braekers et al. (2014) which has the highest average gap (1.16%) but also by far the lowest computation time, and the evolutionary local search (ELS) proposed by Chassaing et al. (2016).

Note that LNS(2)–RSCP is executed without any parameter tuning other than the one of the

FSM-DARP-RC instances. we do not scale computational times here as CPU information is not available for all methods and our goal is not to compete with these three methods. Nevertheless, these experiments show that the performance of LNS(2)–RSCP remain competitive on simpler benchmark problems.

Method	NbBKS (20)	Avg Time	Avg Obj		Avg Gap	
			Best	Avg	Best	Avg
Braekers et al. (2014) (TA)	3	83.2	512.06	514.31	0.81%	1.16%
Chassaing et al. (2016)(ELS)	6	592.4	511.14	513.67	0.64%	1.04%
Gschwind and Drexler (2016) (ALNS)	9	209.5	509.04	510.19	0.32%	0.50%
Ours (LNS2-RSCP)	8	727.1	509.53	511.62	0.38%	0.72%

Table 3.6: Benchmark classic DARP Cordeau and Laporte (2003) (Detailed results in Section A.3)

3.6.3.2 Performance evaluation on the heterogeneous DARP with configurable vehicle capacity

Another relevant benchmark was proposed by Qu and Bard (2013) for the heterogeneous dial-a-ride with configurable vehicle capacity (HDARP-C). The three main differences between this problem and the FSM-DARP-RC are: (i) a limited fleet of heterogeneous vehicles with configurable capacity; (ii) the vehicle configuration is decided at the depot, not en-route; and (iii) the presence of users with walkers in some instances, which means that two seats or one seat and half a wheelchair space are occupied.

These differences were integrated into our algorithm to run Qu and Bard (2013) instances. The SCP was adapted in a simple manner to consider a limited fleet of vehicles as follows: first, every route was explicitly assigned the vehicle type for which it was generated, and second, a constraint limiting the number of vehicles for each vehicle type was added to the SCP model.

In this benchmark, two instance sets of 100 requests each are proposed. A first set “A” is characterized by integrating various proportions of appointment requests (between 5% to 25%). The second instance set “B” instead differentiates the proportion of users with walkers.

According to Qu and Bard (2013), we denote by C_{total} , the total transport cost computed as a sum of the vehicle ownership cost, the time cost related to driver wages, and the penalty cost related to the passengers ride time. In Table 3.7 the LNS(2)-RSCP is compared to the multi-start ALNS (MSALNS) of Qu and Bard (2013). Each row corresponds to a different scenario. In all experiments LNS(2)–RSCP uses the same parameter tuning proposed in Table 4.5 with 3 runs per instance and a stopping criteria of 1 hour.

The first scenario, corresponding to the first row, considers the instance-set “A”. In “Gap Avg” column we observe an average total cost improvement of 1.02% over 3 runs. Column Gap(*) refers to the best average total cost improvement of the best solution found among

the 3 runs. In Gap(*), we obtain 1.34% improvement on average with respect to the MSALNS best solution. 6 new best solutions are found out of the 10 instances of the benchmark. These results demonstrate the good performance and stability of LNS(2)-RSCP from an experimental point of view.

A second scenario, corresponding to the second row, is performed on instance-set “B”. The main difference between set “B” and set “A” is the presence of very few time windows. Looking at the average gap (Gap Avg), we find more clearly that LNS(2)-RSCP outperforms the MSALNS with an 17.23% improvement in total cost. From detailed results presented in A.4, we find that LNS(2)-RSCP is barely affected by the proportion of walkers. This result is confirmed by last scenario, B-IWR, where walkers’ requirements are ignored. Without accounting for walkers, we do not find differences in performance among all instances of the B set (see Table A.6 for more details). Finally, looking at the average gap, we find that LNS(2)-RSCP improves the solution quality by 12.71%.

Scenario	Nb Inst.	MSALNS	LNS(2)-RSCP			
		Avg C_{total}	Avg C_{total}	Gap (*)	Gap Avg	NewBKS
A	10	49.23	48.19	-1.34%	-1.02%	6
B	10	46.25	37.98	-17.82%	-17.23%	10
B-IWR	10	42.69	37.22	-12.71%	-12.71%	10

Table 3.7: Benchmark Qu and Bard (2013) (detailed results in A.4)

3.6.4 Managerial insights

Having shown the efficiency of our matheuristic, we performed simulations in order to provide some general insights regarding, in particular, the relevance of en-route reconfiguration of vehicles. First, we measured the impact of en-route reconfiguration for two variants of the DARP. We then varied the fixed cost of reconfigurable vehicles to determine under which conditions it is worth buying the vehicles.

3.6.4.1 Vehicle fleet insights

In this section we analyze the gain of enabling en-route reconfigurations. Two variants of the DARP are compared with the real instance set, as shown in Table 3.8. Both variants employ the vehicle fleet of Table 3.4. The first variant, denoted as FSM-DARP-C, consist in allowing vehicles to be configured only once: at the depot, before starting their respective routes.

The second variant, denoted FSM-DARP-RC, allows vehicles to be reconfigured en-route. Strictly speaking FSM-DARP-RC is a relaxed problem of FSM-DARP-C, which explains why gains in the last column are always greater than 0. Cost reductions can go up to 2.45% of the total cost, depending on the instance.

Inst	Time	FSM-DARP-C			FSM-DARP-RC					
		Avg	Best	Routes	Avg	Best	Routes	Rec	RR	Savings
I01-80	16	1,774.74	1,774.74	18	1,757.22	1,757.22	18	1	1	0.99%
I02-60	16	924.68	923.51	8	916.22	912.28	8	1	3	1.22%
I03-80	16	1,838.57	1,838.47	15	1,838.41	1,838.33	15	1	1	0.01%
I04-70	16	1,602.48	1,602.48	18	1,599.49	1,599.49	18	1	1	0.19%
I05-80	16	1,242.55	1,242.48	15	1,242.48	1,242.48	15	0	0	0.00%
I06-80	16	1,419.81	1,418.74	12	1,418.75	1,418.74	12	0	0	0.00%
I07-60	16	1,546.61	1,546.61	11	1,537.63	1,537.63	11	2	3	0.58%
I08-65	16	1,223.65	1,223.58	10	1,205.11	1,204.89	10	1	2	1.53%
I09-120	40	3,167.81	3,166.29	23	3,129.75	3,129.15	23	2	5	1.17%
I10-135	40	2,094.77	2,068.71	17	2,083.10	2,061.52	17	1	3	0.35%
I11-160	40	2,283.03	2,271.57	21	2,249.11	2,233.19	21	3	7	1.69%
I12-160	40	2,659.34	2,653.40	24	2,653.78	2,629.19	23	2	3	0.91%
I13-280	100	5,027.30	5,008.69	37	4,901.06	4,885.77	36	4	13	2.45%
I14-295	100	5,017.01	4,982.85	45	5,026.49	4,981.14	44	2	8	0.03%

Table 3.8: Savings due to en-route configuration

Regarding the structure of solutions, let us consider the best solution (Best) found among the 5 runs for each instance. The next columns (Routes, Rec, RR) refer to some characteristics of these Best solutions. Passing from FSM-DARP-C to FSM-DARP-RC, we observe that the number of routes (Routes) is similar for most instances. However, in some instances, like I14-280, cost savings are related to a reduction in the number of routes (from 37 to 36).

The maximum number of performed reconfigurations among the solution routes (Rec) shows that en-route reconfiguration is usually performed once or twice inside a route. The number of routes performing en-route reconfiguration (RR) indicates how many configurable vehicles are actually reconfiguring en-route in the whole solution. For example, in instance I02-60, we find that 3 out of 8 routes actually reconfigure en-route, this is 37.5% of the vehicle fleet. In average this proportion is 10.28% for small instances, 21.43% for medium-size and 26% for large instances.

3.6.4.2 Fixed cost analysis

So far, we have seen that reconfigurations can allow cost reductions; however, the resulting gain is correlated to the cost of buying reconfigurable vehicles rather than standard ones. Figure 3.8 pictures how, for the instance in I13-280, the percentage of reconfigurable vehicles decreases when their cost increases compared to the cost of standard vehicles. In this figure, the point with coordinates (2%;12%) means that re-configurable vehicles are 2% more expensive than standard vehicles, and in the cheapest solution found, 12% of the vehicles are reconfigurable. In the same way, when reconfigurable vehicles are 20% more expensive than standard vehicles, they are completely excluded from the solution.

The set of vehicle types considered in these experiments can be found in Table 3.9 and their

respective costs in Table 3.10.

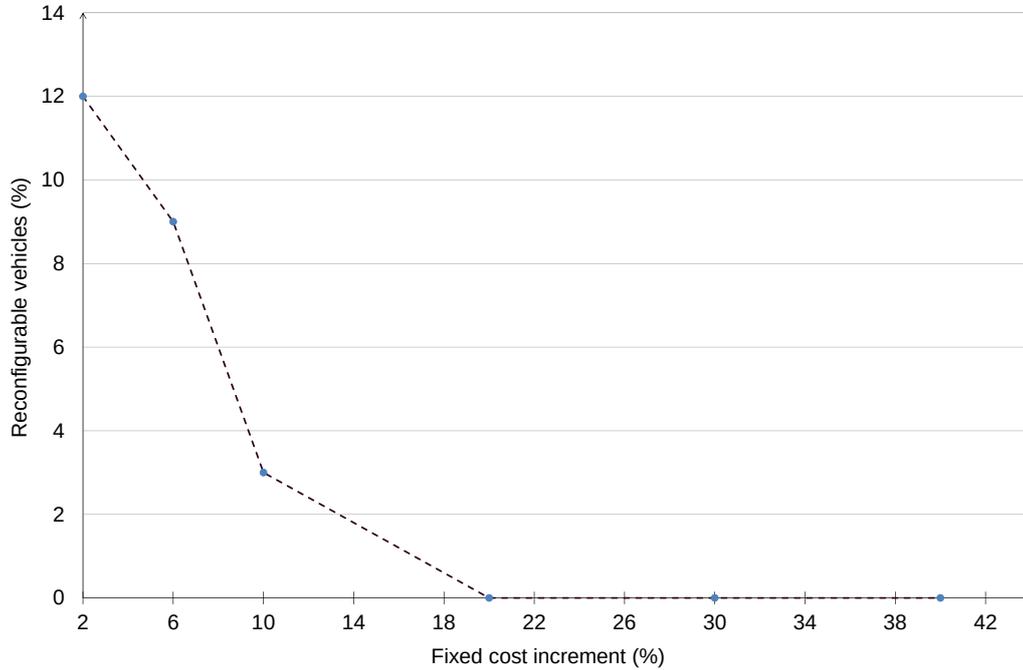


Figure 3.8: Fixed cost impact of reconfigurable vehicles (Instance I13-280)

Vehicle type	Configuration 1		Configuration 2		Configuration 3	
	Seats	Wheelchairs	Seats	Wheelchairs	Seats	Wheelchairs
V_4	7	1	-	-	-	-
V_5	6	2	-	-	-	-
V_6	4	3	-	-	-	-
V_7	4	3	6	2	7	1

Table 3.9: Characteristics of vehicle types for fixed cost analysis

3.7 Conclusion

In this chapter, we investigated a new variant of the dial-a-ride problem characterized by en-route reconfiguration of vehicle capacity. This feature was studied in the context of door-to-door transportation of children with disabilities, considering heterogeneous users and vehicles. We aimed to determine the size and composition of the fleet as a strategic decision. A metaheuristic based on Large Neighbourhood Search and a Reactive Set Covering Problem (LNS-RSCP) was proposed to solve this problem for real instances of up to 295 user requests. Results on the real data set showed significant performance compared with six other LNS-based metaheuristics

Vehicle type	Fixed cost	Time-related cost	Distance-related cost
V_4	€50	€23.81 /h	€0.17 /km
V_5	€50	€23.81 /h	€0.17 /km
V_6	€50	€23.81 /h	€0.17 /km
	(+2%) €51		
	(+6%) €53		
V_7	(+10%) €55	€23.81 /h	€0.17 /km
	(+20%) €60		
	(+30%) €65		
	(+40%) €70		

Table 3.10: Costs of vehicle types for fixed cost analysis

variants. Experiments show that the SCP component increases not only solution quality but also convergence speed. LNS–RSCP was also tested on literature instances that achieved competitive results for the classic DARP and outstanding results in the heterogeneous DARP with configurable vehicle capacity. Although en-route reconfiguration is not a usual practice in companies, companies often own configurable vehicles, as in the case study. Yet route designers do not plan routes considering this extra degree of flexibility. In this study we show that companies can easily save up to 2.5% in the total cost just by allowing vehicles to use en-route reconfiguration. Finally, we show that the utilization of reconfigurable vehicles is strongly dependent on the vehicle ownership cost (fixed cost). For the evaluated instance, supposing that all operations are the same on each day, we found that reconfigurable vehicles are advantageous for companies when their cost are no more than 20% of the cost of standard non-reconfigurable vehicles. Further research includes the extension of the problem studied in this chapter with operational features. First, we plan to study passengers time consistency in a multi-periodic transportation framework (see, e.g. Kovacs et al. (2014a); Feillet et al. (2014)). Then, incorporating real-time information (traffic, variable demand) may require the use of state-space-time networks with a fine time discretization (see, e.g. Mahmoudi and Zhou (2016)) and specific optimization algorithms such as the Dynamic Discretization Discovery (Boland et al., 2017).

Weekly transport problem: The time-consistent dial-a-ride problem

In the context of door-to-door transportation of people with disabilities, service quality considerations, such as maximum ride time and service time consistency, are critical requirements. These requirements together with traditional route planning define a new variant of the multi-period dial-a-ride problem called the time-consistent DARP. A perfectly consistent planning defines for each passenger the same service time all along the planning horizon. This planning can be too expensive for Medico-Social Institutions that it is necessary to find a compromise solution between costs and time-consistency objectives. The time-consistent DARP is solved using an epsilon-constraint approach to illustrate the trade-off between these two objectives. In this chapter, the time-consistency is defined by the number of different timetables for each user. Each solution of the Pareto Front is computed using a matheuristic framework based on a master set partitioning problem and a large neighborhood search procedure. This approach is benchmarked in the context of time-consistent vehicle routing problem with instances from the literature. The context of door-to-door transportation for people with disabilities is also experimented by using real data provided by GIHP Company. Those experiments allow us to propose some managerial insights regarding the interaction between costs and service quality.

Keywords: Dial-a-ride problem, consistency, set partitioning, large neighborhood search, inconsistency

4.1 Introduction

The design of efficient para-transit systems relies both on minimizing operational costs and providing users with an adequate quality of service. In the operations research literature, the Dial-A-Ride Problem (DARP) is a well-known optimization problem that consists of designing minimal cost vehicle routes to fulfill a set of transportation requests while satisfying several quality service requirements. Common applications concern door-to-door transportation of elderly people or people with disabilities. In Medico-Social Institutions (MSI) in France, transportation is considered as the main expense after wages ANAP (2016a). Transportation plans are defined every year and partially revised several times a year whenever necessary. Due to the high pressure on expenses, minimizing costs is often the main objective function of MSIs.

The DARP formulation considers a single period, typically half a day. Passengers are generally subject to ride time constraints: they must not travel longer than a predefined duration. In this chapter, we address the case of para-transit systems for people who need to be transported regularly, for example, handicapped workers or scholars. The DARP formulation is extended over multiple periods and each period has a known set of transportation demands coming from passengers. Most passengers emit the same transportation demand every day but it is common to observe variations (attendance, pickup or destination modifications according to medical appointments...). A passenger demand variation may impact the schedule of other passengers in its route. As a result, a need for regularity (or consistency) of service times is expressed from passengers and MSIs: for medical, cognitive or convenience reasons, it is desired that a passenger who has the same demand on several week-days has the same pickup/drop off time these days. This chapter aims at designing an algorithm to be integrated into a dial-a-ride application for passengers with regularity requirements.

This work has been motivated by a real-life case study in the area of Lyon, France. The transportation in the area is mainly operated by a single carrier¹ who works for multiple MSIs with a fleet of adapted vehicles. Every morning, from Monday to Friday, disabled children from the region are transported from their home to one MSI. In the afternoon, they are driven back home. Without loss of generality, this chapter presents the result of our research for morning trips. Thus, we address a multi-period dial-a-ride problem and study the trade-off between service time consistency and transportation costs. As this problem introduces time consistency within a DARP setting, we call this new variant the Time-Consistent DARP (TC-DARP). This research has been conducted in tight cooperation with SMIs and the passenger transportation company.

The chapter is organized as follows: Section 4.2 presents how the TC-DARP is related to the existing literature in operations research. In Section 4.3, we give a formal definition of the TC-DARP and formulate it as a mixed-integer linear program (MILP). Sections 4.4 presents a general approach for solving the TC-DARP. Sections 4.5 detailed the algorithm used

¹GIHP: www.gihp-sa.com

for generating the routes. In Section 4.6, computational results and management insights are reported.

4.2 Literature review

The mono-period DARP of our application has been presented in Tellez et al. (2018b). We focus our literature review on the consistency aspects that appear in the multi-periodic version of the problem.

The integration of time consistency has appeared recently in the vehicle routing problem (VRP) literature. Applications were first identified in the context of fast parcel delivery (Groër et al., 2009) and rapidly extended to passengers transportation Feillet et al. (2014). Readers interested in an extensive review of vehicle routing with consistency considerations can refer to Kovacs et al. (2014a). Consistency is very routing problems that can be divided into three main categories: service time consistency, driver consistency or territory consistency. *Service time consistency* refers to visiting regular customers at approximately the same hour during the planning horizon. As the main focus of our chapter, the service time consistency will be detailed in the next section.

Driver consistency consists in minimizing the of different drivers assigned to each passenger during the planning horizon. This aims at reinforcing the relationship between drivers and passengers to improve the quality of service. Braekers and Kovacs (2016) computed the average cost of a solution where each passenger is served by one, two and three drivers, respectively, showing that a solution with two drivers can be near-optimal whereas solutions with one driver are 10% costlier on average. Other approaches using soft constraints result in similar conclusions (see Smilowitz et al. (2013); Milburn and Spicer (2013)). In Feillet et al. (2014), drivers are assigned to routes a posteriori, so that service time consistency and drivers consistency are considered as independent problems in a lexicographical way.

Territory consistency aims at increasing drivers' efficiency through their knowledge of the geographical area where they operate. A common way of addressing territory consistency is to design independent districts in advance where independent routing problems are solved every day. This approach was studied in Lei et al. (2012); Zhong et al. (2007); Smilowitz et al. (2013); Schneider et al. (2015).

This chapter focuses on service time consistency applied to a Dial-a-Ride Problem (DARP). In contrast to the VRP, the DARP considers one origin and destination for each user and maximum ride time. The main applications of the DARP concern door-to-door transportation of people, particularly elderly or disabled people (Ho et al., 2018; Lehuédé et al., 2014).

4.2.1 Service time consistency models

Service time consistency consists in serving regular needs at approximately the same hour during the whole planning horizon. This is either modeled by hard constraints, that is, imposing an acceptable level of service time variation, or by soft constraints, that is, penalizing service time variations in the objective function.

Groër et al. (2009) defined the maximum arrival time variation as the difference between the latest and earliest service times during the whole planning horizon for each customer. This consistent VRP (conVRP) is an extension of the multi-period VRP where the maximal arrival time variation is bounded above by a constant value L_{\max} . However, this measure, initially proposed for the small package shipping industry, has some practical drawbacks in the context of people transportation.

Feillet et al. (2014) defines a passenger-oriented time consistency model based on the concept of time-classes. The authors assume that very small variations (for example 1 or 2 minutes) in-service time are not significant for users, especially considering approximations and variations due to traffic conditions or unexpected events. What accounts for passengers is the number of significantly different service times proposed in a week. Similar times are regrouped in the same time-class. Regularity is then improved by minimizing the maximum number C_{\max} of time-classes overall users. The difference between this measure and L_{\max} is highlighted in Figure 4.1-4.2.

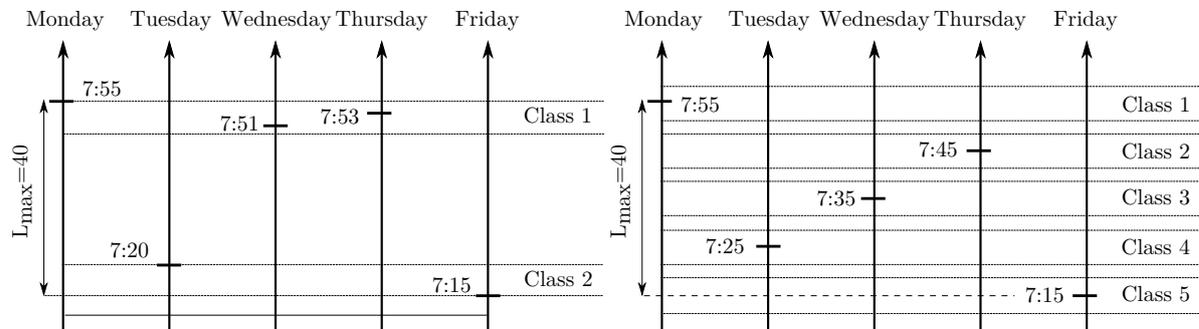


Figure 4.1: Solution with $C_{\max} = 2$ time-classes Figure 4.2: Solution with $C_{\max} = 5$ time-classes

Figures 4.1 and 4.2 represent the service time of a passenger in two distinct solutions. Each vertical line represents the service time from Monday to Friday. In Figure 4.1, these times can be regrouped into two intervals of 10 minutes width: $[7:00-7:10]$ and $[7:50-8:00]$. This passenger is said to have 2 time-classes. In Figure 4.2, service times are evenly spread between 7:00 and 8:00, leading to 5 time-classes. While both solutions have the same value $L_{\max} = 1$ hour, they do not offer the same consistency to passengers as far as service time is concerned. In our application, a measure based on the number of time-classes offers a better quality of service than a solution measure with L_{\max} .

Nevertheless, there is one limitation related to using objective C_{\max} . It is the largest number

of time-classes in the solution overall passengers. Hence, a solution with 99% of users having C_{\max} time-classes is equivalent to another solution having only 1% of users in the same situation. To overcome this limitation, we propose a lexicographic optimization. We first minimize the number of passengers from the highest to the lowest number of time classes: first passengers with C_{\max} time-classes, then $C_{\max} - 1$, $C_{\max} - 2$ and so forth.

To the best of our knowledge, this approach is a new refinement of the Feillet et al. (2014) model. Still, according to the passenger transportation company, many good trade-off solutions fit between the cost-optimal solution with $C_{\max} = 2$ and the cost-optimal solution with $C_{\max} = 1$. According to the fair optimization literature Ogryczak et al. (2014), the proposed model corresponds to a lexicographic minimax refinement of the min-max model, using counting functions. We show that this lexicographic objective adapts well to the context of passenger transportation.

4.2.2 Solution approaches for time-consistent routing problems

In the conVRP model introduced by Groër et al. (2009), the objective is to optimize service time consistency (L_{\max}) without compromising a perfect driver consistency (1 driver per customer). The authors proposed a record-to-record travel algorithm and developed benchmark instances for up to 100 customers.

The consistency measure L_{\max} has been used in several subsequent papers (i.e. Sungur et al. (2010); Tarantilis et al. (2012); Kovacs et al. (2014b, 2015a); Luo et al. (2015); Xu and Cai (2018)). The current best results on Groër et al. (2009) benchmark instance set is obtained by Xu and Cai (2018), who proposed a Variable Neighborhood Search procedure using dedicated local search methods for quickly solving local optima. This approach is based on improving template solutions generated by 3 different shaking methods. A problem extension, denoted the genConVRP, is proposed by Kovacs et al. (2015a) in which: routes do not necessarily start at the same time, customers are associated with AM/PM time windows and a maximum number of drivers per customer is defined. Subramanyam and Gounaris (2016) proposes a branch-and-cut framework to solve the consistent traveling salesman problem which is a particular case of the conVRP using a single-vehicle. They solve randomly generated instances with up to 51 customers.

The Time-Consistent VRP (TCVRP) of Feillet et al. (2014) is solved with a dedicated Large Neighborhood Search (LNS) framework. At each iteration, the routes of all periods are destroyed. A VRP with multiple time windows and no waiting time (VRPmTW-nw) is defined to reduce the number of time-classes of one passenger. A heuristic a branch-and-price is used to recreate the routes. The minimum cost solutions for $C_{\max} = 1$ to 5 are saved in the process.

Another related work is the Time Window Assignment Vehicle Routing Problem (TWAVRP) introduced by Spliet and Gabor (2015). In the TWAVRP, a single time window of fixed width has to be assigned to some regular customers before the effective daily demand is known.

The assignment is based on a set of demand scenarios, each one being associated with a given probability. The objective is to minimize the expected traveling cost. The TWAVRP is a particular case of the genConVRP if scenarios are seen as periods and the number of drivers per user is set to infinity. However, the objective differs: genConVRP optimizes the total transportation cost and TWAVRP the average transportation cost. A branch-price-and-cut algorithm is proposed to optimally solve instances with up to 25 customers. Spliet and Desaulniers (2015) proposes a variant called the discrete-time window assignment vehicle routing problem where the chosen time windows are selected from a discrete set.

Consistency issues are also often related to having stochastic customers in the VRP (Ritzinger et al., 2016). For example, Sungur et al. (2010) uses a combination of robust optimization in a first phase master problem and stochastic programming with recourse for daily schedules to address the uncertainty in service times and customer occurrence. Erera et al. (2009) investigate the opportunity to give a main fixed route as well as a backup one to frequent customers in a stochastic context.

Finally, the question of service time consistency presents some similarities with some non-periodic applications such as the synchronization of multiple vehicles at the same node. In this case, the arrival time of multiple vehicles at a given location should be synchronized to perform a collective operation. Then, vehicles continue their routes independently. A survey on synchronization in VRP is given by Drexel (2012).

4.2.3 Contributions to the literature

The literature review shows that there is still some gap between what has been proposed in the literature and practical implementation of time consistency for a DARP application.

In this chapter, we use the notion of time-classes introduced by Feillet et al. (2014) and explore solutions where some users accept several time-classes. Compared to the TCVRP proposed in Feillet et al. (2014), we propose a refinement of the C_{\max} minimization approach.

Regarding the VRP attributes, we investigate a more realistic setting than Feillet et al. (2014). In particular, we consider time windows and maximum ride times in a problem with multiple MSIs (e.g. multiple pickup and delivery locations). As a result, solutions might contain routes that contain waiting times. We keep the assumption that improving consistency by artificially introducing waiting times within routes is not realistic for drivers. Similarly to Kovacs et al. (2015a), we consider that the departure time of routes can be changed to improve consistency.

As we consider a paratransit user application, we will refer to passengers or patients as *users* in the remaining of the chapter.

4.3 Modeling the consistent dial-a-ride-problem (TC-DARP)

We consider a set of users \mathcal{U} to be transported during a planning horizon \mathcal{T} . Users may require a particular space $v \in \mathcal{V}$ in the vehicle such as seats or wheelchair spaces. There is a homogeneous fleet of vehicles based at a single depot o . The vehicle capacity is defined by a vector $\mathcal{Q} = \{Q_1, \dots, Q_{|\mathcal{V}|}\}$ representing the availability of each space type $v \in \mathcal{V}$.

Each user $u \in \mathcal{U}$ has a pickup node denoted by $p_u \in \mathcal{P}$, a delivery node $d_u \in \mathcal{D}$, a maximum ride time \bar{T}_u , and a demand indicator $\beta_u^t \in \{0, 1\}$ for each period $t \in \mathcal{T}$. $\beta_u^t = 1$ if user u requires transportation at period t and 0 otherwise. Each user can be serviced at most once on any period. Note that if two users have a common origin or destination, nodes are duplicated so that each pickup node and each delivery node has exactly one user.

The TC-DARP is defined on a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ with the vertex set $\mathcal{V} = \{\mathcal{P} \cup \mathcal{D} \cup o\}$ and the arcs set \mathcal{A} contains arcs of the following forms: (o, i) where $i \in \mathcal{P}$; (i, j) where $i, j \in \mathcal{P} \cup \mathcal{D}, i \neq j$; and (i, o) where $i \in \mathcal{D}$. Each node $i \in \mathcal{V}$ is associated with a service duration ζ_i and a time window $[a_i, b_i]$. Every arc (i, j) represents the fastest path from node i to node j and is associated with a travel time t_{ij} and a distance d_{ij} .

We propose a route based MILP formulation of the TC-DARP. A route is any feasible sequence of nodes visited by a vehicle. Each route ω starts and finishes at node o and is characterized by a set of visited pickup and delivery nodes. Every pickup or delivery node $i \in \mathcal{P} \cup \mathcal{D}$ on a route ω is associated with a time $H_{i,\omega}$. $H_{i,\omega}$ is the earliest possible service time of node i in a schedule of ω that minimizes its duration and respect time windows and maximum ride time constraints for each user in the route. In addition, for each route a *maximum forward time shift* Δ_ω^+ which represents the maximal amount of time by which its departure time can be postponed, without violating any time window or ride time constraints. The detailed scheduling procedure with the calculation Δ_ω^+ values is presented in Appendix B.0.4.

Each route $\omega \in \Omega$ is operated by one vehicle. Each vehicle is associated: weekly ownership cost λ , a cost per kilometer τ related the fuel consumption, and a cost per hour α related to driver cost. The TC-DARP can be seen as a bi-objective problem which consists of selecting a subset of routes from Ω such that all transportation requests on the planning horizon are satisfied within their time windows and maximum rime times. The first objective is to minimize the sum of fixed and variable traveling costs. The second objective is to minimize service time inconsistency.

Note that we do not consider explicitly users with more than one address. Actually, time consistency is meaningful for a given address. A person with two distinct addresses is modeled as two different people (one per address).

Tables 4.1, 4.2 and 4.3 synthesizes the mathematical notations for the sets, data and variables used in the TC-DARP mathematical model.

To model the problem, we introduce the binary decision variable y_ω^t which is equal to 1 if route $\omega \in \Omega$ is selected at period $t \in \mathcal{T}$. The binary variable z_{uc}^t is equal to 1 if user $u \in \mathcal{U}$ is

\mathcal{U}	set of users
\mathcal{T}	set of time periods
\mathcal{T}_u	set of time periods in which user $u \in \mathcal{U}$ needs to be transported.
Ω	set of all routes
Ω_u	set of routes serving user $u \in \mathcal{U}$
\mathcal{C}	set of time-classes

Table 4.1: Sets

β_u^t	equal to 1 if user $u \in \mathcal{U}$ must be serviced on period $t \in \mathcal{T}$, and 0 otherwise
C_ω	variable cost of route $\omega \in \Omega$
λ	weekly vehicle cost ownership cost.
$H_{u\omega}$	earliest time service to user $u \in \mathcal{U}$ by route $\omega \in \Omega$
Δ_ω^+	maximum forward time shift of route $\omega \in \Omega$
Λ	time-class width

Table 4.2: Data

<i>Binary Variables</i>	
$y_\omega^t \in \{0, 1\}$	=1 if route $\omega \in \Omega$ is selected at period $t \in \mathcal{T}$, and 0 otherwise
$z_{uc}^t \in \{0, 1\}$	=1 if user $u \in \mathcal{U}$ is assigned to time-class $c \in \mathcal{C}$ at period $t \in \mathcal{T}$, and 0 otherwise
$\mu_{uc} \in \{0, 1\}$	=1 if user $u \in \mathcal{U}$ uses time-class $c \in \mathcal{C}$, and 0 otherwise
<i>Other variables</i>	
$\delta_\omega^t \in [0, \Delta_\omega^+]$	time shift (used margin) of route $\omega \in \Omega$ at period $t \in \mathcal{T}$
$s_{uc}^-, s_{uc}^+ \in \mathbb{R}^+$	lower and upper bounds for the time-class $c \in \mathcal{C}_u$ for user $u \in \mathcal{U}$
$h_u^t \in \mathbb{R}^+$	beginning of service for user $u \in \mathcal{U}$ at period $t \in \mathcal{T}$
v	number of vehicles needed for the whole planning horizon
m_u	number of users having $c \in \mathcal{C}$ time-classes (post-processed variable)

Table 4.3: Variables

assigned to time-class $c \in \mathcal{C}$ at period $t \in \mathcal{T}$. Binary variables μ_{uc} indicate which time-classes from set \mathcal{C} are actually used by user u . For example, if u has 3 time-classes, it is said to use classes 1, 2 and 3. Hence, $\mu_{u,1} = \mu_{u,2} = \mu_{u,3} = 1$. The shift of route $\omega \in \Omega$ departure time at period $t \in \mathcal{T}$ is expressed by continuous variables $\delta_\omega^t \in [0, \Delta_\omega^+]$.

Objectives

Two objective functions are defined for the TC-DARP. The first one is the minimization of transportation costs and the second one is the minimization of time inconsistency. Transportation costs are defined by the sum of fixed and routing costs. Fixed costs are related to the cost of owning the vehicles. Then, the cost C_ω of a route $\omega \in \Omega$ depends on its duration (which may include some waiting time) and on its length.

$$\min f = \lambda v + \sum_{\omega \in \Omega} \sum_{t \in \mathcal{T}} C_{\omega} y_{\omega}^t. \quad (4.1)$$

The second objective minimizes time inconsistency, which is modeled with a lexicographical refinement of the time-class model proposed by Feillet et al. (2014). The expression which is used in the MILP model is the following:

$$\text{lexmin } \hat{\mathbf{g}} = \left(\sum_{u \in \mathcal{U}} \mu_{u|\mathcal{C}|}, \dots, \sum_{u \in \mathcal{U}} \mu_{u2} \right) \quad (4.2)$$

This expression lexicographically minimizes the number of people having more than c time-classes, where c decreases from $|\mathcal{C}|$ to 2. The expression $\sum_{u \in \mathcal{U}} \mu_{uc}$ counts the number of users with c or more time-classes.

This is equivalent to the lexicographical minimization of the number of users with exactly $|\mathcal{C}|$ classes, $|\mathcal{C}| - 1$ classes, down to 1 classes. In the remaining of the chapter, we denote by

$$\text{lexmin } \mathbf{g} = (m_{|\mathcal{C}|}, \dots, m_1) \quad (4.3)$$

the alternative formulation of this objective, where m_c denotes the number of users having c time-classes. It is post-processed from the values of the μ_{uc} variables using the following expressions:

$$\begin{cases} m_1 = |\mathcal{U}| - \sum_{u \in \mathcal{U}} \mu_{u,2} \\ m_c = \sum_{u \in \mathcal{U}} \mu_{uc} - \sum_{u \in \mathcal{U}} \mu_{u,c+1} \quad \forall c \in \{1, \dots, |\mathcal{C}| - 1\} \\ m_{|\mathcal{C}|} = \sum_{u \in \mathcal{U}} \mu_{u|\mathcal{C}|} \end{cases} \quad (4.4)$$

Constraints

The set of TC-DARP feasible solutions is defined by the following constraints:

$$\sum_{\omega \in \Omega_u} y_{\omega}^t = \beta_u^t \quad \forall u \in \mathcal{U}, t \in \mathcal{T}, \quad (4.5)$$

$$\sum_{\omega \in \Omega} y_{\omega}^t \leq v \quad \forall t \in \mathcal{T}, \quad (4.6)$$

$$\sum_{c \in \mathcal{C}} z_{uc}^t = 1 \quad \forall u \in \mathcal{U}, t \in \mathcal{T}_u, \quad (4.7)$$

$$h_u^t = \sum_{\omega \in \mathcal{R}_u} (H_{u\omega} y_{\omega}^t + \delta_{\omega}^t) \quad \forall u \in \mathcal{U}, t \in \mathcal{T}_u, \quad (4.8)$$

$$\delta_{\omega}^t \leq \Delta_{\omega}^+ y_{\omega}^t \quad \forall \omega \in \Omega, t \in \mathcal{T}, \quad (4.9)$$

$$s_{uc}^- \leq h_u^t + M(1 - z_{uc}^t) \quad \forall c \in \mathcal{C}, u \in \mathcal{U}, t \in \mathcal{T}_u, \quad (4.10)$$

$$h_u^t \leq s_{uc}^+ + M(1 - z_{uc}^t) \quad \forall c \in \mathcal{C}, u \in \mathcal{U}, t \in \mathcal{T}_u, \quad (4.11)$$

$$(4.12)$$

$$s_{uc}^+ - s_{uc}^- = \Lambda \quad \forall c \in \mathcal{C}, u \in \mathcal{U}, \quad (4.13)$$

$$s_{uc}^+ \leq s_{uc+1}^- \quad \forall c \in \mathcal{C}/\{|\mathcal{C}|\}, u \in \mathcal{U}, \quad (4.14)$$

$$\sum_{\omega \in \mathcal{R}_u} y_r^t = \sum_{c \in \mathcal{C}} z_{uc}^t \quad \forall u \in \mathcal{U}, t \in \mathcal{T}_u, \quad (4.15)$$

$$z_{uc}^t \leq \mu_{uc} \quad \forall c \in \mathcal{C}, u \in \mathcal{U}, t \in \mathcal{T}_u, \quad (4.16)$$

$$\mu_{uc+1} \leq \mu_{uc} \quad \forall c \in \mathcal{C}, \forall u \in \mathcal{U}, \forall t \in \mathcal{T}, \quad (4.17)$$

$$y_\omega^t, z_{uc}^t, \mu_{uc} \in \{0, 1\} \quad \forall c \in \mathcal{C}, u \in \mathcal{U}, t \in \mathcal{T}, \omega \in \Omega, \quad (4.18)$$

$$\delta_\omega^t, h_u^t s_{uc}^-, s_{uc}^+, v \in \mathbb{R}^+ \quad \forall c \in \mathcal{C}, u \in \mathcal{U}, t \in \mathcal{T}, \omega \in \Omega. \quad (4.19)$$

Constraints (4.5) are partitioning constraints ensuring the satisfaction of the users demand. Constraints (4.6) count the number of vehicles needed during the planning horizon. Constraints (4.7) state that each user served in period $t \in \mathcal{T}$ should be given a single time-class. Constraints (4.8) determines the service time for each user of route ω when its departure is shifted by the value δ_ω^t . Constraints (4.10) and (4.11) linearize the following logical expression:

$$z_{cut} = 1 \Rightarrow s_{uc}^- \leq h_u^t \leq s_{uc}^+ \quad \forall u \in \mathcal{U}, t \in \mathcal{T}_u, c \in \mathcal{C}. \quad (4.20)$$

They state that if a user u is assigned to the time-class c at period t , then its service time should be within the bounds of this time-class c . Constraints (4.13) set the width of a time-class. Constraints (4.14) avoid overlap between time-classes. Constraints (4.15) links the number of routes that serve one given user and the number of time-class variables. Constraints (4.16) define variables μ_{uc} necessary for counting the number of time-classes of each user. Constraints (4.17) ensures that time-classes are defined in increase order. For example, time-class #2 is allocated to a user only if time-class #1 already exists and is not compatible with a given service time. Finally, variables definition is given by constraints (4.18) and (4.19).

4.4 Solution method

In this section, we present the solution method to solve the bi-objective TC-DARP. This method consists of a Master Set Partitioning Problem (MSPP) that is fed with a set of routes generated through a Large Neighborhood Search algorithm (LNS). This section is structured as follows: Section 4.4.1 presents the general matheuristic framework named MSPP that traces the Pareto front between the two objectives of the TC-DARP. This framework relies on two main components. Section 4.4.2 presents the first component of the framework that is the construction of the initial solution. The second component is the mono-objective optimization procedure of the TC-DARP presented in Section 4.4.3. Then, Section 4.4.4 presents the rules to select the routes that are used during the solving.

4.4.1 The matheuristic framework (MSPP)

In this section, we present a general framework (see Algorithm 6) based on a ε -constraint procedure to trace the Pareto front between the two objectives, the transportation cost f and the time inconsistency g (see Figure 4.3). In a nutshell, the algorithm starts from the best solution found with cost reduction objective and gradually improving the inconsistency objective by allowing an increase of the transportation cost by ε percent to find new non-dominated solutions along the Pareto front. The procedure stops when every user has 1 time-class.

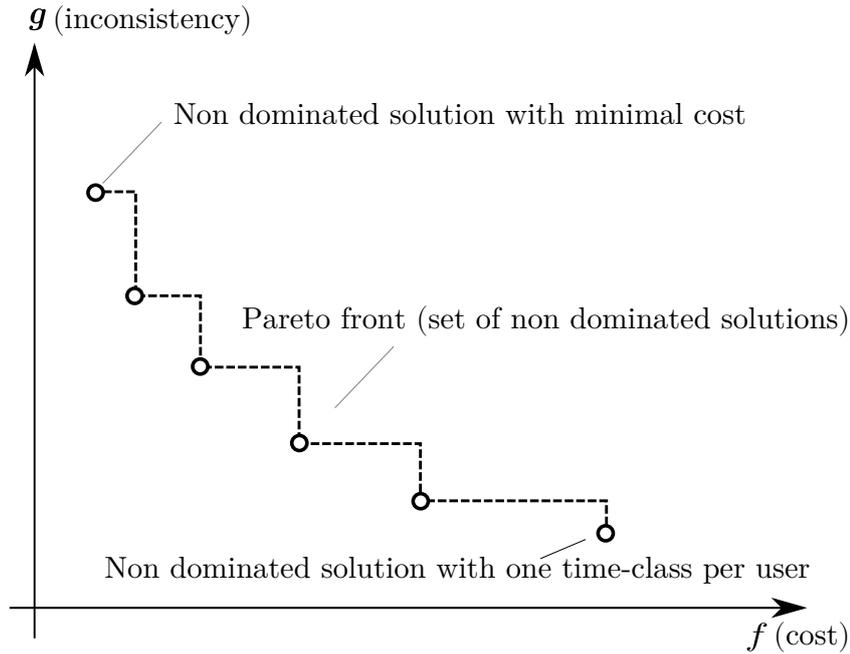


Figure 4.3: Pareto

Algorithm 6 presents the MSPP framework. In this algorithm, the pool of routes \mathcal{L} is generated at the same time as the initial solution. Two solutions are defined, a temporal solution S (line 3) and a best found solution S^* , initialized with the procedure described in Section 4.4.2 (line 4). This last procedure consists in solving a multi-period DARP with cost objective only. Note that this procedure generates the pool \mathcal{L} during the solving of the multi-period DARP. This solution is required to find the cost upper bound \bar{f} (line 5).

Lines 7 to 15 describe an iteration of the algorithm. In order to find a new temporal solution S two procedures detailed in Section 4.4.3 have to be executed (see Figure 4.4). The first procedure is the optimization of the inconsistency g subject to a maximal cost \bar{f} (line 7). The value of the best inconsistency found after this first optimization is $g(S)$. Given that, the procedure starts with a feasible solution S^* , the result of the first procedure ensures that $g(S) \leq_{lex} g(S^*)$. The \leq_{lex} operator represents a lexicographic comparison between the two vectors. This comparison can also be an strictly lower, in that case written as $<_{lex}$.

Algorithm 6: Matheuristic framework (MSPP)

Data: ε : initial value epsilon, ϕ : Increase factor of epsilon,
Result: Pareto front

```

/* Initialization */
1 ParetoFront  $\leftarrow \emptyset$ : Empty Pareto front
2  $\mathcal{L} \leftarrow \emptyset$ : Pool of routes
3  $S \leftarrow \emptyset$ : temporal solution initially empty
4  $S^*, \mathcal{L} \leftarrow \text{initialSolution}()$  /* See Section 4.4.2 */
5 Set the cost upper bound:  $\bar{f} \leftarrow f(S^*)$ 

/* Iterations */
6 while stopping criterion is met do
    /* Optimize inconsistency and cost objectives, see Section 4.4.3 */
7      $S, \mathcal{L} \leftarrow \text{solveTCDARP}(\text{lexmin } \mathbf{g}, f \leq \bar{f}, S^*, \mathcal{L})$ 
8      $S, \mathcal{L} \leftarrow \text{solveTCDARP}(\min f, \mathbf{g} \leq_{\text{lex}} \mathbf{g}(S), S, \mathcal{L})$ 

    /* Update solution */
9     if  $\mathbf{g}(S) <_{\text{lex}} \mathbf{g}(S^*) \vee f(S) < f(S^*)$  then
10      |  $S^* \leftarrow S$ 
11      | Update ParetoFront with solution  $S^*$ 
12     else
13      | Increase epsilon step:  $\varepsilon \leftarrow \phi \times \varepsilon$ 
14     end

    /* End of one iteration */
15     Update epsilon constraint:  $\bar{f} \leftarrow f(S^*) \times (1 + \varepsilon)$ 
16     Limit the size of  $\mathcal{L}$  to  $N_{\max}$ 
17 end
18 return ParetoFront

```

The second procedure is the optimization of cost f subject to a maximal inconsistency $\mathbf{g}(S)$ found in the previous optimization step (line 8). During these two procedures, \mathcal{L} can be fed with new routes.

If the temporal solution S is strictly better than S^* for at least one of the objectives (i.e. if $f(S) < f(S^*)$ or $\mathbf{g}(S) <_{\text{lex}} \mathbf{g}(S^*)$, line 9) then solution S^* is updated with S (line 10) and solution S^* is added to the Pareto front (if S^* is a non-dominated solution, line 11). Otherwise, the step ε is increased geometrically by ϕ (line 13).

At the end of each iteration, the maximal cost \bar{f} is updated, for the next iteration, based on the cost of S^* (line 15). Pool \mathcal{L} is filled with the new routes at each iteration. Because the size of \mathcal{L} can be too large for regular computer memory, the size is limited to N_{\max} . Routes in \mathcal{L} are first ordered by the consistency-first criteria described in Section 4.4.4 and only the first N_{\max} routes are kept in \mathcal{L} (line 16).

In our case, the fleet size is not constrained, so it is trivial to show that it exists a extreme point in the Pareto front such that every user has only one time-class (i.e. $\mathbf{g} = (0, \dots, |\mathcal{U}|)$).

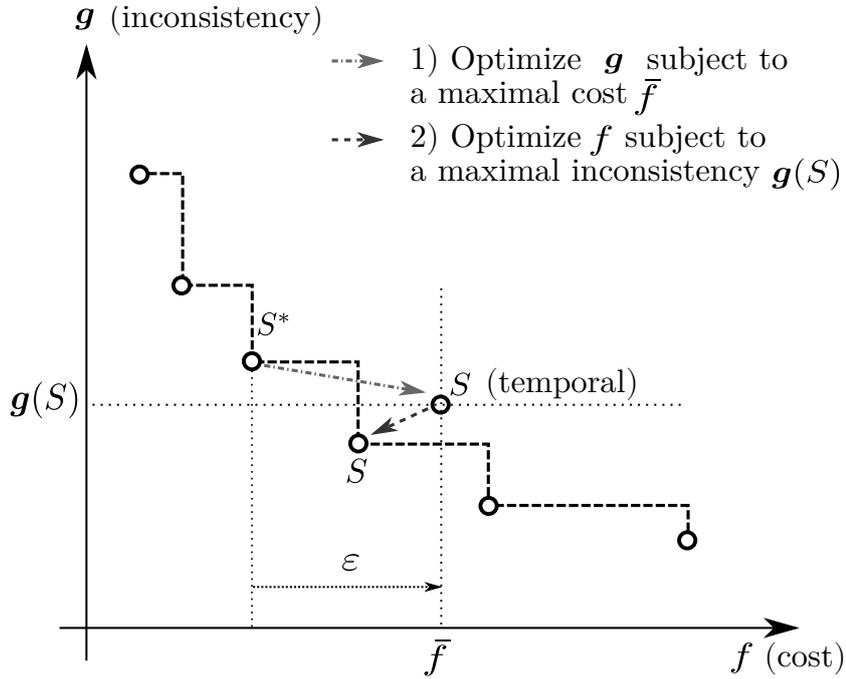


Figure 4.4: Pareto with the two optimization procedures of one iteration (Algorithm 6 lines 7 and 8).

Therefore the stopping criteria (line 6) is met when $\mathbf{g} = (0, \dots, |\mathcal{U}|)$.

4.4.2 Solving the multi-period DARP (MP-DARP)

The set of non-dominated solutions of the TC-DARP defines a Pareto front in which one of the extreme points corresponds to a solution with minimum cost but with a high number of time-classes. This extreme point is found by solving a simplified version of the TC-DARP, called multi-period DARP (MP-DARP) that ignores the consistency requirements. The MP-DARP is modeled as follows:

$$\min f = \lambda v + \sum_{\omega \in \Omega} \sum_{t \in \mathcal{T}} C_{\omega} y_{\omega}^t \quad (4.21)$$

s.t.

$$\sum_{\omega \in \mathcal{R}_u} y_{\omega}^t \geq 1 \quad \forall u \in \mathcal{U}, t \in \mathcal{T}_u \quad (4.22)$$

$$\sum_{r \in \mathcal{R}} y_r^t \leq v \quad \forall t \in \mathcal{T} \quad (4.23)$$

$$y_{\omega}^t \in \{0, 1\} \quad \forall \omega \in \Omega, t \in \mathcal{T} \quad (4.24)$$

$$v \in \mathbb{N} \quad (4.25)$$

The objective function (4.21) which is the same as equation (4.1) represents the transportation cost. Constraints (4.22) are set covering constraints for demand satisfaction. Constraints (4.24) accounts for the number of vehicles v required. As a set covering model, this problem can be easily solved optimally provided the number of routes remains reasonable. Moreover, solutions of the MP-DARP provide feasible routes for the TC-DARP. Hence, the MP-DARP model can be used to build a good initial solution for the TC-DARP.

Algorithm 7 presents the MP-DARP algorithm. A key element of this algorithm is the pool of routes \mathcal{L} . This pool is a set of non-strongly dominated routes defined as follows:

Definition 4.1. Strong route dominance. With ω and ω' two routes, and C_ω and $C_{\omega'}$ the costs of these routes respectively, route ω is said to *strongly dominated* ω' if $C_\omega \leq C_{\omega'}$ and both routes visit the same set of users in the same sequence.

So by definition and with the same notation, route ω' is said to be *non-strongly dominated* by ω if, either both routes visit the same set of users in the same sequence and $C_\omega > C_{\omega'}$, or neither route visits the same set of users in the same sequence.

In this algorithm, the pool of routes \mathcal{L} is initialized with non-strongly dominated routes found during the solution of independent Dial-A-Ride problems, one for each period $t \in \mathcal{T}$ (line 3). Each independent problem minimizes the total transportation cost composed of the distance-related cost and the time-related cost.

The DARP of each period t is solved using the LNS-SCP framework presented in Tellez et al. (2018b). This framework is a large neighborhood search based metaheuristic that quickly yields cost-efficient routes. This LNS-SCP framework uses a record to record procedures with a temporary, current and best solutions. All the routes that were part of the current solution are added to \mathcal{L} .

In the MP-DARP, the inconsistency objective is not considered. So, the visiting times of passengers are not relevant. Thus, this problem can be solved by using a subset of the pool \mathcal{L} considering only non-weakly dominated routes following definition 4.2. This subset is called $D(\mathcal{L})$.

Definition 4.2. Weak route dominance. Let us consider two routes $\omega \in \Omega$ and $\omega' \in \Omega$, with respective costs C_ω and $C_{\omega'}$. Route ω is said to *weakly dominate* ω' if both routes visit the same set of users (in any order) and $C_\omega \leq C_{\omega'}$.

Each iteration of Algorithm 7 consists in four steps: 1) select a subset of non-weakly dominated routes; 2) solve the MP-DARP with the routes selected ; 3) pool management, and 4) update the best solution. This process iterates until $MaxIter^*$ iterations with no improvement of objective value has been performed.

In step 1) a set of routes is selected from \mathcal{L} (line 10). This set is enriched with projection and complementary routes (see Definition 4.3) and added to the restricted pool of routes \mathcal{L}' (line 12). The set of selected routes \mathcal{L}' is obtained by first restricting \mathcal{L} to $D(\mathcal{L})$ and then by applying one of the selection rules that will be described in Section 4.4.4.

Algorithm 7: initialSolution

Data: t_{max} : solver time limit, N_c : number of selected routes, \mathcal{B} : list of selection rules, $MaxIter$: maximal number of iteration without improvement

```

/* Initialization */
1  $\mathcal{L} \leftarrow \emptyset$ : Pool of routes
2 for each period  $t \in \mathcal{T}$  do
3   | Add to  $\mathcal{L}$  all non-strongly dominated routes generated by solving one DARP for
   | period  $t$  with LNS-SCP framework ;
4 end
5  $S \leftarrow \emptyset$ : current solution
6  $S^* \leftarrow \emptyset$ : best solution found
7  $\mathcal{L}' \leftarrow \emptyset$ : restricted pool of routes
8  $itNonImp \leftarrow 0$ : number of iterations without improvement of  $S^*$ 
9 while  $itNonImp < MaxIter^*$  do
   |
   | /* 1. Select routes */
10  | Select a subset  $l \subseteq D(\mathcal{L})$  of  $N_c$  non weakly dominated routes using a selection rule
   |  $o \in \mathcal{B}$ 
11  | Add projection and complementary routes in  $l$ 
12  | Add route selected routes  $l$  to  $\mathcal{L}'$ 
   |
   | /* 2. Solve the MP-DARP */
13  | Create  $S$  by solving a MP-DARP with time limit  $t_{max}$ , pool of routes  $\mathcal{L}'$  and
   | warmstart on  $S^*$ 
14  | if at least one user is visited more than once then
15  |   | Repair  $S$  to get all user visited exactly once
16  |   | Add to  $\mathcal{L}$  all new routes created during the repair
17  | end
   |
   | /* 3. Pool management */
18  | if MP-DARP is not solved to proven optimality then
19  |   |  $\mathcal{L}' \leftarrow \emptyset$ ;
20  | end
   |
   | /* 4. Update best solution and  $itNonImp$  */
21  | if  $f(S) < f(S^*)$  then
22  |   |  $S^* \leftarrow S$ 
23  |   |  $itNonImp \leftarrow 0$ 
24  | else
25  |   |  $itNonImp \leftarrow itNonImp + 1$ 
26  | end
27 end
28 return  $S^*, \mathcal{L}$ 

```

Definition 4.3. Projection and complementarity A route ω' is called a *projection* of a route ω on a period $t \in \mathcal{T}$ if it contains only the users of ω who have a demand at period t , in the same sequence as in ω . The route $\omega'' = \omega \setminus \omega'$ is called the *complementary* route of ω' .

Definition 4.3 is illustrated by Figure 4.5. At period t , Route ω starts from depot D, serves user requests 1, 2 and 3 and returns to the depot. If users 1 and 2 have a transportation request at period t' and user 3 does not, the route ω' serving user requests 1 and 2 is the projection of ω and the route ω'' serving request 3 is the complementary of ω .

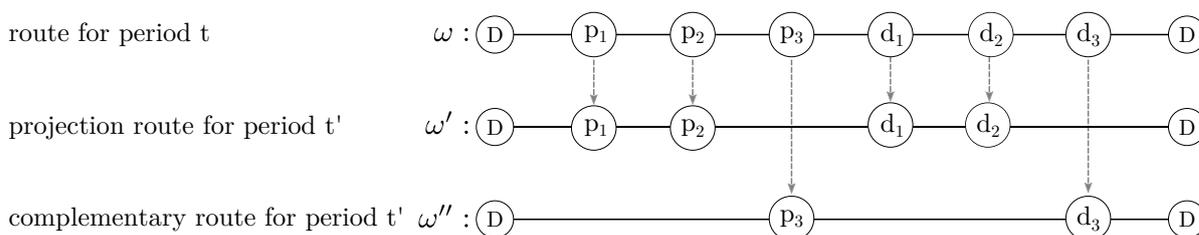


Figure 4.5: Example of projection and complementary routes

In step 2), the MP-DARP instance formed by the pool \mathcal{L}' and S^* is solved by a MILP solver with a time limit t_{max} (line 13). If the best solution S^* is not empty, S^* is used to initialize the MILP solver from a feasible solution. This initialization is called *warm-start*.

Given the presented MP-DARP formulation of is a set covering problem and not a set partitioning problem, solution S may contain some users served by more than two routes. In this case, the solution can be easily repaired by solving the MP-DARP with a new pool of routes \mathcal{L}_{temp} and a warm-start on S (line 15). The new pool of routes \mathcal{L}_{temp} is initialized with the routes of the solution S and enriched as follows. For each user u visited more than once in S and for each route r that visit user u , a new route r' identical to route r but that does not visit user u is added to \mathcal{L}_{temp} . Finally, routes in \mathcal{L}_{temp} are added to \mathcal{L} (line 16).

Step 3) (lines 18-20) performs pool management. The current pool of routes is cleared if the MP-DARP could not be solved optimally. This mechanism aims to keep a manageable pool size and it is inspired by previous works (see e.g. Grangier et al. (2017); Tellez et al. (2018b)) to maintain a manageable pool size.

Finally, the current best solution S^* and the counter of iterations without improvement are updated on Step 4) (lines 21-26). The possibility to use the previous solutions as a warm-start has one main consequence, the value of the objective function can not increase after each iteration. Given this property, when the value of the objective function does not change after several iterations, we suppose the procedure has reached a local optimum. This is why the value of *itNonImp* is used as a stopping criterion indicator.

4.4.3 Mono-objective optimization procedure

This section presents the procedure to solve the TC-DARP for one objective constrained by a maximum value on the other objective. This procedure, detailed in Algorithm 8, is used in Algorithms 6 to solve either the optimization of inconsistency subject to a maximal cost or the optimization of cost subject to a maximal inconsistency.

Algorithm 8 uses the same structure as Algorithm 7. However, here the TC-DARP is solved instead of MP-DARP and a new source of routes denoted \mathcal{L}_{new} , is introduced.

Input parameters for Algorithm 8 are the objective function z , the additional constraints k which define the maximum accepted value for the second objective, the initial solution S_{ini} , and the pool of routes \mathcal{L} . The main variables are the restricted pool of routes \mathcal{L}' (initially empty), the current solution S , and the best found solution S^* that is initialized to S_{ini} (lines 2-6).

As for Algorithm 7, each iteration consists in four steps: 1) select routes; 2) solve an instance the mono-objective TC-DARP; 3) pool management; and 4) update the best solution. This process iterates until $MaxIter$ iterations with no improvement of S^* has reached (see line 7).

The set of selected routes is chosen by using the corresponding policy defined by one source of routes s and one selection rule r . The source s provides a pool of routes from which the selection rule r chooses a subset l . There are two distinct sources of routes, pool \mathcal{L} and a new pool called \mathcal{L}_{new} . If the source is pool \mathcal{L} , we select a subset l of size N from \mathcal{L} using the selection rule r (line 10). Otherwise, the source is \mathcal{L}_{new} which is generated on the fly based on solution S^* (line 12). \mathcal{L}_{new} is particularly designed to be consistent with the current time-classes of S^* . The procedure for generating \mathcal{L}_{new} is detailed in Section 4.5.

Similarly to source \mathcal{L} , we select a subset l from \mathcal{L}_{new} using the selection rule r (line 10). Then, \mathcal{L}_{new} is added to the pool \mathcal{L} (line 14).

We define multiple selection rules. Each selection rule proposes a different way of choosing routes from a source according to a given criterion. A detailed description of each rule is presented in Section 4.4.4.

For diversification purposes, we define several policies combining the two sources and two selection rules into an ordered list \mathcal{O} . Policies are selected at each iteration sequentially, one after the other, based on the following rule: the procedure jumps to the next policy in \mathcal{O} if the solution S^* has not been improved in the last iteration. If the solution has been improved, the same policy is kept for the next iteration. When the last policy in \mathcal{O} is reached, the procedure sets off again with the first policy.

As in Algorithm 7, the first step finishes when the selected routes and their projection and complementary routes are added to the current pool \mathcal{L}' (lines 16 and 17).

In step 2, line 18, the TC-DARP is solved by optimizing the objective z subject to constraints ((4.5)-(4.19) and constraints k that limits the value on the other objective. When the objective is the transportation cost, constraints (4.26) shall also be included to prevent μ_{uc} variables to take arbitrary values and create empty time-classes.

Algorithm 8: solveTCDARP($z, k, S_{ini}, \mathcal{L}$)

```

1 Parameters:  $z$ : objective,  $k$ : additional constraints,  $S_{ini}$ : initial solution,  $\mathcal{L}$ : pool of
  routes.
   Data:  $t_{max}$ : solver time limit,  $N$ : number of routes to append each time in the pool,
    $MaxIter$ : maximum number of iteration without improvement,  $\mathcal{O}$ : list of policies
   Result: best solution found  $S^*$ 
2  $S \leftarrow \emptyset$ : current solution
3  $S^* \leftarrow S_{ini}$ : best solution found
4  $\mathcal{L}' \leftarrow \emptyset$ : restricted pool of routes
5 Initialize policies in  $\mathcal{O}$ 
6  $itNonImp \leftarrow 0$ : number of iterations without improvement of  $S^*$ 
7 while  $itNonImp < MaxIter$  do
   | /* 1. Select routes */
   | Select the corresponding source of routes  $s$  and selection rule  $r$  from list  $\mathcal{O}$ 
   | if the source  $s$  is  $\mathcal{L}$  then
   | | Select a subset  $l \subseteq \mathcal{L}$  of  $N$  routes using  $r$  /* see Section 4.4.4 */
   | | else
   | | | Get new routes solving a DARP with multiple time windows:
   | | |  $\mathcal{L}_{new} \leftarrow solveDARPMTW(S^*)$  /* See Section 4.5 */
   | | | Select a subset  $l \subseteq \mathcal{L}_{new}$  of  $N$  routes using  $r$  /* see Section 4.4.4 */
   | | | Add new routes in  $\mathcal{L}_{new}$  to  $\mathcal{L}$ 
   | | end
   | | Add projection and complementary routes in  $l$ 
   | | Update pool  $\mathcal{L}'$  with routes from  $l$ 
   | | /* 2. Solve an instance of the mono-objective TC-DARP */
   | | Create  $S$  by solving a TC-DARP with time limit  $t_{max}$ , objective  $z$ , additional
   | | constraints  $k$ , restricted pool  $\mathcal{L}'$  and warmstart  $S^*$ 
   | | /* 3. Pool management */
   | | if TC-DARP is not solved to proven optimality then
   | | |  $\mathcal{L} = \{\}$ 
   | | end
   | | /* 4. Update best solution and  $itNonImp$  */
   | | if  $z(S) < z(S^*)$  then
   | | |  $S^* \leftarrow S$ 
   | | |  $itNonImp \leftarrow 0$ 
   | | | else
   | | | |  $itNonImp \leftarrow itNonImp + 1$ 
   | | | end
   | | end
8 end
9 return  $S^*, \mathcal{L}$ 

```

$$\sum_{t \in \mathcal{T}_u} z_{uc}^t \geq \mu_{uc} \quad \forall c \in \mathcal{C}, u \in \mathcal{U}, \quad (4.26)$$

The input routes are \mathcal{L}' and the constituent routes of S^* . The MILP solver is initialized with the routes of S^* to ensure to find a solution with at least the same performance as S^* and to increase convergence speed. The computation time for the solver is limited to t_{\max} . When the objective is lexmin \mathbf{g} , the TC-DARP is solved multiple times following the hierarchical order of the objectives $\{m_{|C|-1}, \dots, m_1\}$. First we solve the problem minimizing $m_{|C|}$. The result obtained with this objective must then be considered in the model as a constraint. Then, we have to solve the model minimizing $m_{|C|-1}$, and the result obtained will not affect or reduce the result obtained with the first objective. This process is done for all the objectives defined in \mathbf{g} .

Finally steps 3 and 4 are identical and have the same function that steps 3 and 4 of Algorithm 7 respectively. Step 3 performs pool management and Step 4 the best solution and stopping criteria management.

4.4.4 Selection rules

When the restricted pool \mathcal{L}' becomes too large, the MILP solver cannot improve the given initial solution. Hence, a key element is to select a subset of routes of reasonable size in \mathcal{L} (line 10) or \mathcal{L}_{new} (line 13). We present in this section three performance indicators used to guide route selection and two route selection strategies called Sequential Selection and Random Biased Selection.

Performance indicators used for route selection

To select routes efficiently, they have to be sorted by their expected performances in a weekly solution. So we introduce several performance indicators that are used to evaluate each route $\omega \in \mathcal{L}$:

- F_ω : Solution cost. This is the best cost of the solution of which route ω has been part of. Initially F_ω is set to infinity. Each time a new solution S is found in Algorithm 6, F_ω is updated as follows: $F_\omega = \min\{F_\omega, f(S)\}$.
- G_ω : Solution inconsistency. This is the lowest inconsistency of the solution in which route ω has been part of. Initially G_ω is set to infinity. Each time a new solution S with route ω is found, the performance is updated with the minimum value $G_\omega = \min\{G_\omega, \mathbf{g}(S)\}$.

Two sorting criterion are defined. The first sorting criteria is the cost-first criteria $(F_\omega, \bar{C}_\omega^t, C_\omega)$ that lexicographically sort the routes in ascending order, first by the solution cost F_ω , then by the normalized sub-problem cost \bar{C}_ω^t and finally by the route variable cost C_ω . The second sorting criteria is the inconsistency-first criteria $(G_\omega, F_\omega, \bar{C}_\omega^t)$ that lexicographically sort the

routes in ascending order, first by the solution inconsistency G_ω , then by the solution cost F_ω , and finally by the normalized sub-problem cost \bar{C}_ω^t . This last criteria is used to limit the size the \mathcal{L} in Algorithm 6.

Sequential selection (SS)

The Sequential Selection (SS) rule is the first selection rule. The purpose of this rule is to gradually cover the routes inside a pool of routes. This method sequentially chooses routes differently according to a given objective. If the objective is to minimize costs, routes are sorted with cost-first criteria and if the objective is to minimize inconsistency, routes are sorted with inconsistency first criteria. The SS rule starts by sorting routes of the pool.

From the sorted list, routes are processed one by one to finally select N routes. For each route, we check whether it was selected in a previous call or not. If it was selected before, it can be selected again with a probability of γ . Otherwise, it is always selected. The memorizing of these selected routes is initialized at the beginning of Algorithm 8. Figure 4.6 shows an example of SS with $N = 6$.

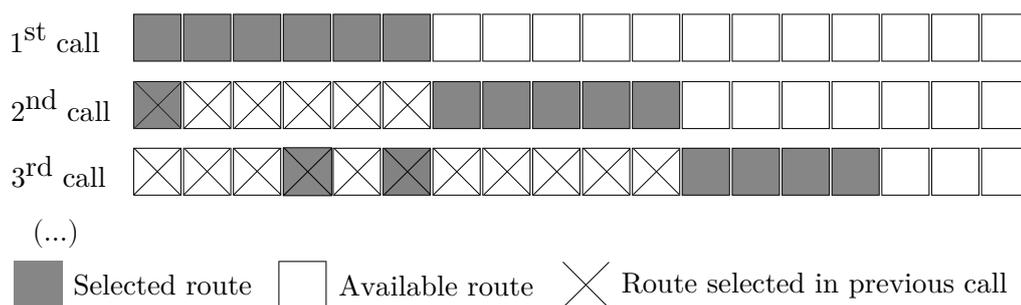


Figure 4.6: Example of 3 successive calls of the sequential selection rule with $N = 6$.

The mechanism used to select routes ensures that routers belonging to the same previous solution (i.e. with the same overall performance, F_ω or G_ω), will be kept together. This mechanism aims to increase the chances to find complete feasible solutions within the routes selected. Nonetheless, this sorting step has to be performed each time the rule is called because performance indicators such as F_ω and G_ω may change from one iteration to another.

Roulette wheel selection (RW)

The roulette wheel is the second selection rule. It is inspired by the roulette wheel mechanism proposed in Pisinger and Ropke (2007). Similar to the SS rule, routes are first sorted according to criteria $(F_\omega, C_\omega^t, C_\omega)$ if the objective is the cost minimization, and criteria $(G_\omega, C_\omega^t, C_\omega)$ if the objective is the inconsistency minimization. Then N routes are chosen from the sorted list l based on their position following the next rule: $\text{position} = \xi^\rho \times |l|$, where $0 \leq \xi < 1$ is a

random number and $\rho \in [1, +\infty[$. Because the list is sorted by decreasing performance, this probabilistic choice assigns a higher probability to the routes with higher performance.

4.5 Route generation: DARP with multiple time windows

The MSPP framework is initialized with routes that have been generated by solving independent DARPs. These routes are combined to form more consistent solutions. Nevertheless, time-consistency can be improved by generating routes that are dominated in terms of cost (i.e. routes which cost can be improved by modifying the sequence of visits).

These dominated routes, used in Algorithm 8 (line 12), may never be generated by the LNS operators used to solve the DARP. This section describes the procedure used to generate new routes that are likely to improve time-consistency. The proposed approach was inspired by Feillet et al. (2014).

Section 4.5.1 presents the main process for the generation of a new set of routes denoted \mathcal{L}_{new} . Section 4.5.2 presents how time windows are defined.

4.5.1 Main process for the generation of new routes

The process for generating new routes consistent with the current solution works as follows:

1. One user \bar{u} is selected randomly among the set of users that have the maximum number of time-classes C_{\max} .
2. For each period $t \in \mathcal{T}$, new routes are generated as follows:
 - (i) A set \mathcal{TW}_u^t of multiple time windows are defined for each user $u \in \mathcal{U}$ according to their service times: if $u = \bar{u}$, multiple times windows are defined to decrease its current number of time-classes, while for the rest of users ($u \neq \bar{u}$) the number of time-classes can be maintained (Section 4.5.2).
 - (ii) Routes are generated by solving an auxiliary Dial-A-Ride Problem with Multiple Time Windows (DARPMTW) for the period t . The routes generated during this process are gathered into a sub-pool of routes denoted \mathcal{L}_{new}^t (Section 4.5.3).
3. The sub-pools \mathcal{L}_{new}^t are gathered into a single pool denoted $\mathcal{L}_{new} = \cup_{t \in \mathcal{T}} \mathcal{L}_{new}^t$ and returned to MSPP (line 12, Algorithm 8).

In Feillet et al. (2014), the TC-VRP does not have time windows. The auxiliary problem solved is a vehicle routing problem with multiple time windows and no waiting time (VRPMTW-nw). The multiple time windows help to define new routes that can decrease the number of time-classes for user \bar{u} . Waiting times are forbidden since the insertion of unnecessary waiting times could artificially improve consistency.

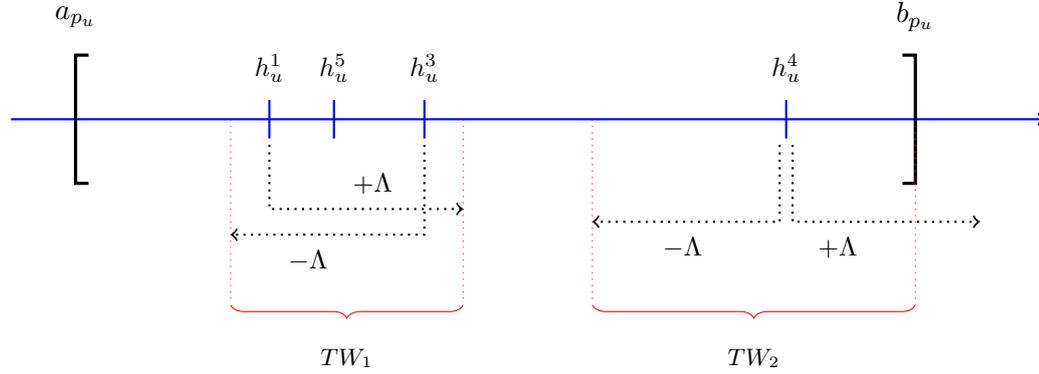


Figure 4.7: Construction of $\mathcal{W}(\mathcal{H}, u) = \{TW_1, TW_2\}$ for user u with pickup time windows $[a_{p_u}, b_{p_u}]$ and $\mathcal{H} = \{h_u^1, h_u^5, h_u^3, h_u^4\}$.

In our case, the initial problem integrates time windows. The sub-problem is a Dial-A-Ride Problem with Multiple Time Windows and minimal route duration (DARPMTW). Routes may integrate waiting times due to time windows but waiting times that artificially improve time-consistency are still forbidden.

4.5.2 Definition of time windows for the DARPMTW

Let \mathcal{TW}_u^t be the set of multiple time windows to be defined for each user $u \in \mathcal{U}$ in the DARPMTW solved to generate new routes for the period t . If $u = \bar{u}$, \mathcal{TW}_u^t is defined in order to decrease its current number of time-classes, while for the rest of users ($u \neq \bar{u}$) \mathcal{TW}_u^t is defined such that the number of time-classes can be maintained.

Let us recall some notation: we consider a user $u \in \mathcal{U}$ with pickup time window $[a_{p_u}, b_{p_u}]$, the subset of time periods with transportation demands $\mathcal{T}_u \subseteq \mathcal{T}$ and variables h_u^t representing the pickup service time of user u at period $t \in \mathcal{T}_u$. We denote by $\mu(\mathcal{H})$ the number of time-classes for the set of service times \mathcal{H} . So $\mu(\mathcal{H}_u)$ is the number of time-classes of user u . The service times are allocated to time-classes as in Feillet et al. (2014).

In the formal definition of \mathcal{TW}_u^t , we use a function $\mathcal{W}(\mathcal{H}, u)$. This function returns a set of time windows within $[a_{p_u}, b_{p_u}]$. Any service time inserted in these time windows will not create a new class. This is illustrated on Figure 4.7. The detailed construction process is given in Appendix B.0.1.

For each user $u \in \mathcal{U}$, the answers to the following two *yes/no* questions determine how the multiple time windows \mathcal{TW}_u^t are defined:

Q1 Is u the selected user \bar{u} ?

Q2 Is the number of time-classes of user u decreased by 1 if the service time h_u^t at period t is removed? This question can be answered by checking if the inequality $|\mu(\mathcal{H}_u \setminus \{h_u^t\})| < |\mu(\mathcal{H}_u)|$ holds.

The answers to questions **Q1** and **Q2** yield four ways to define multiple time windows:

Yes/Yes Changing the service time h_u^t to any service time in the time windows \mathcal{TW}_u^t has to decrease the number of time-classes for this user. So $\mathcal{TW}_u^t = \mathcal{TW}(\mathcal{H}_u \setminus \{h_u^t\}, \bar{u})$.

Yes/No Let $[e_u^t, l_u^t] \in \mathcal{TW}(\mathcal{H}_u, \bar{u})$ be the time window satisfied by service time h_u^t ($e_u^t \leq h_u^t \leq l_u^t$). Defining $\mathcal{TW}_u^t = \mathcal{TW}(\mathcal{H}_u, \bar{u}) \setminus \{[e_u^t, l_u^t]\}$ will enforce a decrease in the number of time-classes for this user.

No/Yes In this case, the service time h_u^t defines a time-class on its own, so $\mathcal{TW}_u^t = \{[a_{p_u}, b_{p_u}]\}$. As a result, any feasible service time at period t can be accepted.

No/No We distinguish two possibilities. In the first one, \mathcal{TW}_u^t is defined similarly as in case Yes/No. This possibility is selected with a probability $\nu = \theta \times (C_{\max} - \mu(\mathcal{H}_u))$, where C_{\max} is the maximum number of time-classes in the current solution, and θ is a fixed parameter. In the second one (selected with probability $1 - \nu$), time windows \mathcal{TW}_u^t are set as in Case No/Yes. This may generate routes that increase the number of time-classes of this user. However, as in Feillet et al. (2014), we observe that this temporary relaxation helps decrease the number of time-classes of user \bar{u} .

These four cases are summarized in Table 4.4.

		Q2: $\mu(\mathcal{H}_u \setminus \{h_u^t\}) < \mu(\mathcal{H}_u)$?	
		Yes	No
Q1: $u = \bar{u}$?	Yes	$\mathcal{TW}_u^t = \mathcal{TW}(\mathcal{H}_u \setminus \{h_u^t\}, \bar{u})$	$\mathcal{TW}_u^t = \mathcal{TW}(\mathcal{H}_u, \bar{u}) \setminus \{[e_u^t, l_u^t]\}$
	No	$\mathcal{TW}_u^t = \{[a_{p_u}, b_{p_u}]\}$	With probability ν : $\mathcal{TW}_u^t = \mathcal{TW}(\mathcal{H}_u, u) \setminus \{[e_u^t, l_u^t]\}$ Otherwise: $\mathcal{TW}_u^t = \{[a_{p_u}, b_{p_u}]\}$

Table 4.4: Overview of cases for the definition of \mathcal{TW}_u^t .

4.5.3 Solving the DARPmTW

The DARPmTW is solved by adapting the LNS-SCP algorithm proposed by Tellez et al. (2018b). The LNS-SCP algorithm has been designed to solve a single period DARPs with a single time window at each pickup or delivery point. We extended it to handle multiple time windows. Additionally, routes should not include unnecessary waiting times that artificially improve time consistency at the cost of increasing waiting times. To the best of our knowledge, the DARPmTW has not been treated in the literature.

The LNS-SCP algorithm is a Large Neighborhood Search (LNS) procedure in which a Set Covering Problem (SCP) is periodically solved to improve the current solution. As the SCP component cannot handle multiple time windows, we do not consider it here. Each iteration of the LNS procedure consists in partially destroying and repairing the current solution through dedicated heuristics. The repairing heuristic must build new feasible routes, and thus satisfy a set of timing and capacity constraints. The introduction of multiple time windows led us to modify the scheduling algorithm which embeds the time-feasibility check. We have developed an extension to the classic scheduling algorithm proposed in Tang et al. (2010) that tests whether the multiple time windows \mathcal{TW}_u^t are satisfied once all other constraints are satisfied.

Solving the multiple time-windows scheduling problem

The scheduling procedure aims to determine if there is a schedule that satisfies timing constraints at each node of a given route. Besides the set of multiple time windows \mathcal{TW}_u^t , each user in the route has the original constraints defined in Section 4.3. That is a maximum ride time \bar{T}_u , a pickup time window $[a_{p_u}, b_{p_u}]$ and a delivery time windows $[a_{d_u}, b_{d_u}]$.

The scheduling algorithm consists of two sequential phases. In the first phase, it solves a classing DARP scheduling problem considering only the original constraints. This problem has been studied by Tang et al. (2010); Gschwind, Timo (2015) and Tellez et al. (2018b). We use the scheduling algorithm presented in Tellez et al. (2018b). Given a route $\omega \in \Omega$, this algorithm checks time feasibility and, if the route is feasible, computes service times following the next lexicographic criterion: 1) minimize route duration, 2) schedule service times as early as possible. At the same time, a maximum forward time shift Δ_ω^+ is computed. This value indicates how much the route schedule can be shifted forward while preserving both its feasibility and the route duration. The detailed procedure can be found in Appendix B.0.4.

Given the scheduling obtained in Phase I, Phase II checks the satisfaction of multiple time windows without increasing the duration of the route. Phase II is described by Algorithm 9. Note, the satisfaction is verified at pickup nodes because the time consistency is only defined at pickups times.

Each iteration of Algorithm 9 corresponds to the feasibility check of the service time at the current node. Nodes are considered sequentially following the route sequence. This algorithm visits all nodes and chooses a feasible service in a time window for each pickup node or detects infeasibility. The only lever to modify service times is the maximum time shift of the route: service times can be delayed by at most Δ_ω^+ units without violating any timing constraint nor increasing the route duration. Each time the schedule has to be shifted forward to ensure feasibility, the value of Δ_ω^+ is decremented by the value of the corresponding time shift.

In lines 5-20, the algorithm checks time windows \mathcal{TW}_u^t until finding a feasible one (line 5). Three cases are considered: (i) If service time takes place before the opening of the time window ($w_{n_i} < a$), the route is shifted forward by the quantity $w_{n_i} - a$, so that the new service time is

Algorithm 9: Phase II: Scheduling extension for solving the DARPmTW

Input: ω : Route with minimal duration and nodes from n_0, \dots, n_N being the first and last node the depot, \mathcal{TW}_u^t : Set of multiple time windows sorted in ascending order, w_i : service time at node n_i , Δ_ω^+ : maximum forward time shift of the route.

Output: if route ω is **feasible** or **unfeasible**

```

1  $i \leftarrow 1$ 
2 while  $i < N$  do
3   if  $n_i \in \mathcal{P}$  then
4      $u \leftarrow$  user with pickup  $n_i$            /* Implies  $n_i = p_u$  */
5     for each time window of the form  $[a, b]$  in  $\mathcal{TW}_u^t$  do           /* time windows */
6       if  $w_{n_i} < a$  then
7          $\delta = a - w_{n_i}$ 
8         if  $\delta < \Delta_\omega^+$  then           /* feasible forward time shift */
9           for each node  $j$  in  $\omega$  do
10             $w_j = w_j + \delta$ 
11             $\Delta_\omega^+ \leftarrow \Delta_\omega^+ - \delta$ 
12             $i \leftarrow 1$ 
13            go to line 3           /* feasible pickup */
14          else
15            return unfeasible
16          else if  $w_{n_i} \leq b$  then
17             $i \leftarrow i + 1$ 
18            go to line 3           /* feasible pickup */
19          else
20             $\mathcal{TW}_u^t \leftarrow \mathcal{TW}_u^t \setminus \{[a, b]\}$ 
21          return unfeasible
22     $i \leftarrow i + 1$ 
23 return feasible

```

exactly a . After shifting the route, the algorithm starts over the scheduling procedure with the first node because this shift can provoke time windows violation at another node (line 9-12). If the value $w_{n_i} - a$ is larger than Δ_ω^+ , then the route schedule is unfeasible (lines 14-15). (ii) If the service time at current node takes place in time window $[a, b]$, then the next node is checked (lines 16-18).

(ii) If the service time takes place after the end of the time window ($n_i > b$), the time window $[a, b]$ cannot be used. it is removed from the set \mathcal{TW}_u^t (lines 19-20) and the next time window is checked. If there is no more time window available in \mathcal{TW}_u^t , infeasibility is detected.

4.6 Computational experiments

The matheuristic described in Section 4.4.1 was coded in C++ and the mathematical models were solved with CPLEX Concert Technology 12.6 running on a single thread on an Intel Xeon E5-1620 v3 @3.5Ghz processor.

This section details computational experiments on these two families of instances. It is structured as follows: Section 4.6.1 presents the value of parameters used by our algorithms. Section 4.6.2 introduces the instances used to evaluate our approach. They are built from real data provided by the GIHP Company. Section 4.6.3 evaluates the main components of the matheuristic approach. In Section 4.6.4, the MSPP is assessed on benchmark instances (Groër et al. (2009), Feillet et al. (2014)) for the time-consistent VRP. Finally, Section 4.6.5 presents managerial insights regarding cost performances and time consistency.

4.6.1 Parameters settings

After preliminary tests on a representative subset of instances, parameters shown in Table 4.5 were found to provide the best average performance.

<i>Global parameters</i>	
$\varepsilon = 0.01$	initial value of epsilon
$\phi = 1.5$	epsilon increase factor
$It_{\max} = 4$	maximum number of iterations without improvement
$N = 100 \times \lceil \mathcal{U} /100 \rceil$	number of routes to be appended to the pool at each call of the TC-DARP
$N_c = N \times \mathcal{T} $	number of routes to be appended to the pool at each call of the MP-DARP
$t_{\max} = 60s$	MILP solver time limit
$N_{\max} = 50000$	size of the pool \mathcal{L}
$N_{new} = 5000$	maximum number of routes in \mathcal{L}_{new}
<i>Parameters route policies</i>	
$\theta = 1\%$	probability of relaxing non-target users in the DARPmTW
$\gamma = 10\%$	percentage of routes that can be re-selected in the sequential selection rule
$\rho = 6$	roulette wheel parameter

Table 4.5: Parameters MSPP (all defined in Sections 4.3-4.5).

The value of parameter ε has a strong impact on the computing time. A higher value of ε enables to decrease the computation time, however, the quality of the Pareto front approximation is considerably deteriorated. Thus, $\varepsilon = 0.01$ seems to be a good trade-off between computing time and the quality of the solution. The MSPP is less sensitive to parameter ϕ but its value needs to be greater than 1.5 to have a significant impact on the value of ε .

We found that a It_{\max} greater than 4 does not improve the quality of each point of the Pareto front. Parameters N and t_{\max} were determined to maximize the number of times the MILP solver can solve the TC-DARPs to proven optimality. In the same way, the parameter N_c was determined to solve the MP-DARP. Note that parameter N_c is much bigger than N

because the MP-DARP contains less binary variables (so is an easier problem to solve) than the TC-DARP.

To keep the number of routes in memory under control, limits in the maximum size of the pool \mathcal{L} and of the \mathcal{L}_{new} were set to 50000 routes and 5000 routes, respectively.

The choice of selection rules for the MP-DARP and route policies for the TC-DARP are the followings:

- Selection rules MP-DARP: $\mathcal{B} = \{SS\}$
- Route policies TC-DARP: $\mathcal{O} = (\{\mathcal{L}, SS\}, \{\mathcal{L}_{new}, SS\})$

SS stands for the sequential selection rule and \mathcal{L}_{new} the pool generated through the DARP with the multiple time windows (see Section 4.4.2).

4.6.2 Description of instances

The time-consistent DARP studied in this chapter arose in the context of transportation of people with disabilities. We collected real data from the GIHP Company based in Lyon, France. This data concerns the transportation of hundreds of persons (users) to MSIs. We decomposed this data according to geographical areas and built 8 small instances with 60 to 80 users, 4 medium-size instances with 120 to 160 users, and 2 large instances with 280 to 295 users. We assume an infinite homogeneous fleet; each vehicle has a capacity of 4 seats and 3 wheelchair spaces. Vehicles cost are composed of an hourly cost $\alpha = 23.8\text{€}$ and a cost per kilometer $\tau = 0.17\text{€}$. Additionally, we artificially set an arbitrarily small vehicle fixed cost of $\lambda = 1$ to favor solutions with the same variable cost but fewer vehicles.

Travel times and distances are obtained from the Open Source Routing Machine² (OSRM) proposed by Luxen and Vetter (2011). For each user $u \in \mathcal{U}$, we defined maximum ride times according to direct travel time t_{p_u, d_u} between the pickup location p_u and the delivery location d_u . The following formula generates maximum ride times (RT) that are between 15 and 30 minutes larger than direct travel times: $RT = 15 \times \lceil (t_{p_u, d_u} + 15) / 15 \rceil$.

Time windows at MSIs are 15 minutes wide. The size of time-classes is 10 minutes wide. Finally, time windows at pickup locations and service times strongly influence the actual design of routes, but their impact is independent of the efficiency of our solution method. Thus, we ignored them for the sake of simplicity.

4.6.3 Evaluation of the metaheuristic components

In this section, we present the tests that we used to evaluate the main components of the proposed metaheuristic MSPP. We compare several settings of the algorithm on a representative sample of five instances of different sizes. Five runs are performed for each instance and setting.

²<http://project-osrm.org/>

Comparison based on time-classes

In this test we compare transportation costs for the cheapest solution found time-class. Although these instances have 5 time periods, users in non-dominated solutions present at most 3 time-classes. Thus, for each run, we report the value of the cheapest solutions found with 3, 2 and 1 time-classes, respectively. Table 4.6 reports the corresponding computational results.

Instance	Best			MSPP-noDARPMTW			MSPP-withRW			MSPP		
	C_{\max}	3	2	1	3	2	1	3	2	1	3	2
C01_60	1291.03	1291.73	1345.41	0.11%	2.15%	19.20%	0.11%	0.58%	5.17%	0.11%	0.32%	2.52%
C02_80	2523.21	2528.83	2561.40	0.01%	0.20%	3.74%	0.01%	0.26%	1.09%	0.01%	0.40%	0.49%
C09_135	2857.18	2863.86	3107.12	0.34%	0.93%	15.46%	0.34%	0.99%	8.69%	0.34%	1.22%	3.74%
C10_160	2621.34	2621.34	2779.06	0.19%	1.04%	19.75%	0.14%	0.78%	10.07%	0.15%	0.70%	5.88%
C12_280	7712.22	7730.04	9487.86	0.26%	0.54%	10.25%	0.26%	0.82%	3.55%	0.27%	1.22%	3.67%
			Avg	0.18%	0.97%	13.68%	0.17%	0.69%	5.72%	0.17%	0.77%	3.26%

Table 4.6: Avg Gaps of MSPP variants for solutions with a C_{\max} of 3, 2 and 1 time-classes.

The instance name is reported in Column 1. Columns 2-4 present the value of the objective function of the best solutions found regardless of the run and the setting. The best results are shown in boldface. In the next columns, we report the relative gap between the average cost obtained with the corresponding selection rule and the best solution found (Best). It is computed as $(\text{Avg Cost} - \text{Best}) / \text{Best} \times 100$. Row Avg is the average value of the corresponding column.

Three variants are tested. The variant called MSPP-noDARPMTW that does not consider the route generation solving a DARPMTW. The second variant MSPP-withRW with an additional selection rule named Roulette Wheel. Finally, the MSPP considers the sequential selection rule and the DARPMTW. Detailed routes policies of each variant can be found in Appendix B.0.3.

The first observation of Table 4.6 is that average values (Avg) among variants differ significantly only in solutions with one time-class $C_{\max} = 1$. Comparing MSPP-noDARPMTW to MSPP, we observe that the DARPMTW brings significant improvement in the quality of solutions with 1 time-class. Comparing the MSPP-withRW to MSPP, it can be seen that the based on Roulette Wheel procedure does not drive to a significant improvement in solutions with 3 and 2 time-classes and worsen the performance of single class solutions. According to this experiment, MSPP configuration seems to outperform other variants.

Considering only the cheapest solution of each time-class is not enough to evaluate the quality of Pareto fronts produced for each variant. Moreover, this study examines intermediate solutions between these points. In the next section, we present a deeper study of solutions with 2 time-classes computing the hypervolume of the curve which consider all non-dominant solutions found.

Comparison based on the quality of the Pareto front

The hypervolume indicator was introduced by Zitzler et al. (2002) in order to compare different Pareto fronts. In multi-objective problems, the hypervolume measures the volume between the set of non-dominated solutions and a reference point. In this study, we use nadir point as a reference point which is an artificial point composed by the best value found on each objective. The larger the hypervolume indicator, the better is the front Pareto (Zitzler et al., 2002).

Table 4.7 reports hypervolume results for the instances and variants presented in the previous section.

Instance	Best	Avg Gap hypervolumes		
		MSPP-noDARPMTW	MSPP-withRW	MSPP
C01_60	9979.72	28.93%	3.91%	1.88%
C02_80	2788.82	19.04%	5.61%	4.06%
C09_135	32847.86	20.34%	5.94%	5.36%
C10_160	32969.07	26.48%	4.68%	3.25%
C12_280	253897.10	13.42%	6.49%	6.82%
Avg		21.64%	5.33%	4.28%

Table 4.7: Avg Gap(%) with respect to the best hypervolume (5 runs)

The instance name is reported on Column 1. Column 2 presents the best hypervolume found regardless the run and the setting. The best results are shown in bold face. In the next columns, we report the relative gap between the average value hypervolumes with the corresponding setting and the best hypervolume found (Best). It is computed as $(\text{Avg hypervolume} - \text{Best}) / \text{Best} \times 100$. Row Avg is the average value of the corresponding column.

The first observation of Table 4.7 is that MSPP has the best results in 4 out of 5 instances. Surprisingly the MSPP-withRW provides slightly better results in the biggest instance. Comparing the average values (Avg) of MSPP-noDARPMTW against MSPP, we conclude that DARPMTW generate routes that significantly improve the solution quality (21.6%).

Overall we can confirm with this experiment that MSPP configuration outperforms other variants. As an example, Figure 4.8 shows graphically the difference of 3 Pareto Fronts of the same instance I01_60 using the 3 MSPP variants.

4.6.4 Performance evaluation on benchmarks from the literature

Since the TC-DARP is a new problem, there is no benchmark in the literature. However, to evaluate the performance of the MSPP, we solve reference instances for two other time-consistent routing problems. The first benchmark is an adaptation of the conVRP instances from Groër et al. (2009) called RconVRP. This adaptation, proposed by Feillet et al. (2014), transforms

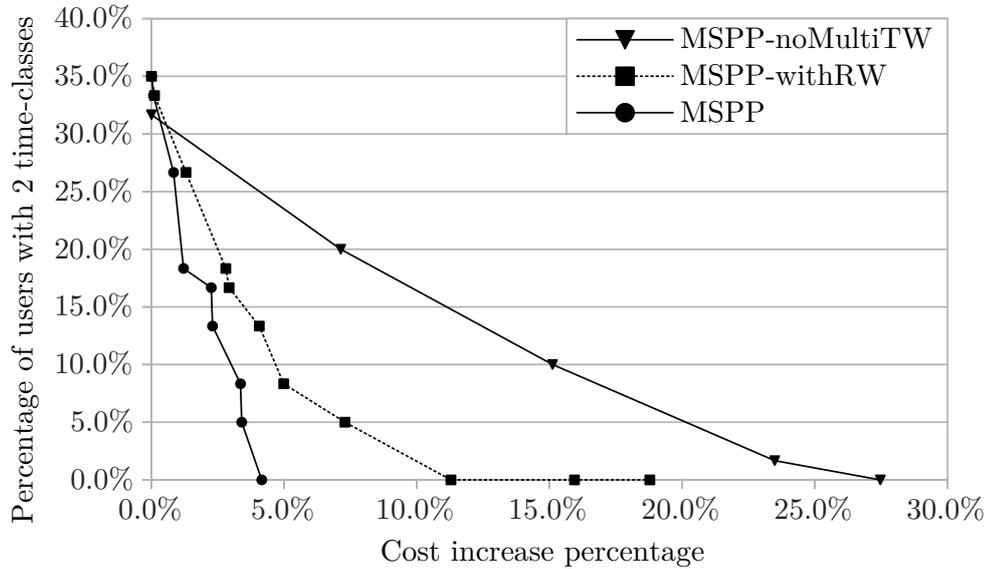


Figure 4.8: Comparison Pareto front instance C01_60

conVRP instances into TC-VRP compatible. RconVRP instances consider up to 12 users over 3 days. Due to the instance size, comparisons against the exact solutions found by a MILP solver are presented. The second benchmark set is the TC-VRP from Feillet et al. (2014). Instances for up to 65 users over 5 time periods are proposed.

Although TC-VRP is the closest problem to the TC-DARP, there are some differences between both problems. The TC-VRP has the following limitations: (i) it has a single depot and no time windows; (ii) it assumes a limited fleet of vehicles; (iii) the objective function is to minimize C_{\max} ; and (iv) routes must start at time 0 and no waiting time is allowed.

Nevertheless, every instance of the TC-VRP can be converted into an instance of the TC-DARP by defining one copy of the depot for each request. Ride times and time windows are relaxed by setting arbitrarily large values. Finally, since the VRP routes are no subject to time windows nor ride times, routes can be traveled in either direction. Thus, each time a route is appended to the pool, the reverse route is also appended.

RconVRP instances by Feillet et al. (2014)

This benchmark proposes 10 small instances of the TC-VRP. The first 5 instances with 10 users and the next 5 with 12 users. Due to the size of the instances, we solved the instances optimally by using CPLEX for an exact comparison. Table 4.8 presents the comparison with the MSPP over 10 runs.

In columns 2-4, optimal solutions (Opt Cost) for each time-class are presented. In the next 3 columns, we report the average performance of the MSPP for each time-class. Note that the MSPP could not find most of the solutions with 3 time-classes and one with 2 time-classes.

Instance	Opt. Cost			Ours (MSPP)			
	C_{max}	3	2	1	≤ 3	≤ 2	1
RconVRP10-1	92.91	92.91	92.91	92.91	92.91	92.91	
RconVRP10-2	80.42	80.42	80.96	80.42	80.42	82.83	
RconVRP10-3	94.12	94.12	94.37	94.12	94.12	94.37	
RconVRP10-4	93.71	93.71	94.09	93.71	93.71	94.09	
RconVRP10-5	83.84	83.84	96.01	83.84	84.50	96.70	
RconVRP10-1	103.65	103.65	104.40	103.65	103.65	109.19	
RconVRP10-2	73.89	73.89	81.25	73.89	73.89	83.40	
RconVRP10-3	82.77	82.77	83.12	82.77	82.77	83.12	
RconVRP10-4	97.57	97.57	101.91	97.57	98.55	104.31	
RconVRP10-5	83.63	83.63	89.25	83.63	83.63	91.38	
Avg Gap (%)				0.0%	0.2%	1.5%	

Table 4.8: Benchmark RconVRP instances from Feillet et al. (2014)

For each instance, the Gap is computed as $(\text{Avg Cost} - \text{Best}) / \text{Best} \times 100$. The last row is the average gap (Avg Gap) overall instances which is the average value of the corresponding column.

For single time-class solutions, the MSPP found 4 out of 10 instances are optimal solutions in the 10 runs. The average gap is 1.5%. The performance is better with 2 time-classes as 7 out of 10 solutions are optimal and an average gap is 0.2%.

Instances TC-VRP

This benchmark was built from real data collected in 14 distinct MSIs, with several users ranging from 15 to 65, and 5 time periods (Monday to Friday). For each MSI, 5 profiles of transportation requests were randomly generated, where the average percentage of users requests varies between 50% to 90% every day. This leads to a total of 70 benchmark instances.

Transportation cost of solutions with 1 to 5 time-classes are provided for most instances. Feillet et al. (2014) solved the TC-VRP with an LNS-based matheuristic with a time limit of 1 hour. However, the MSPP does not stop until all users reach a single class. For each value of C_{max} , our lexicographic optimization explores all non-dominated solutions. This approach is more time consuming but returns a more complete Pareto Front that can help decision-makers to select appropriate solutions.

Tables 4.9 and 4.10 shows the average gap of the MSPP with respect to the LNS method, aggregated in two different ways. For each instance, we compute the gap as $(\text{Cost MSPP} - \text{Cost LNS}) / \text{Cost LNS} \times 100$. Thus, any negative gap represents an improvement.

Table 4.9 shows the numerical results aggregated by the percentage of presence during the week. For example, data-5-Y aggregates instances where, on average, 50% of users are

transported each day, while in the group data-9-Y the average percentage of users transported rises to 90%.

Instance	Avg Gap Transportation cost					
	C_{\max}	≤ 5	≤ 4	≤ 3	≤ 2	1
data5-Y		-1.10%	-1.00%	-1.02%	-0.21%	-0.37%
data6-Y		-1.03%	-1.03%	-0.81%	-0.26%	0.83%
data7-Y		-1.09%	-0.96%	-0.86%	0.13%	-0.13%
data8-Y		-1.06%	-1.00%	-0.72%	0.00%	-0.25%
data9-Y		-0.61%	-0.61%	-0.49%	-0.27%	-1.76%
Avg Gap (%)		-0.98%	-0.92%	-0.78%	-0.12%	-0.33%
Nb Sols		70	70	70	70	70
Nb new BKS		63	59	53	35	35

Table 4.9: Results aggregated by percentage of users requests on Feillet et al. (2014) benchmark

The average relative gap (Avg Gap) overall instances between the results obtained with the MSPP and the LNS was improved for all time-classes. However, slightly better results are reported for solutions with 3, 4 and 5 time-classes. This result is confirmed with the number of new best know solutions (Nb new BKW) which is more than 50 for solutions above 3 time-classes and 35 for solutions with 1 and 2 time-classes. A total number of 245 strictly new best solutions were found, as shown on the last row of the table. Detailed results can be found in Appendix B.1.

Table 4.10 shows the numerical results aggregated by MSI. The last two digits of the instance name represent the number of users. This table shows that MSPP has better performance with larger size instances. However, particularly dataX-59 instances are the most difficult to solve for the MSPP with an over the cost of 3.41% for $C_{\max} = 2$ solutions and 2.39% for $C_{\max} = 1$ solutions.

4.6.5 Managerial insights about time-consistency and transportation costs

This section provides a set of managerial insights regarding the relationship between time-consistency and transportation costs. Figures 4.9-4.11 show the Pareto Front obtained on GIHP instances with 60, 160 and 280 users respectively. Transportation cost is presented as the cost increase percentage with respect to the cheapest solution found on that instance (x-axis). The time consistency of non-strongly dominated solutions is shown in a vector form on the vertical axis. Each element of the vector represents the number of users having 3, 2 and 1 time-classes on that solution, respectively. Note that solutions with 4 or 5 time-classes are not represented in our tests because they have always been dominated by a solution with 3 time-classes.

These Pareto fronts provide decision-makers with a fine intuition of the cost of time-consistency associated with each user. A first observation is that all Pareto Fronts start with

Instance	Avg Gap Transportation cost					
	C_{\max}	≤ 5	≤ 4	≤ 3	≤ 2	1
dataX-15		0.03%	0.03%	0.00%	0.47%	3.04%
dataX-21		-0.15%	0.07%	0.18%	0.24%	1.53%
dataX-25		-0.21%	-0.06%	0.01%	0.19%	0.53%
dataX-26		-0.21%	-0.21%	-0.03%	0.04%	0.08%
dataX-27		-0.27%	-0.12%	-0.16%	0.03%	-0.93%
dataX-32		-0.54%	-0.54%	-0.19%	0.10%	0.49%
dataX-41		-1.15%	-1.15%	-1.65%	-1.19%	-2.46%
dataX-44		-0.75%	-0.75%	-0.61%	-0.72%	-1.19%
dataX-46		-1.05%	-1.05%	-0.62%	-0.18%	-0.34%
dataX-48		-1.24%	-0.98%	-0.77%	-0.20%	-1.38%
dataX-55		-1.91%	-1.91%	-1.63%	-0.92%	-4.84%
dataX-59		-2.17%	-2.12%	-1.62%	3.41%	2.39%
dataX-64		-2.30%	-2.30%	-2.14%	-1.66%	-0.18%
dataX-65		-1.77%	-1.79%	-1.71%	-1.33%	-1.42%
Avg Gap (%)		-0.98%	-0.92%	-0.78%	-0.12%	-0.33%

Table 4.10: Results aggregated by instance size on Feillet et al. (2014) benchmark

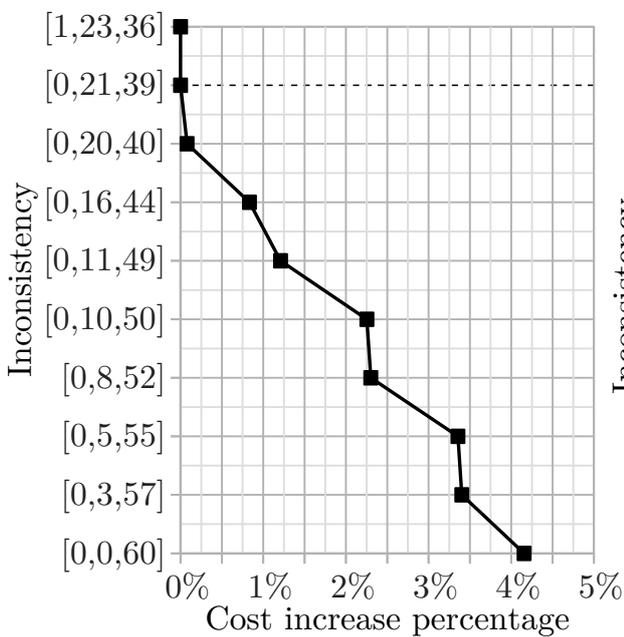


Figure 4.9: Pareto front C01_60

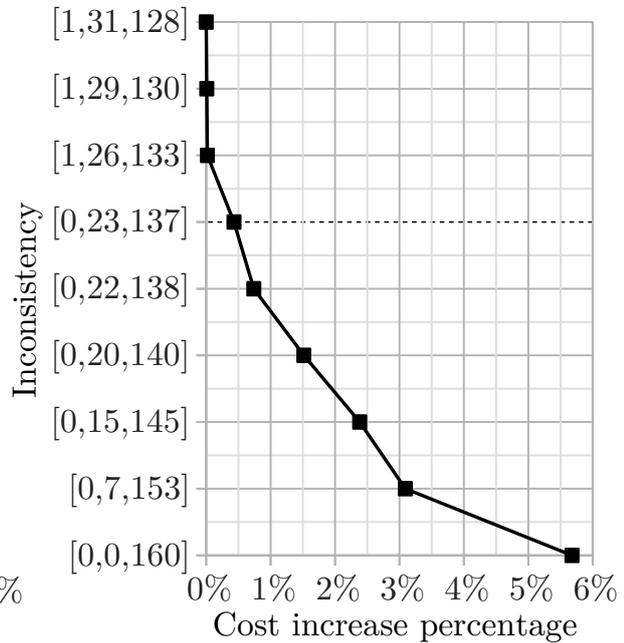


Figure 4.10: Pareto front C10_160

the majority of users having a single time-class and very few users with 3 time-classes. With a minor cost increase percentage, all users have at most 2 time-classes (until the dotted line). This means that a useful consistent solution can be found with respect to the cost of the cheapest

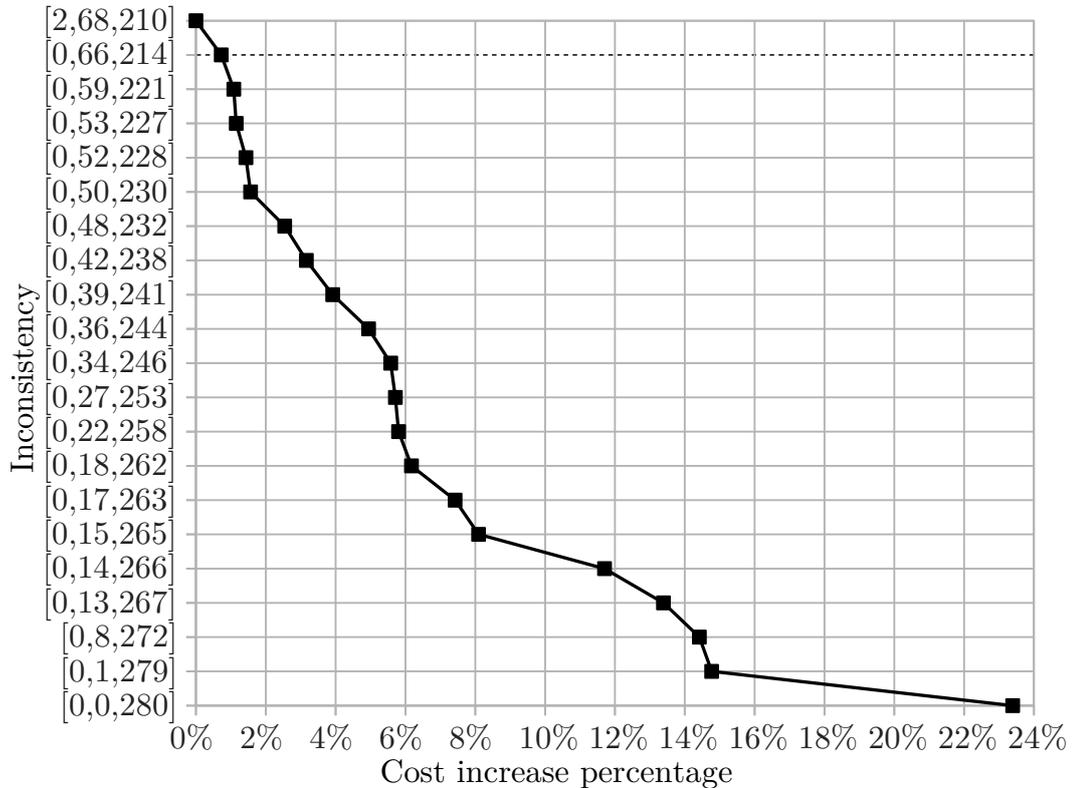


Figure 4.11: Pareto front C12_280

solution, and a small cost increase percentage can significantly improve the situation of highly inconsistent users.

Depending on the instance size, the cost increase percentage for reaching single class solutions can vary from 4% in C01_60 to 23% in the C12_280 instance. Note that values in the y-axis are ordered but non-scaled as the distance between points is always constant. To draw a scaled and readable Pareto Front, we reduce the scope of solutions and focus on a single time-class. As an example, Figure 4.12 shows the instances scaled for solutions with a 2 time-classes. The y-axis presents the percentage of users in that instance with having 2 time-classes.

4.6.6 Economic impact route time shift

In this section, we compute the impact of flexible departure time of routes on time-consistency. This effect has been studied by Kovacs et al. (2014b) for the conVRP, showing that departure time flexibility provides considerable improvement in the solution quality under tight consistency requirements. As far as the TC-DARP is concerned, the departure flexibility is limited by time-windows and maximum ride-time constraints. Routes departure can be scheduled at any time between its earliest and its latest departure date. We define the *maximum time shift* as the difference between these two schedules. The maximum time shift of route ω is denoted Δ_{ω}^{+} .

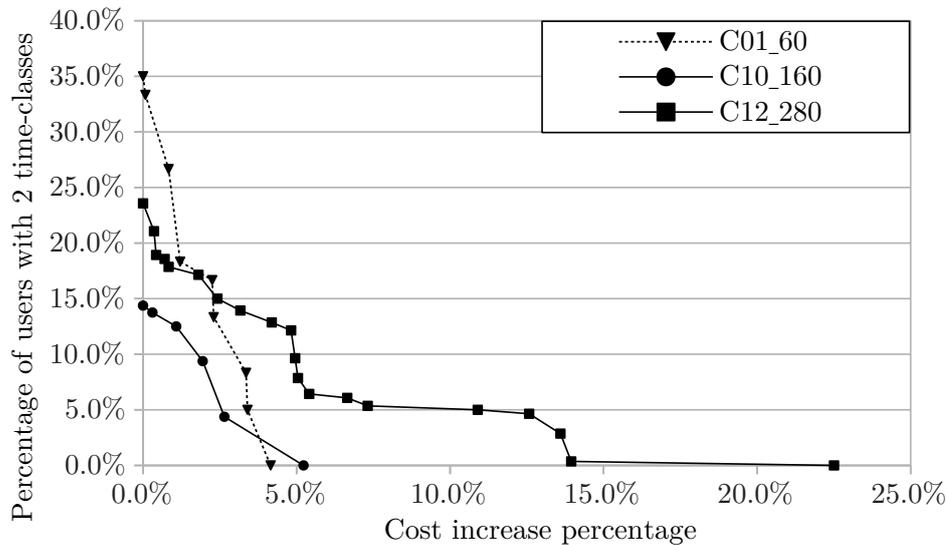


Figure 4.12: Scaled Pareto fronts

Table 4.11 measures the impact of the departure time shift on the complete set of GIHP instances. Columns 2–4 present the best results on 5 runs of the MSPP when the route time shift is allowed, for values of C_{max} decreasing from 3 to 1. Missing values in column 2 mean that all solutions found with $C_{max} = 3$ were dominated by another solution with $C_{max} = 2$. Column 5 (%R-Shift), shows the percentage of routes in which a departure time shift is implemented. Columns 6–8 present the minimum gap on 5 runs with respect to the minimum cost found by the MSPP when no time shift is allowed ($\Delta_{\omega}^+ = 0$). The last row shows the average values.

The last row shows the minimum gap (Gap Min) on 5 runs with respect to the minimum cost found for both variants. This row shows that flexible departure time reduces the transportation costs for highly consistent solutions. The average gain is found on single-class solutions with 1.44% ($= 1.68 - 0.24$) saving and decreases with increment of C_{max} . The managerial implication is that shifting time departure can improve time-consistency without modifying transportation costs in a DARP context.

4.7 Conclusions

This chapter introduced a new variant of the multi-period DARP denoted the time-consistent DARP. This problem aims to find a balance between two objectives, the transportation cost and the time consistency of users. Transportation costs include routing and vehicle ownership costs. The time consistency objective is expressed as a lexicographic function of the number of users having different time-classes. Regarding literature, this is a new way to express time consistency. It is more time consuming than traditional min-max objectives but returns a more detailed Pareto frontier that helps decision-makers selecting the appropriate solution.

Instance	MSPP				MSPP($\Delta_{\omega}^+ = 0$)				
	C_{\max}	3	2	1	%R-Shift	3	2	1	
C00_80	2045.54	2044.43	2061.89		49%	-0.07%	0.25%	0.35%	
C01_60	1291.03	1291.73	1345.41		40%	0.18%	0.48%	0.46%	
C02_80	2523.21	2533.86	2561.40		31%	0.00%	-0.01%	0.31%	
C03_70		1735.07	1752.43		33%		0.00%	0.30%	
C04_80		1207.72	1220.91		32%		0.00%	2.44%	
C05_80	1871.25	1871.53	1923.21		40%	0.37%	0.06%	0.08%	
C06_60		3304.41	3332.21		42%		0.00%	0.54%	
C07_65	1865.46	1868.15	1920.95		50%	0.00%	-0.07%	3.22%	
C08_12		5500.45	5534.44		39%		0.02%	1.76%	
C09_135	2857.18	2880.57	3107.12		38%	0.27%	-0.18%	1.17%	
C10_160	2621.35	2621.35	2779.06		42%	0.03%	0.00%	8.68%	
C11_160	3222.10	3233.17	3549.53		42%	0.21%	0.00%	4.84%	
C12_280	7712.44	7740.33	9487.86		33%	0.02%	-0.22%	4.64%	
C13_295	6294.89	6404.49	7664.17		47%	-0.04%	-1.05%	6.10%	
					Avg	40%	0.10%	-0.05%	2.49%

Table 4.11: Economic implications of allowing a later departure of routes in solutions (Best solutions on 5 runs)

Regarding the TC-VRP problems, TC-DARP extends it by considering time windows and maximum ride times in a problem with multiple destinations. This problem was studied in the context of door-to-door transportation of children with disabilities in the Auvergne-Rhône-Alpes, a region of France.

To compute the Pareto Front, we developed a matheuristic framework based on an epsilon constraint procedure and a master set partitioning problem (MSPP). Routes are generated by solving a DARP with multiple time windows and minimal wait time. At each iteration, a subset of routes is chosen to feed the MSPP procedure. Different selection rules are presented to this end. Experiments show the high performance of the MSPP on real-life instances for up to 295 users. MSPP was also tested on literature instances that achieved outstanding results for the TC-VRP benchmark.

Some disabilities are very sensitive to inconsistent schedules. In this study, we show that economic solutions are already fairly consistent with very few passengers having 3 time-classes. Moreover, with a small increase in transportation costs (<1%), passengers' schedules will be at most 2 time-classes each. Finally, we show that allowing a flexible departure of routes improves the transportation costs of highly consistent solutions. For future research, the model can be extended with driver-related constraints such as working time shifts, breaks, and driver-consistency. Regarding vehicles, it can be extended to handle a heterogeneous fleet and en-route reconfigurations.

Case study: Assessment of transport pooling policies

THE GIHP company is one of the biggest transport companies for people with disabilities in the Auvergne-Rhône-Alpes region, France. Everyday GIHP transports more than one thousand people with disabilities to specialized centers called Medico Social Institutions (MSI). Currently, GIHP organizes the transport of each MSI independently. This policy of grouping users has many management and technical advantages, but a major drawback. It is sub-optimal from an economic point of view and the extra cost must be weighed against its benefits. This study reveals the additional cost of the current policy at around 30% compared with the best scenario. Two alternative policies allowing the pooling of transport between several MSIs in different ways are proposed. Policies differ in the flexibility of management rules to construct clusters. Both policies have comparable savings but the policy with higher flexibility has fewer routes. Regarding service quality, we found that the characteristics of user rides were not sensitive to the different policies in the presence of maximum ride time constraints.

Keywords: transport pooling, clustering, proximity measures, dial-a-ride problem, reconfigurable vehicles, heterogeneous fleet

5.1 Context

GIHP Company

The GIHP company is one of the biggest transport companies for people with disabilities in the Auvergne-Rhône-Alpes region in France. Founded in 1998, GIHP is specialized in the transport

for persons with reduced mobility, and on behalf of institutions in the transport of people with physical or mental disabilities. GIHP has the particularity of having a fully associative shareholding. The board of directors is made up of representatives from associations of the medico-social sector, namely: ADAPEI of Rhône, ARIMC, APAJH, OVE Foundation, GIHP Auvergne-Rhône-Alpes, and GIHP National. This particularity is a competitive advantage over other transporters because there is a common interest in improving the quality of service and optimizing costs.

GIHP has around 180 employees and 180 adapted vehicles. Part of the transportation activity is outsourced to about 100 drivers and 175 vehicles. Yearly GIHP provides about 500 000 trips covering 3.5 million kilometers. GIHP is present in a variety of transport services in the medico-social sector for different needs such as:

Transport for Medico-social Institutions (MSI): the transport management for MSIs is the main economic activity of GIHP. This transport is mainly regular during the school year with daily frequencies or twice a week for People With Disabilities (PWD) in boarding houses. GIHP serves a wide range of social care facilities, such as rehabilitation centers for the elderly and MSIs for people with motor, mental or sensory disabilities. In this chapter, we study this transport type.

Public transport: local authorities (i.e. municipalities) set up particular public services for people with reduced mobility to help them live independently. GIHP assures local transports for activities such as work, shopping, and leisure.

Private transport: people with reduced mobility often demand individual transports for their daily life activities. Unlike other types of transportation, this is not financed by the government.

Vehicle rentals: GIHP offers the possibility to rent a vehicle without a driver. There is a wide variety of vehicles that can carry from 1 to 5 people in wheelchairs or up to 9 seats.

The problem

Currently, GIHP organizes the transport for each MSI (Client) independently. Routes, vehicles, and drivers are usually dedicated to each client. There are various advantages of managing MSIs in this way. For example, the optimization of routes for a single MSI is easier than for multiple MSIs using conventional non-automated methods. From a management point of view, contracting and negotiation with each MSI independently are less complex to manage and with a high-bargaining power. However, this clustering policy is certainly sub-optimal from the economic point of view and the additional costs must be weighed against its benefits.

On the contrary transport pooling with MSIs has probably economic advantages but it supposes a simultaneous negotiation process and requires a specialized mechanism for pricing share routes. Figure 5.1 pictures the CURRENT clustering policy for 16 users attending 3 MSIs. Each line represents one user connecting one pickup (node) and one delivery (MSI) point. Each polygon represents an independent cluster.

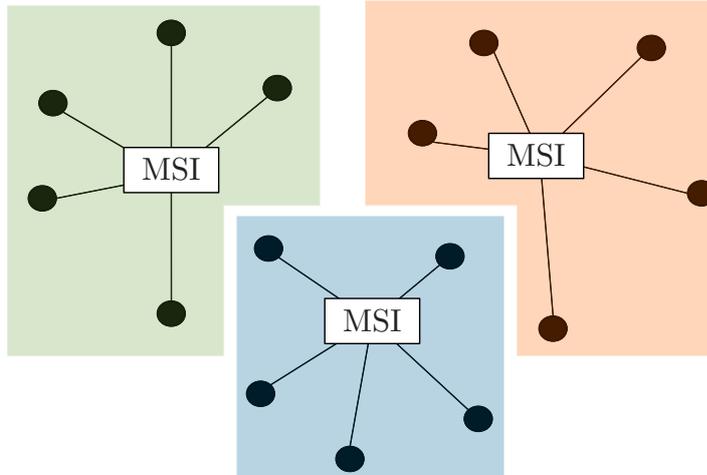


Figure 5.1: CURRENT clustering policy

The scope of the study

There are multiple ways of dividing a big problem into smaller sub-problems. The objective of this study is to evaluate alternative policies of grouping users and compare them with the CURRENT policy from both an economic and a service quality point of view. Alternative policies consider transport pooling among MSIs in different ways.

- The economic indicator is the total transport cost defined as follows:

$$TC = \text{vehicle ownership costs} + \beta \times \text{total traveled time} + \alpha \times \text{total distance covered} \quad (5.1)$$

With α the unitary time cost, β the unitary distance cost.

- The service quality unit is the user ride time which is the time expended in the vehicle from the pickup moment to the delivery time of each at passenger at MSIs.

5.2 The clustering problem

Decomposing a problem into smaller sub-problems is a common approach for dealing with large size routing problems. In the scientific community, this problem is known as *the clustering problem*. It consists of creating groups (clusters) of users according to a clustering policy and to a proximity-based criterion. Clustering policies reflect management practices regarding what is possible or not to group while proximity criteria measures how far(near) are the users.

5.2.1 Proximity measures

The geographic distance is very often the proximity measure between two independent locations (points) in a classic clustering problem. In a Dial-a-Ride system, however, each user $i \in \mathcal{U}$ has one pickup $p(i) \in \mathcal{P}$ and one delivery $d(i) \in \mathcal{D}$ location. Thus, points need to be considered in

couples rather than independently. Additionally, there is a temporal dimension between points that should be considered. Figure 5.2 shows graphically the difference between the proximity of points and users.

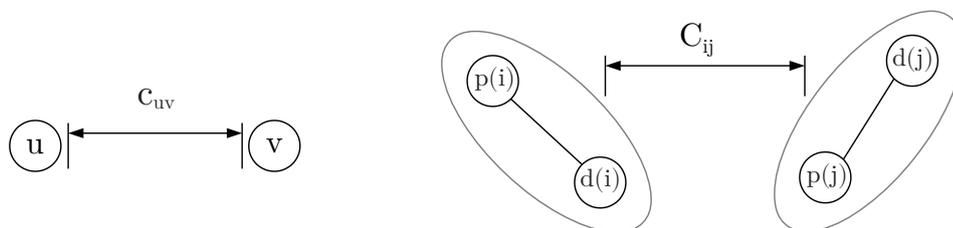


Figure 5.2: Proximity between points (left) and users (right)

Let t_{uv} and d_{uv} be the distance and travel time between points $u, v \in \mathcal{P} \cup \mathcal{D}$. The proximity between u and v is the transport cost c_{uv} defined as follows:

$$c_{uv} = \alpha t_{uv} + \beta d_{uv} \quad (5.2)$$

Each user $i \in \mathcal{U}$ has a maximum ride time \bar{T}_i , and each point (pickup or delivery) has a service duration s , and a time window with the earliest and the latest possible time for the pickup $[e, l]$. There are indicators proposed in the literature to measure the proximity between users in routing problems (Shaw, 1998; Prescott-Gagnon et al., 2009). In the following sections, we present classic measures from the literature.

Pickup relatedness

The pickup relatedness C_{ij}^p between user i and user j is the cheapest transport cost between pickup points.

$$C_{ij}^p = \min\{c_{p(i)p(j)}; c_{p(j)p(i)}\} \quad (5.3)$$

With $c_{p(i)p(j)}$ the transport cost from pickup $p(i)$ to pickup $p(j)$ following equation 5.2.

Delivery relatedness

The delivery relatedness C_{ij}^d between user i and user j is the cheapest transport cost between delivery points.

$$C_{ij}^d = \min\{c_{d(i)d(j)}; c_{d(j)d(i)}\} \quad (5.4)$$

With $c_{d(i)d(j)}$ the transport cost from delivery $d(i)$ to delivery $d(j)$ following equation 5.2.

Relatedness of Shaw (1998)

The relatedness of Shaw (1998) C_{ij}^{p-d} between user i and user j take into account both pickup and delivery locations. This measure is based on the Shaw (1998) distance-related operator

which considers only the distance between points. Given that our objective function is cost-based rather than distance-based, the relatedness is computed as follows:

$$C_{ij}^{p-d} = \frac{c_{p(i)p(j)} + c_{p(i)d(j)} + c_{d(j)p(i)} + c_{d(i)d(j)}}{4} \quad (5.5)$$

With c_{uv} the transport cost from u to v following equation 5.2.

Time relatedness

Prescott-Gagnon et al. (2009) proposes a spatial-temporal measure for the vehicle routing problem with time windows. This measure was adopted by Lehu ed e et al. (2014) for a DARP context. The proposed adaptation considers only the temporal dimension conserving the structure of the measure. The time relatedness C_{ij}^t between user $i \in \mathcal{U}$ and user $j \in \mathcal{U}$ is computed as follows:

$$C_{ij}^t = \frac{\min\{t_{p(i)p(j)}; t_{p(j)p(i)}\}}{2T^{\max}} + \frac{1}{T_{ij} + T_{ji}} \quad (5.6)$$

With T_{ij} the width of the interval of feasible visiting times defined as follows:

$$T_{ij} = \max\{1; \min\{l_{d(j)}, l_{d(i)} + t_{d(i)d(j)}\} - \max\{e_{d(j)}, e_{d(i)} + t_{d(i)d(j)}\}\} \quad (5.7)$$

T^{\max} is a scaling factor set to the largest value of t_{uv} with $u, v \in \mathcal{P} \cup \mathcal{D}$.

5.2.2 Clustering policies

We define two new clustering policies. The MSI clustering policy (C-MSI) consisting of grouping MSIs and the user clustering policy (C-USER) consisting of grouping users.

MSI clustering policy (C-MSI)

The MSI clustering policy (C-MSI) consists of grouping close related MSIs. The particularity of this policy is that each MSI can belong to only one cluster and so there is a single contractualized solution. Figure 5.3 shows a 2-cluster solution following this policy. The cluster on the right side (blue) has 2 MSIs while the cluster on the left side (green) has only one MSI.

The MSI clustering policy can be modeled as a mixed-integer linear program. Let \mathcal{U} be the set of users. C_{ij} is the relatedness between user $i \in \mathcal{U}$ and user $j \in \mathcal{U}$ computed with one proximity measure of Section 5.2.1. $\mathcal{K}(m)$ is the set of users attending MSI $m \in \mathcal{M}$ and $\mathcal{M}(i)$ the MSI of user $i \in \mathcal{U}$. N is the number of clusters. It is assumed in this modeling approach that each cluster has one centroid which can be placed in the same position as any of the users belonging to this cluster. Note that users' positions do not have a physical representation because it is associated with a pair of points.

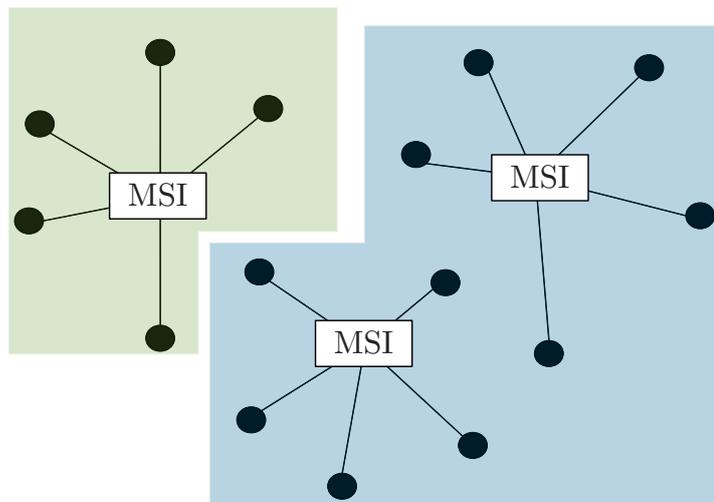


Figure 5.3: Clustering MSIs

Decision variables $x_{ij} \in \{0, 1\}$ are 1 if user i is assigned to cluster whose centroid is j and 0 otherwise. Variable $y_j \in \{0, 1\}$ are 1 when the cluster with centroid in j is chosen. The objective is to find the best assignment of users to clusters such that the sum of distances is minimized.

$$\min f = \left(\sum_{i,j \in \mathcal{U}} C_{ij} x_{ij} \right) \quad (5.8)$$

s.t.

$$\sum_{j \in \mathcal{U}} x_{ij} = 1 \quad \forall i \in \mathcal{U} \quad (5.9)$$

$$\sum_{j \in \mathcal{U}} y_j = N \quad (5.10)$$

$$x_{ij} \leq y_j \quad \forall i, j \in \mathcal{U} \quad (5.11)$$

$$x_{rj} \leq x_{ij} \quad \forall i, j \in \mathcal{U}, r \in \mathcal{K}(\mathcal{M}(i)) \quad (5.12)$$

$$x_{ij}, y_j \in \{0, 1\}, \quad \forall i, j \in \mathcal{U} \quad (5.13)$$

Constraints 5.9 make sure that each user $i \in \mathcal{U}$ is assigned to only one cluster. Constraints 5.10 define the number of clusters needed. Constraints 5.11 force y_j variables to take 1 when at least one user is assigned to the cluster located in j . Constraints 5.12 make sure that users from the same MSI are assigned to the same cluster. Finally, variables definitions are given by constraints 5.13.

Intuitively, this policy is cheaper than the CURRENT policy because MSIs are now allowed to pool transport routes. Thus, vehicles can pickup users for multiple MSIs. However, the tendency of service quality is not predictable because longer routes not necessarily imply longer ride times.

The user clustering policy (C-USER)

The second policy is the user clustering policy (C-USER). It consists of grouping users independently of the notion of MSIs. From a management perspective, this policy supposes that MSIs can share the transport management of a group of users with other MSIs. Additionally, as these users can share the same route there should be a pricing mechanism to bill users individually.

Figure 5.4 shows the example of a 2-cluster solution following this policy. The cluster above (orange) covers users from 2 MSIs and the cluster below (blue) from the 3 MSIs. Although this

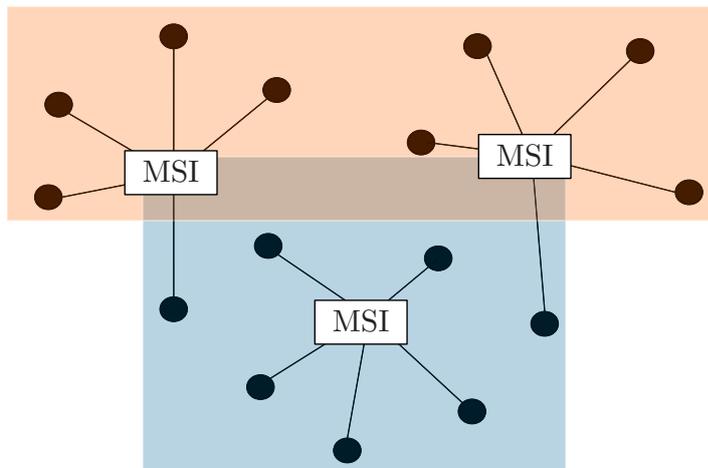


Figure 5.4: Clustering of users (C-USER)

policy imposes some relevant management challenges to MSI, there is an economic advantage that this study aims to quantify. This policy is a relaxation of the clustering problem in which users can be assigned to any cluster without constraint. We expect transport routes to take longer than other policies. However, the quality of service not necessarily worse.

5.3 Solving Approach

The general solving approach consists of 2 stages, clustering first and then, in solving each cluster independently. Figure 5.5 shows the procedure graphically. In the first stage, clusters are determined by solving optimally the mathematical model of Section 5.2. A standard MILP solver is used to this end. In the routing stage, we solve as many DARPs as clusters. The algorithm used to solve each sub-problem is the LNS-SCP of Tellez et al. (2018b) that was presented in cite 3. The GIHP problem is a DARP variant known as the fleet size and mix DARP with reconfigurable vehicle capacity. This problem considers an unlimited fleet of reconfigurable vehicles. Finally, the solution of sub-problems is gathered into one global solution which is the one that we used to compare clustering policies.

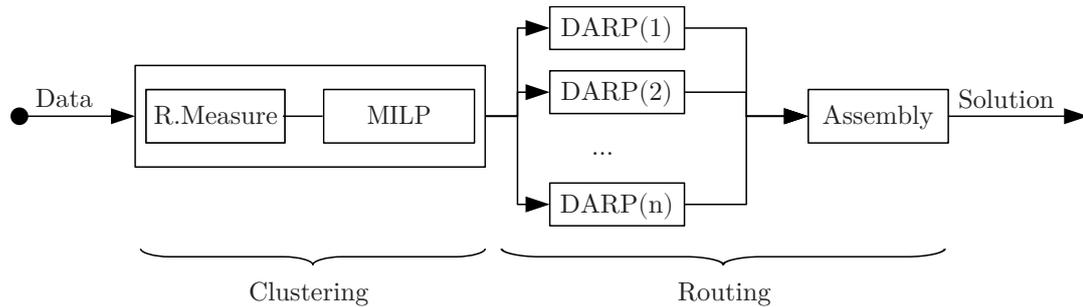


Figure 5.5: Solving approach

5.4 Input data

We consider a sample data set of 562 users attending to 31 MSIs of the Auvergne-Rhône-Alpes region. Transportation requests correspond to the period of Monday morning including users some in boarding schools. As this is a single period problem, the terms of requests and users are equivalent and can be used interchangeably. MSIs can be of 3 different sizes: 14 small with between 1 and 9 users, 12 medium-size with between 10 and 40 users, and 5 big-size with between 41 and 76 users as shown in Table 5.1. Travel times and distances are obtained from the Open Source Routing Machine¹ (OSRM) by Luxen and Vetter (2011). Common characteristics for the instances are:

Data set	Number
Users total	565
MSIs total	33
Small MSIs	16
Medium-size MSIs	12
Big MSIs	5

Table 5.1: Global characteristics of the test data set

Ride times

One of the main measures of service quality is the users' riding time. To avoid long times values a maximum ride time constraint is usually imposed on each user. This constraint should be proportional to the travel time following a direct way. For GIHP there are several constraints to limit the travel time, but their method can not be generalized for any transport time. Thus, we propose the following step-wise function proposing acceptable limits for users: $maxRideTime = 15 \times \lceil (t_{p_r, d_r} + 15) / 15 \rceil$. This function adds up 15 minutes to the direct ride time rounded to the next multiple of 15 as shown in Figure 5.6.

¹<http://project-osrm.org/>

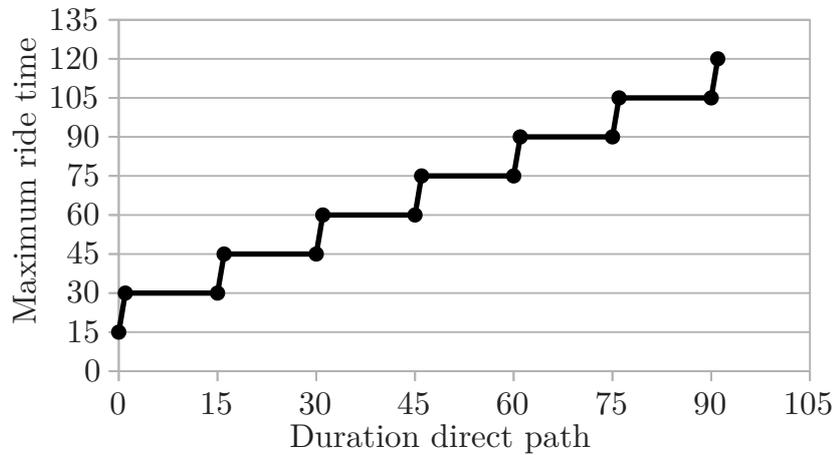


Figure 5.6: Maximum ride time function

Time windows and service times

The most common time window is at MSIs. Vehicles should arrive between 15 minutes of the MSI opening time. The service duration for users using seats is 2 minutes for pickup and 1 minute for delivery operations. For users in a wheelchair, it is 5 minutes at pickup locations and 2 minutes at delivery locations.

Vehicles

GIHP has 22 vehicle types with different configurations for people in wheelchairs (W) and seats (S). Each vehicle has fixed ownership cost (C.FIXED), and a variable distance cost (α). There is a driving cost (β) which is dependent on the vehicle type. Big vehicles are more expensive because drivers require a special license. Table 5.2 presents the list of vehicles and their configurations. Vehicles are sorted for the lowest to the highest fixed cost.

5.5 Computational experiments

The experiments were run on a CPU Intel Xeon E5-1620 v3 @3.5Ghz. The clustering problem was solved using CPLEX 12.6. The routing problem was solved using the LNS-SCP algorithm with the parameters of Appendix C.0.1. As this algorithm is not an exact approach, each problem instance is solved 5 times. However, in the evaluation of policies, only the cheapest solution is considered in each case.

5.5.1 Comparison proximity measures

In Section 5.2.1 we present 4 proximity-based measures named: pickup-related, delivery-related, relatedness of Shaw (1998) and time-related by Lehuédé et al. (2014). The objective of this

Id vehicle	Configurations						Cost		
	S	W	S	W	S	W	C.Fixed	α (km)	β (hour)
V1	4	0					26.45	0.14	27.39
V2	4	0					26.67	0.15	27.39
V3	6	0					27.84	0.16	27.39
V4	4	0					28.09	0.17	27.39
V5	4	0					29.45	0.12	27.39
V6	6	0					30.18	0.17	27.39
V7	2	1					38.03	0.18	27.39
V8	4	0					38.31	0.08	27.39
V9	7	0					39.73	0.21	27.39
V10	4	2	5	1	6	0	42.11	0.24	27.39
V11	4	1					42.25	0.19	27.39
V12	3	3	5	1			42.47	0.23	27.39
V13	3	3	2	4	5	2	43.47	0.24	27.39
V14	3	3					43.9	0.24	27.39
V15	7	1					44.95	0.26	27.39
V16	4	3	5	2			45.25	0.25	27.39
V17	16	0					46.61	0.32	30.13
V18	8	0					47.43	0.22	27.39
V19	7	1	6	2			51.03	0.24	27.39
V20	2	1					53.66	0.11	27.39
V21	3	5	4	4	5	3	53.68	0.27	27.39
V22	22	0					75.13	0.40	30.13

Table 5.2: Vehicle list

section is to evaluate which measure is better for the GIHP problem. Note, the GIHP problem is a DARP with relatively few delivery points.

The complete problem of 565 users is solved using the procedure of Section 5.3. In the first stage, all the possibilities between proximity measures and clustering policies (MSI and USER) are computed. For each possibility, four scenarios are evaluated. The problem is decomposed in 2, 4, 8, and 16 clusters respectively. Figure 5.7 presents the best results for each measure used in any of both clustering policy MSI and USER. The y-axis shows the relative cost with respect to the best solution found in the corresponding scenario.

Overall the pickup-based measure outperforms other measures. The delivery-based does not show good performance probably because of the small number of delivery points (MSI) compared to pickup points. Surprisingly, the Shaw measure was dominated for both pickup and delivery measures. Only in the solution with 16 clusters performs slightly better than the delivery measure. It seems the distance between pickups and/or between deliveries is more important than other distances (i.e. pickup to delivery or conversely). Finally, the time-related measure has worse results overall. This measure considers both pickup and delivery locations from a temporal standpoint.

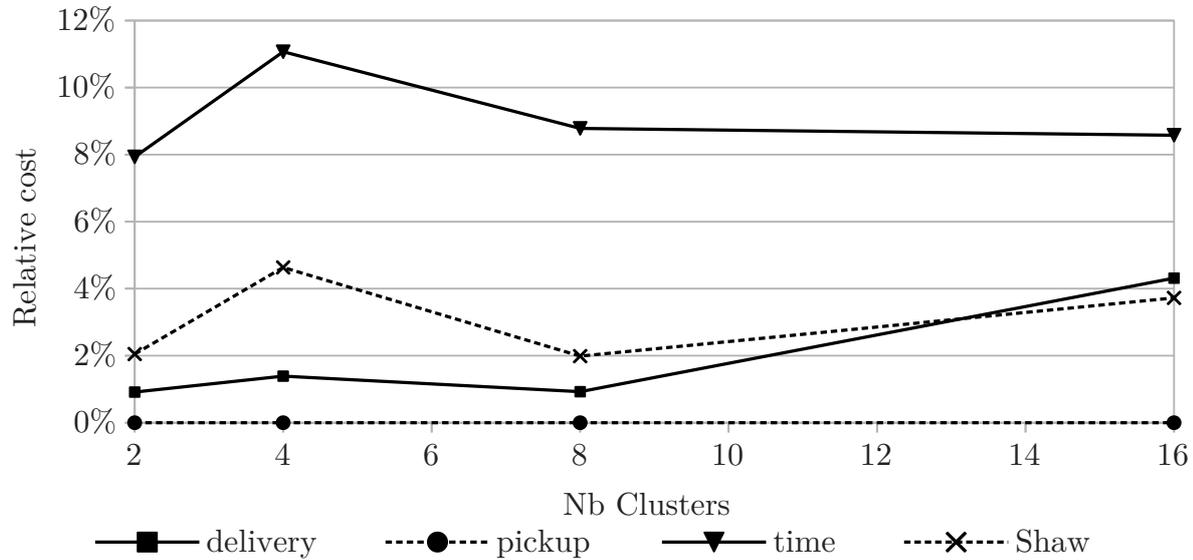


Figure 5.7: Comparison of proximity measures

5.5.2 Comparison of MSI clustering policies (C-MSI)

The C-MSI policy consists of clustering MSIs as presented in Section 5.2.2. Table 5.3 presents the best solution of the C-MSI policy compared with the best solution found (BKS). The BKS solution was computed solving the complete problem without the clustering phase during a very long computation time until the solution is stucked in a local optimum. BKS is a reference solution. Five different splits were tested for the C-MSI policy. Division in 2, 4, 8, 16 and 33 clusters as shown in parentheses. The last split in of 33 clusters corresponds to the CURRENT policy which is one cluster per MSI.

	MSI(33)	MSI(16)	MSI(8)	MSI(4)	MSI(2)	MSI(1)
Relatedness	-	pickup	pickup	pickup	delivery	-
Total cost	11604.7	10323.9	9730.6	9337.3	8894.7	8620.1
Cost increase (%)	34.6%	19.8%	12.9%	8.3%	3.2%	0.0%
Nb. Routes	129	97	85	77	69	65
Routes pooling (%)	0%	40%	51%	68%	77%	85%
Routes reconfiguring (%)	0%	9%	7%	9%	13%	18%
Max Nb. Reconfigurations	0	1	1	2	2	3
Median. Ride time (min)	24.55	24.8	24.6	24.7	23.7	23.3
Avg. Ride time (min)	27.5	27.0	27.0	26.7	26.2	25.9
StdDev Ride time (min)	16.4	15.9	16.0	15.9	16.0	15.8
Coef. Variation (CV=StdDev/Mean)	0.6	0.6	0.6	0.6	0.6	0.6

Table 5.3: Comparison MSI clustering policy

From an economic perspective, the results confirm the correlation between the number of clusters and the transport costs. The smaller the number of clusters, the lower the costs. Note, in this study the CURRENT policy increases by 34.6% the transport cost. With the C-MSI policy, it can be saved between 14.8% (=34.6%-19.8%) and 31.4% (=34.6%-3.3%) depending on the chosen scenario. Cost reductions are explained by a relevant decrease in the number of routes from 129 (CURRENT) to 65 (C-MSI(2)). This reduction is reflected in longer routes with a growing percentage of MSI pooling (up to 85%) and a growing percentage of en-route reconfigurations (up to 18%).

From the service quality perspective, ride times are similar. There is a slight decrease in the average ride time with the reduction of clusters. However, this difference is not significant regarding the standard deviation. The variability of rides times is small and stable overall scenarios (CV=0.6). These results show that despite a stronger restriction in the clustering possibilities of this policy, there are significant economic advantages and the service time is not affected by the number of clusters.

5.5.3 Comparison of user clustering policies (C-USER)

The C-USER policy is a more flexible scenario for clustering users. Table 5.4 presents the best solutions for the C-USER policy and the CURRENT policy. The number in parentheses represents the number of clusters of each scenario. The CURRENT policy has 33 clusters because of the 33 MSIs of the case study.

	CURRENT	USER(16)	USER(8)	USER(4)	USER(2)	BKS(1)
Relatedness	-	pickup	pickup	pickup	pickup	-
Total cost	11604.7	10364.0	9814.9	9253.6	8814.1	8620.1
Cost savings (%)	34.6%	20.2%	13.9%	7.3%	2.2%	0.0%
Nb. Routes	129	81	75	69	67	65
Routes pooling (%)	0%	77%	80%	83%	82%	85%
Routes reconfiguring (%)	0%	15%	19%	17%	16%	18%
Max Nb. Reconfigurations	0	2	2	2	2	3
Median. Ride time (min)	24.6	24.0	24.0	24.3	23.3	23.3
Avg. Ride time (min)	27.5	26.0	26.3	26.6	26.1	25.9
StdDev Ride time (min)	16.4	14.2	14.9	15.7	15.7	15.8
Coef. Variation (CV=StdDev/Mean)	0.6	0.5	0.6	0.6	0.6	0.6

Table 5.4: Comparison users clustering policy

From an economic perspective, these results show the same behavior about the relationship between the number of clusters and the total cost. With the C-USER policy, it can be saved between 14.4%(=34.6%-20.2%) and 32.4%(=34.6%-2.2%). However, savings decrease with the number of clusters. Cost savings are explained by the reduction of routes from 127 (CURRENT)

to 67 (C-USER(2)). Again, this cost decrease is accompanied by a significant reduction in the number of routes, an increase in the number of routes visiting several MSIs (up to 85%) and an increasing proportion of routes reconfiguring en-route (up to 18%). These results confirm the economic potential of promoting transport pooling policy and the use of reconfigurable vehicles. Overall, ride times do not vary considerably from one scenario to another. We can only observe see a slight increase in the median of C-USER(4). However, this change it is not significant.

5.5.4 Global comparison

From a global perspective, we can compare all clustering policies at a time. Table 5.5 presents the general characteristics of the best solutions found with each policy. BKS refers to the best overall cost-based solution.

From an economic point of view, the CURRENT policy is the most expensive overall. The C-USER strategy has a lower total cost than the C-MSI strategy, except in the 4-cluster scenario where it is 1% more expensive (= 13.9-12.9). However, cost differences are so small that cost can be considered equal. Regarding the number of routes, the C-USER policy is superior to MSI policy. The difference in the number of routes grows with the number of clusters from 2 (=69-67) to 16 (=97-81). The number of routes in the C-USER policy is smaller because routes are longer in distance and duration.

Figure 5.8 shows the behavior of routes for C-USER and C-MSI policies all along with the number of clusters. The first two lines, from top to bottom, show graphically the number of routes in the solution. The next two lines, the number of routes pooling transport between different MSIs. C-USER has fewer routes than C-MSI, and when the difference gets bigger, the difference in the number of routes pooling increases as well. For the C-MSI policy, the chances of transport pooling decrease with the number of clusters (33 clusters gets 0 routes pooling). The number of routes reconfiguring en-route is quite stable. The C-USER policy, however, is less sensitive to the variation in the number of clusters. Regarding the quality of service, there are no significant differences between C-MSI and C-USER policies. The low variability and stability of ride times confirm the robustness of the maximum ride time function.

Figure 5.9 shows the number of requests in each cluster for a solution of each policy. The number in parentheses represents the number of clusters of the solution. The CURRENT policy shows the actual number of users at each MSI. The solution of the USER (16) policy increases the number of users in medium and big size clusters. This behavior is accentuated with the MSI (16) policy assigning a larger proportion of users to already big clusters. Thus, the C-USER policy seems to create more equitable clusters than the MSI policy. If this balance is an objective of the decision-maker, the C-USER policy would be preferred.

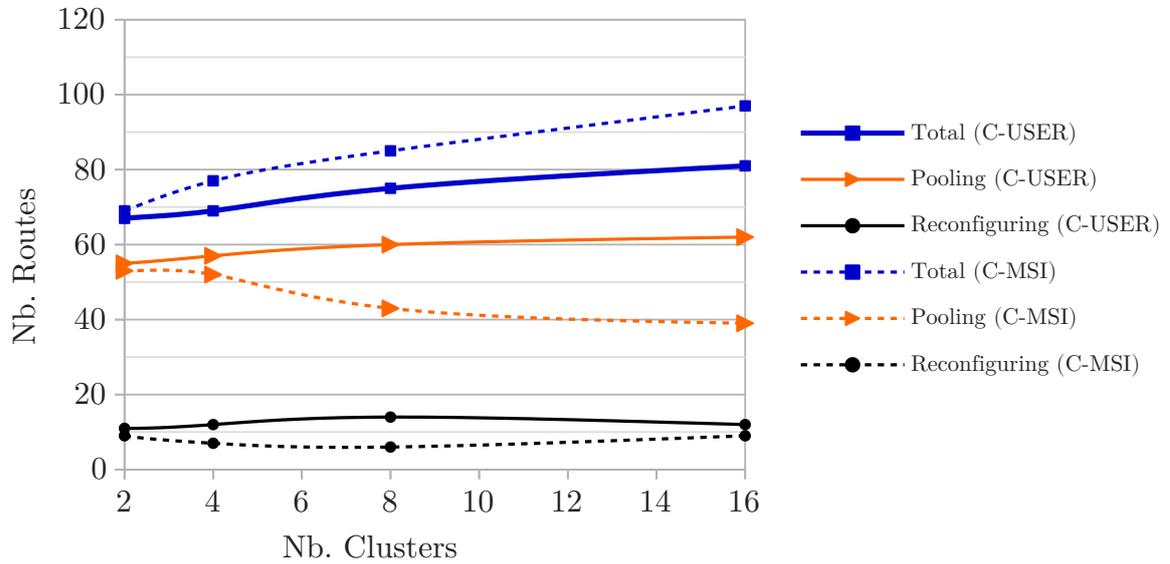


Figure 5.8: Comparison clustering policies

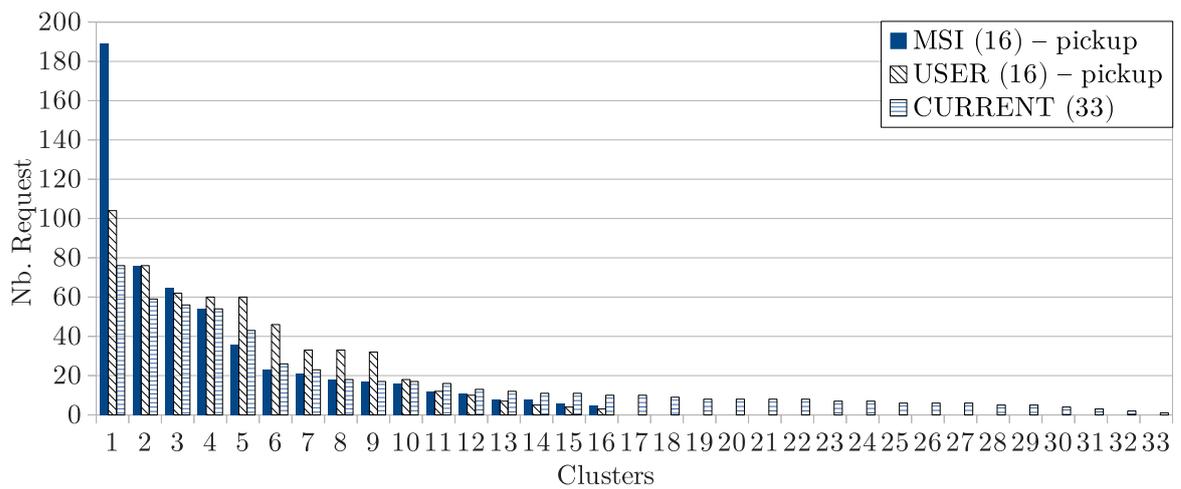


Figure 5.9: Assignment of MSI to clusters

Nb.Clusters	CURRENT	MSI				USER				BKS
	33	16	8	4	2	16	8	4	2	1
Relatedness	-	pickup	pickup	pickup	delivery	pickup	pickup	pickup	pickup	-
Total cost	11604.7	10323.9	9730.6	9337.3	8894.7	10364.0	9814.9	9253.6	8814.1	8620.1
Cost Increase (%)	34.6%	19.8%	12.9%	8.3%	3.2%	20.2%	13.9%	7.3%	2.2%	0.0%
Routes distance (km)	7998	7598	7315	7178	6943	8297	7802	7298	7056	6896
Routes duration (h)	190	187	182	178	172	202	192	181	175	171
Nb. Routes	129	97	85	77	69	81	75	69	67	65
Nb. Routes pooling	0	39	43	52	53	62	60	57	55	55
Nb. Routes reconfiguring	0	9	6	7	9	12	14	12	11	12
Max Nb. Reconfigurations	0	1	1	2	2	2	2	2	2	3
Median Ride time	24.6	24.8	24.6	24.7	23.7	24.0	24.0	24.3	23.3	23.3
Avg. Ride time	27.5	27.0	27.0	26.7	26.2	26.0	26.3	26.6	26.1	25.9
Std. Dev Ride time	16.4	15.9	16.0	15.9	16.0	14.2	14.9	15.7	15.7	15.8
Coef. Variation (CV)	0.6	0.6	0.6	0.6	0.6	0.5	0.6	0.6	0.6	0.6

Table 5.5: Global comparison

5.5.5 Fleet utilization

A relevant characteristic of solutions is the fleet of selected vehicles. Table 5.6 presents the percentage of vehicle types selected for 3 representative scenarios: CURRENT, C-MSI(2) and C-USER(2). The last two with 2 clusters each.

Id vehicle	Configurations						Cost		
	S	W	S	W	S	W	CURRENT	C-MSI(2)	C-USER(2)
VAN1	4	0	0	0	0	0	11%	7%	0%
VAN2	4	0	0	0	0	0	0%	0%	0%
VAN3	6	0	0	0	0	0	17%	19%	23%
VAN4	4	0	0	0	0	0	0%	0%	0%
VAN5	4	0	0	0	0	0	2%	3%	3%
VAN6	6	0	0	0	0	0	0%	0%	0%
VAN7	2	1	0	0	0	0	5%	7%	7%
VAN8	4	0	0	0	0	0	0%	0%	0%
VAN9	7	0	0	0	0	0	2%	0%	0%
VAN10	4	2	5	1	6	0	3%	4%	3%
VAN11	4	1	0	0	0	0	0%	0%	4%
VAN12	3	3	5	1	0	0	40%	42%	39%
VAN13	3	3	2	4	5	2	17%	15%	19%
VAN14	3	3	0	0	0	0	0%	0%	0%
VAN15	7	1	0	0	0	0	0%	0%	0%
VAN16	4	3	5	2	0	0	0%	0%	0%
VAN17	16	0	0	0	0	0	0%	0%	0%
VAN18	8	0	0	0	0	0	0%	0%	0%
VAN19	7	1	6	2	0	0	2%	0%	1%
VAN20	2	1	0	0	0	0	3%	1%	0%
VAN21	3	5	4	4	5	3	0%	0%	0%
VAN22	22	0	0	0	0	0	0%	0%	0%

Table 5.6: Vehicle's utilization

Overall, we can note that selected vehicles are concentrated in very few models. This suggests that the current fleet can be reduced to dominant models. Additionally, the percentage of reconfigurable vehicles is higher in the policies allowing transport pooling (C-MSI and C-USER).

Reconfigurable vehicles VAN12 and VAN13 are predominantly used in all scenarios. These vehicles have interesting configurations (i.e. 3 seats + 3 wheelchairs) and low ownership costs compared to other reconfigurable vehicles. Vehicle VAN10 which is cheaper was very little use probably because of its limited capacity for wheelchairs (up to 2). Vehicle VAN21 with a big capacity was not selected due to the high ownership cost. Even though the CURRENT scenario does not reconfigure en-route, there is significant utilization of these vehicles probably because of the (3-3) configuration.

Non-reconfigurable vehicles are also used but mostly in the CURRENT scenario. Vehicle

1, for example, is significantly used in the CURRENT scenario. Vehicle VAN3 is used in all 3 scenarios. This knowledge of vehicles can guide transporters like the GIHP in the process of designing or buying new vehicles.

5.6 Conclusions

GIHP manages the transport independently for each MSIs. This policy has many advantages in the processes of planning, negotiation, pricing, and contracting with each MSI. However, this policy is sub-optimal from the economic point of view and the additional cost must be compared with the benefits. In this study, the cost of the CURRENT policy is estimated to around 30% when it is compared with the best possible scenario.

Two alternative policies, allowing the pooling of transport between several MSIs, are proposed. The C-MSI policy grouping MSIs and the C-USER policy grouping users. The most flexible policy (C-USER) assumes that users from one MSI can be split into different clusters. These policies can save between 15% (16 clusters) and 30% (2 clusters) of transport costs compared to the CURRENT policy depending on the number of clusters. Both policies have comparable savings, but C-MSI policy has fewer routes which from a management point of view makes C-USER policy better. This reduction is correlated to an increase in the number of routes pooling users from several MSIs. Selected vehicles are concentrated in very few dominant models (mostly reconfigurable). These vehicles have interesting configurations (like 3 seats + 3 wheelchairs) and low fixed ownership costs.

Regarding service quality, we found that user ride times are very stable regardless of the policy in the presence of maximum ride time constraints. Moreover, the proposed maximum ride time function currently used by GIHP company assures a good service level and low ride time variability.

As a perspective, we identified that balancing the number of users in each cluster could be desirable. This criterion could be integrated with the clustering model, as a constraint or an objective, to offer further insights.

Conclusions and perspectives

6.1 Conclusions

In this dissertation, we study the daily transport for people with disabilities (PWD). More precisely, the optimization of the long-term transport of PWD going to Medico Social Institutions (MSI).

In 2010, healthcare transport in France became one of the top ten priorities of the risk management plan due to the increasing cost of these transports. For Medico-Social Institutions (MSI), transport costs often represent the second largest expenditure after that of the staff ANAP (2016b). Despite this, few MSIs have the skills and means to effectively manage the transport, provide high-quality service and keep the costs down ANAP (2016b). In this context, this thesis as part of the NOMAd project proposed on one side a global optimization strategy and a transport pooling strategy for between MSIs on the other side. Both strategies with the objective are to improve economic performance while maintaining social and environmental goals.

The global optimization strategy consists of optimizing the long-term transport planning of PWD using operations research tools. A decision aiding tool for transport optimization was proposed for this purpose. The long-term planning considers one academic year. Following the current practice, we assumed that this transport plan is constructed from a pattern of identical weekly transportation plans. Defining a weekly schedule is a complex task that needs to be split into sub-problems. First, we worked on the planning of each sub-problem called the half-day transportation problem in Chapter 3. Then in Chapter 4 we solved the weekly transport problem considering both cost and service quality objectives into account.

The construction of long-term transportation planning is a complex problem that has not been completely solved in the scientific literature. A literature review about the transportation

problems in the health sector was carried out in Chapter 2. This review helps us to characterize this transport type and to confirm that both the half-day and weekly transport problems are relevant contributions to the healthcare transport of people.

In Chapter 3, we solved the half-day transport problem consisting of designing the routes and determining the vehicles needed to serve the demand at a lower cost respecting service time constraints. This problem was modeled as a new variant of the Dial-A-Ride Problem (DARP) called the “Fleet size and mix DARP with configurable vehicle capacity” (FSM-DARP-RC). This problem considers real-life features such as heterogeneous users (i.e. seat, wheelchair) and vehicles that can be configured en-route. Companies very often own configurable vehicles. Yet route designers do not plan routes considering this extra degree of flexibility. However, this study showed that companies can easily save up to 2.5% of the total cost just by allowing vehicles to use en-route reconfiguration. The utilization of reconfigurable vehicles is strongly dependent on vehicle ownership costs. For the evaluated instance, reconfigurable vehicles are advantageous for companies when vehicle ownership costs are less than 20% the cost of standard non-reconfigurable vehicles. More generally, this is the second contribution to a pickup and delivery problem with configurable vehicle capacity. The proposed approach presents original and efficient handling of reconfigurable capacities in an LNS heuristic. Experiments showed that it outperforms the previously proposed algorithm on this topic.

In Chapter 4, we solved the weekly transport problem to make half-day transport plans time-consistent for each user. This objective is proposed because some disabilities are very sensitive to inconsistent schedules. This problem is modeled as a new variant of the multi-period DARP denoted the “time consistent DARP”. Unlike the half-day problem, this problem defines two objectives, the transport cost, and the user time-consistency. The time consistency objective is expressed as a lexicographic function of the number of users with different time-classes. This approach takes more time than traditional min-max objectives but returns a Pareto frontier that helps decision-makers selecting the appropriate solution. In this study, we showed that economic solutions are already fairly consistent with very few passengers having 3 time-classes. Moreover, with a small increase in transportation cost (<1%) passengers’ schedules will be at most 2 time-classes each. Additionally, we showed that allowing a flexible departure of routes improves the transportation costs of highly consistent solutions.

Finally, in Chapter 5 we evaluated both the economic and service quality impact of pooling the transport between different several MSIs. Currently, each MSI manages its planning (fleet, drivers and routes) independently because of the advantages planning, contracting and negotiating. However, this policy is sub-optimal from an economic point of view. The cost of the current policy is estimated at around 30% when it is compared with the best possible scenario. Two alternative policies allowing the pooling of transport between several MSIs are proposed with different degrees of management rules to construct clusters. We found that both policies have comparable savings but the policy with higher flexibility has fewer routes. Regarding

service quality, we found that the characteristics of user rides were not sensitive to the different policies in the presence of maximum ride time constraints.

6.2 Perspectives

Generally, the methods and algorithms developed in this thesis can be applied to domains other than the transport of patients and passengers. The relaxation of ride time constraints for example results in another routing problem called pickup and delivery problem. This problem is commonly applied in the transport of goods in courier companies. In particular, future research for the different topics treated through the chapters are:

In Chapter 1, the global methodology assumes that morning and afternoon routes are independent of each other. However, PWD willing to accept long ride times in the morning would expect short ride times in the afternoon and conversely. Thus, a model extension would be to consider other objectives or constraints to balance the ride times of users in vehicles.

In Chapter 3, the FSM-DARP-RC model can be extended with additional operational features such as the choice of different departure and arrival depots with limited capacity. For example, at GIHP some vehicles can choose between departing from the driver home, an MSI or the central depot. Incorporating real-time information (traffic, variable demand) may require the use of state-space-time networks with a fine time discretization (see, e.g. Mahmoudi and Zhou (2016)) and specific optimization algorithms such as the Dynamic Discretization Discovery (Boland et al., 2017). Another research area is the extension of the proposed algorithm to other types of pickup and delivery problems where reconfigurations are sometimes needed (e.g. trucks with adjustable compartments).

In Chapter 4, the TC-DARP was solved for a fleet of homogeneous vehicles. Extending the problem to consider a fleet of reconfigurable vehicles would require additional assignment constraints in the weekly problem. The question of driver-consistency is very sensitive for some disabilities. Braekers and Kovacs (2016) studied the economic impact of driver-consistency on a DARP context. However, further research should include real-life features such as driver breaks, work balance, and contracting. Moreover, a multi-objective approach between transport cost, time-consistency, and driver-consistency in a DARP setting will be an interesting tool for decision-makers.

In Chapter 5, the proximity measures studied between requests (users) partially take into account different aspects of DARP. However, an indicator that considers together pickup and delivery locations, time windows, ride times and load are needed. Proximity measures are not only used in the clustering problems but the selection of requests in neighborhood-based heuristics (Pisinger and Ropke, 2007; Prescott-Gagnon et al., 2009). The solved clustering problem minimizes the sum of transport costs between users and the cluster's centroid. However, other objective functions such as TSP tours or spanning trees between the users of the same

cluster can provide a better estimation of the real transport costs.

A more general perspective would be to handle planning modifications in the context of regular transport services. Planning modifications are not too many but can be frequent in large scale applications. In those cases, transport planners should adjust vehicle routes to adapt them to this new context. Transport planners have to minimize two antagonistic objectives: transport costs and the scheduling modifications for passengers already scheduled in the former routes. To our knowledge, this problem has not been studied in the literature.



Chapter 3

A.1 Parameters used for ALNS-based metaheuristics

reinitialize incumbent solution with the best solution every 2000 iters.
score for new best solutions $\sigma_1 = 33$
score for new improving solutions $\sigma_2 = 20$
score for new accepted solutions $\sigma_3 = 15$
reaction factor $r = 0.1$
recompute operator weights = 100 iters
minimum weight operators = 0.1
maximum weight operators = 5

Table A.1: Parameters for the ALNS as in Masson et al. (2014)

A.2 Detailed results comparison of LNS-based algorithms

Inst	Time	LNS(5)		ALNS(5)		ALNS(5)-SCP		LNS(2)-SCP		ALNS(5)-RSCP		LNS(5)-RSCP		LNS(2)-RSCP	
		Best	Avg												
I01-80	16	1,791.69	1,798.30	1,803.42	1,811.81	1,757.22	1,757.70	1,757.22	1,757.22	1,757.22	1,757.22	1,757.22	1,757.22	1,757.22	1,757.22
I02-60	16	911.79	913.64	911.88	926.49	925.09	926.38	911.79	913.21	915.02	922.48	911.79	914.45	915.34	922.63
I03-80	16	1,869.42	1,882.15	1,881.31	1,893.74	1,838.33	1,838.60	1,838.47	1,838.47	1,811.86	1,811.86	1,811.86	1,811.86	1,811.86	1,811.90
I04-70	16	1,600.54	1,603.65	1,599.53	1,602.58	1,599.49	1,599.52	1,599.49	1,599.50	1,599.49	1,599.51	1,599.49	1,599.49	1,599.49	1,599.49
I05-80	16	1,251.24	1,253.37	1,244.86	1,251.00	1,242.48	1,242.62	1,242.48	1,242.48	1,242.48	1,242.48	1,242.48	1,242.48	1,242.48	1,242.48
I06-80	16	1,451.57	1,452.69	1,460.48	1,464.95	1,418.74	1,421.09	1,418.74	1,422.19	1,418.74	1,421.31	1,418.74	1,420.88	1,418.74	1,418.75
I07-60	16	1,537.63	1,537.63	1,537.63	1,537.79	1,537.63	1,537.63								
I08-65	16	1,204.89	1,205.68	1,204.89	1,218.95	1,204.89	1,205.11	1,204.89	1,205.11	1,204.89	1,205.54	1,204.89	1,204.89	1,204.89	1,205.11
I09-120	40	3,165.79	3,186.43	3,178.83	3,189.00	3,128.00	3,128.00								
I10-135	40	2,196.36	2,208.65	2,169.89	2,219.19	2,059.99	2,069.46	2,059.91	2,070.72	2,071.42	2,087.16	2,081.24	2,088.84	2,059.91	2,079.79
I11-160	40	2,343.57	2,347.28	2,310.38	2,331.33	2,224.06	2,238.53	2,226.56	2,243.24	2,225.54	2,229.27	2,222.14	2,225.37	2,222.14	2,228.43
I12-160	40	2,851.89	2,864.54	2,880.07	2,911.25	2,615.00	2,627.80	2,620.93	2,638.95	2,623.44	2,628.19	2,614.47	2,631.98	2,623.24	2,634.22
I13-280	100	5,227.05	5,255.18	5,227.45	5,275.53	5,153.87	5,196.40	5,123.11	5,132.67	4,864.49	4,896.87	4,843.39	4,870.33	4,885.02	4,917.22
I14-295	100	5,475.69	5,497.11	5,476.19	5,533.58	5,352.61	5,411.52	5,387.16	5,431.81	4,893.30	4,927.77	4,921.74	4,948.96	4,904.47	4,931.81
	Avg	2,348.51	2,357.59	2,349.06	2,369.08	2,289.81	2,300.03	2,289.74	2,297.23	2,235.25	2,242.52	2,235.36	2,241.60	2,236.46	2,243.91
	Avg Gap	3.60%	3.94%	3.61%	4.46%	1.34%	1.64%	1.27%	1.52%	0.13%	0.38%	0.12%	0.31%	0.13%	0.39%
	NbBKS	3		2		7		9		9		12		10	

Table A.2: Performance comparison of LNS-based heuristics. Time limits in column 2 are expressed in minutes. Average costs (Avg) are computed over 5 runs. Nb BKS refers to the number of best-known solutions (BKS). The Gap for every instance is computed as $(value - BKS)/BKS \times 100$. General parameters for all metaheuristics can be found in Table A.1. Additional parameters for ALNS can be found in Table A.1.

A.3 Detailed results benchmark Cordeau and Laporte (2003)

Inst	BKS	TA			ELS			ALNS ¹			LNS(2)-RSCP ²		
		Braekers et al. (2014)			Chassaing et al. (2016)			Gschwind and Drexl (2016)			Best	Avg	Time
		Best	Avg	Time	Best	Avg	Time	Best	Avg	Time	Best	Avg	Time
pr01	190.02	190.02	190.02	16.6	190.02	190.02	15.0	190.02	190.02	15.1	190.02	190.02	24.0
pr02	301.34	301.34	301.34	42.0	301.34	301.34	75.0	301.34	301.34	41.1	301.34	301.34	145.9
pr03	532.00	532.10	533.54	48.8	532.43	533.86	138.0	532.00	532.01	62.1	532.00	532.00	256.6
pr04	570.25	577.16	580.52	74.6	570.54	574.47	442.2	570.25	570.61	138.7	571.25	572.23	493.9
pr05	625.64	629.80	632.06	89.2	630.82	637.59	724.2	628.59	630.99	267.4	631.38	633.20	735.7
pr06	783.78	797.78	800.68	107.0	792.80	796.10	1,315.2	789.36	790.44	397.5	789.51	790.23	1,216.6
pr07	291.71	292.23	292.23	22.6	291.71	292.96	28.2	291.71	291.71	19.7	291.71	291.71	55.3
pr08	487.84	490.94	491.00	48.6	491.60	493.16	160.8	489.83	491.13	74.1	489.33	492.49	329.4
pr09	653.94	662.64	666.65	72.2	672.86	681.35	675.0	659.69	660.13	151.7	659.10	663.00	668.8
pr10	845.47	853.98	860.83	114.4	857.36	860.68	1,279.8	853.07	857.18	389.2	856.95	860.91	1,236.6
pr11	164.46	164.46	164.46	23.8	164.46	164.46	16.8	164.46	164.46	17.7	164.46	164.46	43.6
pr12	295.66	295.69	296.06	51.4	295.66	295.72	82.2	295.66	296.22	48.1	295.67	296.83	227.6
pr13	484.83	488.61	490.03	76.2	489.00	490.70	222.0	484.83	484.83	101.3	488.30	488.70	478.4
pr14	529.33	534.99	540.99	117.0	531.08	531.98	612.0	531.19	531.92	204.2	529.33	532.13	966.2
pr15	573.56	581.46	584.33	155.2	578.44	580.23	1,195.8	576.70	578.17	491.6	577.00	579.47	1,584.0
pr16	725.22	743.56	747.19	180.6	731.25	736.59	1,939.2	731.50	736.08	692.6	732.60	737.12	2,331.9
pr17	248.21	249.33	249.33	34.0	248.21	248.21	34.8	248.21	248.21	22.8	248.21	248.21	104.8
pr18	458.73	461.77	462.38	81.0	461.21	462.40	259.2	461.48	461.67	103.0	458.73	461.42	520.8
pr19	592.23	598.23	600.63	146.4	595.39	597.53	745.8	593.83	596.77	361.4	594.22	597.65	1,225.3
pr20	783.81	795.08	801.89	162.8	796.60	803.99	1,887.0	787.14	789.83	591.6	789.49	799.38	1,895.8
Avg	Gap	0.81%	1.16%	83.2	0.64%	1.04%	592.4	0.32%	0.50%	209.5	0.38%	0.72%	727.1
	Nb BKS		3			6			9			8	

Table A.3: Results on the DARP instances of Cordeau and Laporte (2003). Average cost (Avg) is computed over 5 runs. Time is the average computation time in seconds. Nb BKS refers to the number of best known solutions. The Gap for an instance is computed as $(value - BKS)/BKS \times 100$

A.4 Detailed results benchmark Qu and Bard (2013)

According to Qu and Bard (2013), we denote by C_t the cost related to travel time. It is computed as the total time (traveled distance divided by *speed*), times the cost per unit of time $c_{time} = \$0.377/min$. C_p denotes the total passenger ride time cost with the unitary cost $\tau_c = \$0.0001$ per minute. The vehicle ownership cost is $C_v = \$0.01/van$. Finally, the total cost is computed as $C_{total} = C_t + C_p + C_v$. 3 runs were performed of the LNS(2)-RSCP with an stopping criteria of 1 hour. For each instance C_{total} corresponds to the best total cost found over 3 runs. The Gap is computed as in regarding to the best cost of Qu and Bard (2013).

Inst	MSALNS				LNS(2)-RSCP						
	C_t	C_p	C_v	C_{total}	C_t	C_p	C_v	C_{total}	Gap (*)	Gap Avg	
A051	43.00	0.20	0.04	43.24	42.29	0.22	0.09	42.60	-1.47%	-1.71%	
A151	49.82	0.23	0.06	50.11	50.18	0.27	0.06	50.51	0.80%	0.18%	
A101	43.99	0.22	0.05	44.26	44.75	0.25	0.08	45.07	1.84%	1.17%	
A251	62.42	0.29	0.06	62.77	55.18	0.28	0.08	55.54	-11.52%	-10.95%	
A201	53.57	0.23	0.06	53.86	52.81	0.27	0.06	53.14	-1.33%	-2.28%	
A052	36.86	0.20	0.05	37.11	42.25	0.23	0.09	42.58	14.73%	13.99%	
A102	45.19	0.19	0.05	45.43	45.34	0.30	0.07	45.71	0.61%	2.92%	
A152	49.64	0.21	0.05	49.90	44.37	0.29	0.09	44.75	-10.32%	-8.17%	
A202	49.79	0.24	0.06	50.09	48.56	0.30	0.09	48.94	-2.29%	-0.76%	
A252	55.21	0.28	0.06	55.55	52.73	0.28	0.09	53.10	-4.42%	-4.62%	
Avg	48.95	0.23	0.05	49.23	47.85	0.27	0.08	48.19	-1.34%	-1.02%	

Table A.4: Benchmark Qu and Bard (2013) instance “A” scenario (iii) without groups

Inst	MSALNS				LNS(2)-RSCP						
	C_t	C_p	C_v	C_{total}	C_t	C_p	C_v	C_{total}	Gap	Avg Gap	
B201	43.99	0.22	0.05	44.25	37.01	0.37	0.06	37.44	-15.39%	-15.09%	
B301	44.84	0.25	0.05	45.15	37.19	0.30	0.06	37.54	-16.85%	-16.07%	
B401	47.65	0.21	0.05	47.91	38.82	0.26	0.05	39.13	-18.33%	-18.28%	
B501	47.87	0.20	0.05	48.12	38.07	0.30	0.05	38.43	-20.15%	-19.69%	
B601	45.83	0.22	0.05	46.10	40.10	0.20	0.05	40.36	-12.46%	-12.12%	
B202	45.19	0.19	0.05	45.43	36.34	0.31	0.05	36.70	-19.21%	-18.66%	
B302	44.42	0.19	0.05	44.66	37.73	0.21	0.05	37.98	-14.95%	-14.83%	
B402	47.90	0.19	0.05	48.14	36.63	0.24	0.05	36.92	-23.30%	-23.10%	
B502	46.50	0.18	0.05	46.73	36.93	0.24	0.05	37.21	-20.36%	-17.73%	
B602	45.72	0.20	0.05	45.97	37.73	0.30	0.05	38.08	-17.16%	-16.75%	
Avg	45.99	0.21	0.05	46.25	37.66	0.27	0.05	37.98	-17.82%	-17.23%	

Table A.5: Benchmark Qu and Bard (2013) instance “B” scenario (iii) without groups

Inst	MSALNS				LNS(2)-RSCP				Gap	Avg Gap
	C_t	C_p	C_v	C_{total}	C_t	C_p	C_v	C_{total}		
B201	43.44	0.22	0.05	43.72	36.34	0.35	0.05	36.74	-15.97%	
B301	43.90	0.27	0.05	44.21	36.47	0.35	0.05	36.86	-16.62%	
B401	43.99	0.29	0.05	44.32	37.20	0.25	0.04	37.50	-15.39%	-15.59%
B501	42.94	0.22	0.05	43.21	36.43	0.32	0.05	36.81	-14.82%	
B601	43.22	0.21	0.05	43.48	36.47	0.37	0.05	36.90	-15.14%	
B202	42.13	0.27	0.05	42.45	36.87	0.28	0.05	37.19	-12.38%	
B302	42.13	0.27	0.05	42.45	36.87	0.28	0.05	37.19	-12.38%	
B402	42.13	0.27	0.05	42.45	37.99	0.25	0.04	38.28	-9.82%	-9.82%
B502	40.38	0.23	0.04	40.65	37.69	0.26	0.04	37.99	-6.54%	
B602	39.70	0.23	0.05	39.98	36.49	0.25	0.05	36.79	-7.99%	
Avg	42.40	0.25	0.05	42.69	36.88	0.30	0.05	37.22	-12.71%	-12.71%

Table A.6: Benchmark Qu and Bard (2013) instances “B” ignoring walker requirements.

B.0.1 Construction of $\mathcal{TW}(\mathcal{H}, u)$

Algorithm 10: Construction of $\mathcal{TW}(\mathcal{H}, u)$

Parameters: u : user considered. $\mathcal{H} = \{h_1, \dots, h_M\}$: set of the M service times of user u sorted in non-decreasing order.

Data: Λ : width of time-classes.

Output: The set of multiple time windows $\mathcal{TW}(\mathcal{H}, u)$

```

1 mtw =  $\emptyset$           /* initialize an empty set of multiple time windows */
2  $\bar{h} = h_1$ 
3  $\underline{h} = h_1$ 
4 for  $i = 2, \dots, M$  do
5   if  $h_i > \underline{h} + \Lambda$  then
6     mtw  $\leftarrow$  mtw  $\cup$   $\{[\max\{a_{p_u}; \bar{h} - \Lambda\}, \min\{b_{p_u}; \underline{h} + \Lambda\}]\}$ 
7      $\underline{h} = h_i$ 
8      $\bar{h} = h_i$ 
9 mtw  $\leftarrow$  mtw  $\cup$   $\{[\max\{a_{p_u}; \bar{h} - \Lambda\}, \min\{b_{p_u}; \underline{h} + \Lambda\}]\}$ 
10 return mtw

```

B.0.2 Parameters LSN-SCP

Parameters to generate pool \mathcal{L}	
$\chi = 5\%$	record-to-record acceptance criterion.
$penalty = 10000$	penalty cost for incomplete solutions.
$\Phi^- = 10\%$	minimal proportion of removed request used by removal operators.
$\Phi^+ = 45\%$	maximal proportion of removed request used by removal operators.
$p = 6$	roulette wheel parameter for the historical node-pair operator.
$\sigma_{init}^+ = 4\text{-regret}$	repair operator for building the initial solution
$\eta = 1000$	launch frequency of the SCP.
$Iters = 10000$	max number of iterations.
$\psi = 1.25$	RSCP coefficient to recompute the launch frequency of the SCP.
New parameters to generate pool \mathcal{L}_{new}	
$Iters = 250$	max number of iterations.
$\eta = \infty$	the SCP is deactivated.

Table B.1: Parameters LNS-SCP (Tellez et al., 2018b).

B.0.3 Parameters MSPP tests

- **MSPP** : algorithm as presented in this chapter.
 - $\mathcal{B} = \{(D(\mathcal{L}), SS),$
 - $\mathcal{O} = \{(\mathcal{L}, SS); (\mathcal{L}_{new}, SS)\}.$
- **MSPP-withRW** : MSPP with the roulette wheel selection rule
 - $\mathcal{B} = \{(D(\mathcal{L}), SS)\}$
 - $\mathcal{O} = \{(\mathcal{L}, SS); (\mathcal{L}, RW); (\mathcal{L}_{new}, SS)\}.$
- **MSPP-noDARPMTW** : MSPP without using DARPMTW.
 - $\mathcal{B} = \{(D(\mathcal{L}), SS)\}$
 - $\mathcal{O} = \{(\mathcal{L}, SS)\}$

B.0.4 Scheduling algorithm DARP

Algorithm 11: Schedule evaluation

Input: Route $\omega = \{1, \dots, M\}$.
Output: The set of service times $h_i \forall i \in \omega$ and the maximal route time shift Δ_ω^+ , or -1 if infeasible

```

1  $h_1 \leftarrow a_1$  /* beginning of the service */
2  $H \leftarrow 0$  /* total waiting time on the route */
3  $F \leftarrow b_1 - h_1$  /* FTS latest start at node 1 */
4  $F' \leftarrow b_1 - h_1$  /* FTS earliest start at node 1 */
   /* Phase 1: set up nodes at the earliest start */
5 for  $i = 2, \dots, M$  do
6    $h_i \leftarrow \max\{a_i; h_{i-1} + \zeta_{i-1} + t_{i-1,i}\}$ 
7   if  $h_i > b_i$  then return -1
8    $H \leftarrow H + \max\{0; a_i - (h_{i-1} + t_{i-1,i} + \zeta_{i-1})\}$ 
9    $F' = F$ 
10   $F \leftarrow \min\{F; H + \max\{0; l_i - h_i\}\}$ 
11  if  $i = M$  then
12  |  $F' \leftarrow \min\{F'; H\}$ 
13
   /* Phase 2: optimize route duration */
14  $\Delta_\omega^+ = F - F'$  /* route time shift */
15  $h_1 \leftarrow h_1 + F'$ 
16 for  $i = 2, \dots, M$  do
17 |  $h_i \leftarrow \max\{h_{i-1} + \zeta_{i-1} + t_{i-1,i}; a_i\}$ 
   /* Check route duration constraint */
18 if  $(h_M - h_1) > \bar{T}$  then return -1
19
   /* Phase 3: check ride time constraints */
20 for  $i = M - 2, \dots, 1$  do
21 | if  $i \in \mathcal{P}$  then
22 | |  $u \leftarrow$  user of pickup  $i$  /* implies  $i = p_u$  */
23 | |  $\delta \leftarrow (h_{d_u} - h_{p_u} + \zeta_i) - \bar{T}_u$ 
24 | | if  $(\delta > 0)$  then
25 | | |  $h_{p_u} \leftarrow h_{p_u} + \delta$ 
26 | | | if  $h_{p_u} > b_i$  then return -1
27 | | | for  $j = p_u + 1, \dots, M$  do
28 | | | |  $w_j \leftarrow \max\{a_j; h_{j-1} + \zeta_{j-1} + t_{j-1,k}\}$ 
29 | | | | if  $h_j > b_j$  then return -1
30 | | | if  $\bar{T}_u - (h_{d_u} - h_{p_u} + \zeta_i) < 0$  then
31 | | | | return -1
32
33 return  $\{h_i | i \in \omega\}, \Delta_\omega^+$ 

```

B.1 TC-VRP

Instance	Transportation cost					Time (min)
	C_{\max}	≤ 5	≤ 4	≤ 3	≤ 2	
data5-15	663.2	663.2	663.2	674.1	782.3	11.6
data5-21	773.7	779.1	779.1	779.1	817.3	21.9
data5-25	617.4	617.4	617.4	622.2	669.0	29.1
data5-26	767.6	767.6	771.9	778.8	815.9	45.8
data5-27	934.6	934.6	934.6	942.0	1026.1	61.2
data5-32	984.9	984.9	984.9	989.9	1034.5	22.1
data5-41	1420.7	1420.7	1422.6	1461.6	1615.9	126.9
data5-44	1142.9	1142.9	1142.9	1149.4	1220.3	74.3
data5-46	1458.9	1458.9	1465.4	1492.9	1607.2	114.4
data5-48	1440.8	1449.4	1452.4	1459.8	1597.8	154.9
data5-55	1569.1	1569.1	1571.1	1581.0	1696.9	139.4
data5-59	2714.8	2721.5	2721.5	2883.6	3115.9	281.3
data5-64	2082.0	2082.0	2100.8	2112.4	2304.2	118.9
data5-65	1759.4	1759.4	1766.2	1777.1	1923.9	128.6
data6-15	689.4	689.4	689.4	695.6	741.3	2.2
data6-21	792.0	792.0	796.2	798.1	831.7	40.9
data6-25	680.9	680.9	683.5	683.5	728.7	32.9
data6-26	838.7	838.7	838.7	843.2	905.0	56.0
data6-27	949.8	949.8	949.8	954.0	1060.3	57.9
data6-32	991.1	991.1	991.1	996.2	1013.5	10.4
data6-41	1500.3	1500.3	1504.2	1506.7	1735.5	92.8
data6-44	1239.1	1239.1	1243.9	1244.6	1405.9	76.7
data6-46	1485.1	1485.1	1496.6	1526.0	1597.0	93.0
data6-48	1507.4	1507.4	1519.4	1531.6	1702.8	158.5
data6-55	1816.8	1816.8	1826.3	1854.4	1987.7	121.0
data6-59	2931.3	2931.3	2933.0	3037.4	3389.6	204.1
data6-64	2264.3	2264.3	2271.0	2305.6	2527.0	100.6
data6-65	1981.9	1981.9	1990.6	1998.3	2219.5	163.6

Table B.2: Benchmark Feillet et al. (2014)

Instance	Transportation cost					Time (min)
	C_{\max}	≤ 5	≤ 4	≤ 3	≤ 2	
data7-15	746.9	746.9	746.9	752.6	790.3	11.9
data7-21	830.1	833.7	833.7	833.7	899.3	39.8
data7-25	719.5	724.7	724.7	728.8	767.9	34.9
data7-26	880.3	880.3	883.5	887.4	920.7	37.9
data7-27	1053.4	1053.4	1053.4	1056.5	1108.5	39.7
data7-32	1079.7	1079.7	1079.7	1097.3	1154.3	57.2
data7-41	1644.8	1644.8	1648.5	1661.0	1730.4	63.2
data7-44	1295.9	1295.9	1300.3	1307.0	1311.7	20.5
data7-46	1648.6	1648.6	1650.1	1664.6	1715.9	72.4
data7-48	1687.1	1698.9	1698.9	1712.0	1774.3	116.6
data7-55	1889.0	1889.0	1896.3	1918.0	2036.0	136.5
data7-59	3262.4	3262.4	3307.4	3596.6	3892.4	725.2
data7-64	2552.9	2552.9	2552.9	2583.6	2823.4	136.9
data7-65	2196.0	2196.0	2196.0	2224.2	2347.3	159.8
data8-15	773.9	773.9	773.9	780.2	808.5	10.8
data8-21	898.4	898.4	898.4	905.0	956.3	37.8
data8-25	853.3	853.3	853.3	853.3	857.0	0.2
data8-26	962.5	962.5	962.5	970.8	998.2	35.2
data8-27	1184.8	1193.9	1193.9	1193.9	1220.6	17.6
data8-32	1144.0	1144.0	1152.8	1152.8	1181.7	35.8
data8-41	1886.8	1886.8	1888.5	1898.0	1954.3	63.9
data8-44	1409.0	1409.0	1409.0	1414.9	1441.2	28.8
data8-46	1758.5	1758.5	1772.9	1774.8	1817.2	33.8
data8-48	1815.7	1815.7	1820.9	1824.4	1898.1	125.7
data8-55	2007.7	2007.7	2015.4	2037.2	2104.0	120.0
data8-59	3545.7	3545.7	3580.5	3874.4	4020.5	427.4
data8-64	2723.3	2723.3	2723.3	2743.0	2978.2	126.8
data8-65	2404.2	2404.2	2422.9	2433.5	2524.4	159.5
data9-15	797.5	797.5	797.5	804.3	816.1	3.6
data9-21	998.6	998.6	998.6	1003.0	1008.9	5.7
data9-25	894.6	894.6	894.6	894.6	908.1	1.6
data9-26	1024.6	1024.6	1024.6	1024.6	1028.0	0.4
data9-27	1210.6	1210.6	1210.6	1219.5	1241.9	12.3
data9-32	1187.7	1187.7	1199.4	1199.4	1204.9	6.4
data9-41	2022.8	2022.8	2022.8	2022.8	2032.1	6.3
data9-44	1532.5	1532.5	1532.5	1532.5	1595.1	54.1
data9-46	1827.0	1827.0	1827.0	1841.6	1871.0	30.7
data9-48	1973.5	1973.5	1973.5	1992.8	2007.7	28.6
data9-55	2176.0	2176.0	2176.0	2188.1	2275.9	70.3
data9-59	3913.0	3913.0	3916.3	3946.6	4025.3	96.3
data9-64	2942.6	2942.6	2964.9	2964.9	2999.6	28.8
data9-65	2586.5	2586.5	2586.5	2604.7	2624.2	68.0

Table B.3: Benchmark Feillet et al. (2014)



Chapter 5

C.0.1 Parameters LNS-SCP

$\chi = 5\%$	record-to-record acceptance percentage.
$penalty = 10\,000$	penalty cost for incomplete solutions.
$\Phi^- = 10\%$	minimal proportion of removed request used by removal operators.
$\Phi^+ = 40\%$	maximal proportion of removed request used by removal operators.
$p = 6$	roulette wheel parameter for the historical node-pair operator.
$\sigma_{init}^+ = 4\text{-regret}$	repair operator for building the initial solution
$\eta = 1\,000$	launch frequency of the SCP.
$t_{limit} = 60\text{ sec}$	imposed time limit for the SCP.
$\psi = 1.25$	RSCP coefficient to recompute the launch frequency of the SCP.

Table C.1: Parameters LNS-SCP algorithm

Bibliography

- Ahmadi-Javid, A., Seyedi, P., Syam, S. S., 2017. A survey of healthcare facility location. *Computers & Operations Research* 79, 223–263.
- Al-Shaqsi, S., 2010. Models of international emergency medical service (ems) systems. *Oman medical journal* 25 (4), 320.
- Aldaihani, M., Dessouky, M. M., 2003. Hybrid scheduling methods for paratransit operations. *Computers & Industrial Engineering* 45 (1), 75–96.
- Allaoua, H., 2014. Routage et planification des personnels pour l’hospitalisation à domicile. Ph.D. thesis, Université Paris-Nord - Paris XIII.
- ANAP, 2013. Améliorer la gestion des transports sanitaires en établissement de santé. Tech. rep., Appui santé & médico social.
- ANAP, 2016a. Améliorer la gestion des transports de personnes handicapées. Tech. rep., Appui santé & médico social.
- ANAP, 2016b. Améliorer la gestion des transports de personnes handicapées. tome 1: Retour d’expériences. Tech. rep., Appui santé & médico social.
- ANAP, 2016c. Améliorer la gestion des transports de personnes handicapées. tome 2: Modalités organisationnelles. Tech. rep., Appui santé & médico social.
- Archetti, C., Speranza, M. G., 2014. A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization* 2 (4), 223–246.
- Beaudry, A., Laporte, G., Melo, T., Nickel, S., 2010. Dynamic transportation of patients in hospitals. *OR spectrum* 32 (1), 77–107.
- Boland, N., Hewitt, M., Marshall, L., Savelsbergh, M., 2017. The continuous-time service network design problem. *Operations Research* 65 (5), 1303–1321.
- Borndörfer, R., Grötschel, M., Klostermeier, F., Küttner, C., 1999. Telebus berlin: Vehicle scheduling in a dial-a-ride system. In: *Computer-Aided Transit Scheduling*. Springer, pp. 391–422.

- Braekers, K., Caris, A., Janssens, G. K., 2014. Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Research Part B: Methodological* 67, 166–186.
- Braekers, K., Kovacs, A. A., Dec. 2016. A multi-period dial-a-ride problem with driver consistency. *Transportation Research Part B: Methodological* 94, 355–377.
- Bélangier, V., Ruiz, A., Soriano, P., 2019. Recent optimization models and trends in location, relocation, and dispatching of emergency medical vehicles. *European Journal of Operational Research* 272 (1), 1–23.
- Chakraborty, S., 2018. Enablers of co-creation in hospital-supplier relationships: empirical study in indian healthcare context. In: *Supply Chain Forum: An International Journal*. Vol. 19. Taylor & Francis, pp. 331–352.
- Chassaing, M., Duhamel, C., Lacomme, P., 2016. An ELS-based approach with dynamic probabilities management in local search for the Dial-A-Ride Problem. *Engineering Applications of Artificial Intelligence* 48, 119–133.
- Christiaens, J., Vanden Berghe, G., 2016. A fresh ruin & recreate implementation for the capacitated vehicle routing problem. Technical report, KU Leuven, Department of Computer Science.
- Coppi, A., Detti, P., Raffaelli, J., 2013. A planning and routing model for patient transportation in health care. *Electronic notes in discrete mathematics* 41, 125–132.
- Cordeau, J.-F., Laporte, G., 2003. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological* 37 (6), 579–594.
- Cordeau, J.-F., Laporte, G., Potvin, J.-Y., Savelsbergh, M. W., 2007. *Transportation on demand*. *Handbooks in operations research and management science* 14, 429–466.
- Derigs, U., Gottlieb, J., Kalkoff, J., Piesche, M., Rothlauf, F., Vogel, U., 2011. Vehicle routing with compartments: applications, modelling and heuristics. *OR spectrum* 33 (4), 885–914.
- Detti, P., Papalini, F., de Lara, G. Z. M., 2017. A multi-depot dial-a-ride problem with heterogeneous vehicles and compatibility constraints in healthcare. *Omega* 70, 1–14.
- Dick, W. F., 2003. Anglo-american vs. franco-german emergency medical services system. *Prehospital and Disaster Medicine* 18 (1), 29–37.
- Dikas, G., Minis, I., 2014. Scheduled paratransit transport systems. *Transportation Research Part B: Methodological* 67, 18 – 34.

- Doerner, K., Focke, A., Gutjahr, W. J., 2007. Multicriteria tour planning for mobile healthcare facilities in a developing country. *European Journal of Operational Research* 179 (3), 1078–1096.
- Doerner, K. F., Salazar-González, J.-J., 2014. Pickup-and-delivery problems for people transportation. In: Vigo, D., Toth, P. (Eds.), *Vehicle Routing: Problems, Methods, and Applications*. Vol. 18. SIAM, Ch. 7, pp. 193–212.
- Drexel, M., Aug. 2012. Synchronization in Vehicle Routing—A Survey of VRPs with Multiple Synchronization Constraints. *Transportation Science* 46 (3), 297–316.
- Dueck, G., 1993. New optimization heuristics. *Journal of Computational Physics* 104 (1), 86 – 92.
- Erera, A. L., Savelsbergh, M., Uyar, E., Dec. 2009. Fixed routes with backup vehicles for stochastic vehicle routing problems with time constraints. *Networks* 54 (4), 270–283.
- Feillet, D., Garaix, T., Lehuédé, F., Péton, O., Quadri, D., May 2014. A new consistent vehicle routing problem for the transportation of people with disabilities. *Networks* 63 (3), 211–224.
- Fikar, C., Hirsch, P., 2017. Home health care routing and scheduling: A review. *Computers & Operations Research* 77, 86–95.
- Firat, M., Woeginger, G. J., 2011. Analysis of the dial-a-ride problem of Hunsaker and Savelsbergh. *Operations Research Letters* 39 (1), 32–35.
- Fleischmann, B., Borgwardt, K.-H., Klein, R., Tuma, A., 2009. *Operations Research Proceedings 2008: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR) University of Augsburg, September 3-5, 2008*. Springer Science & Business Media.
- Foster, B. A., Ryan, D. M., 1976. An integer programming approach to the vehicle scheduling problem. *Journal of the Operational Research Society* 27 (2), 367–384.
- Fu, L., 1999. Improving paratransit scheduling by accounting for dynamic and stochastic variations in travel time. *Transportation Research Record* 1666 (1), 74–81.
- Fu, L., 2002. Scheduling dial-a-ride paratransit under time-varying, stochastic congestion. *Transportation Research Part B: Methodological* 36 (6), 485–506.
- Fu, L., Ishkhanov, G., 2004. Fleet size and mix optimization for paratransit services. *Transportation Research Record* 1884 (1), 39–46.
- Fu, L., Teply, S., 1999. On-line and off-line routing and scheduling of dial-a-ride paratransit vehicles. *Computer-Aided Civil and Infrastructure Engineering* 14 (5), 309–319.

- Golden, B., Assad, A., Levy, L., Gheysens, F., 1984. The fleet size and mix vehicle routing problem. *Computers & Operations Research* 11 (1), 49–66.
- Gonzalez, L., Roussel, R., Héam, J.-C., Mikou, M., Ferretti, C., 2018. Les dépenses de santé en 2017 - résultats des comptes de la santé - Édition 2018. Tech. rep., Ministère des solidarités et de la santé.
- Grangier, P., Gendreau, M., Lehuédé, F., Rousseau, L.-M., 2017. A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking. *Computers & Operations Research* 84, 116–126.
- Groër, C., Golden, B., Wasil, E., Oct. 2009. The Consistent Vehicle Routing Problem. *Manufacturing & Service Operations Management* 11 (4), 630–643.
- Gschwind, T., Drexl, M., 2016. Adaptive Large Neighborhood Search with a Constant-Time Feasibility Test for the Dial-a-Ride Problem. LM-2016-08, Gutenberg School of Management and Economics, Johannes Gutenberg University.
- Gschwind, Timo, 2015. Route Feasibility Testing and Forward Time Slack for the Synchronized Pickup and Delivery Problem. Tech. Rep. 1503, Gutenberg School of Management and Economics, Johannes Gutenberg University.
- Hains, I. M., Marks, A., Georgiou, A., Westbrook, J. I., 2011. Non-emergency patient transport: what are the quality and safety issues? a systematic review. *International Journal for Quality in Health Care* 23 (1), 68–75.
- Hanne, T., Melo, T., Nickel, S., 2009. Bringing robustness to patient flow management through optimized patient transports in hospitals. *Interfaces* 39 (3), 241–255.
- Henke, T., Speranza, M. G., Wäscher, G., 2015. The multi-compartment vehicle routing problem with flexible compartment sizes. *European Journal of Operational Research* 246 (3), 730–743.
- Ho, S. C., Szeto, W., Kuo, Y.-H., Leung, J. M., Petering, M., Tou, T. W., 2018. A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part B: Methodological* 111, 395–421.
- Huggins, C., Shugg, D., 2015. Non-emergency patient transport in victoria: An overview. *Australasian Journal of Paramedicine* 6 (4).
- Hunsaker, B., Savelsbergh, M., 2002. Efficient feasibility testing for dial-a-ride problems. *Operations Research Letters* 30 (3), 169–173.
- Jaw, J.-J., Odoni, A. R., Psaraftis, H. N., Wilson, N. H., 1986. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. *Transportation Research Part B: Methodological* 20 (3), 243–257.

- Kelle, P., Woosley, J., Schneider, H., 2012. Pharmaceutical supply chain specifics and inventory solutions for a hospital case. *Operations Research for Health Care* 1 (2-3), 54–63.
- Koç, c., Bektas, T., Jabali, O., Laporte, G., 2016. Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research* 249 (1), 1–21.
- Koch, H., Henke, T., Wäscher, G., 2016. A genetic algorithm for the multi-compartment vehicle routing problem with flexible compartment sizes. Tech. rep., Otto-von-Guericke-Universität Magdeburg, Faculty of Economics and Management.
- Kovacs, A. A., Golden, B. L., Hartl, R. F., Parragh, S. N., Oct. 2014a. Vehicle routing problems in which consistency considerations are important: A survey. *Networks* 64 (3), 192–213.
- Kovacs, A. A., Golden, B. L., Hartl, R. F., Parragh, S. N., Nov. 2015a. The Generalized Consistent Vehicle Routing Problem. *Transportation Science* 49 (4), 796–816.
- Kovacs, A. A., Parragh, S. N., Hartl, R. F., Jan. 2014b. A template-based adaptive large neighborhood search for the consistent vehicle routing problem. *Networks* 63 (1), 60–81.
- Kovacs, A. A., Parragh, S. N., Hartl, R. F., Dec. 2015b. The multi-objective generalized consistent vehicle routing problem. *European Journal of Operational Research* 247 (2), 441–458.
- Landry, S., Philippe, R., 2004. Supply chain forum: An international journal. In: *How logistics can service healthcare*. Vol. 5. Taylor & Francis, pp. 24–30.
- Laurent, V., Frédéric, V., Agnès, J., Anne, A., Claire-Marie, F.-G., 2012. Établissements et services pour personnes handicapées offre et besoins, modalités de financement. Tech. rep., Inspection Générale des Finances & Inspection Générale des Affaires Sociales.
- Lehuédé, F., Masson, R., Parragh, S. N., Péton, O., Tricoire, F., 2014. A multi-criteria large neighbourhood search for the transportation of disabled people. *Journal of the Operational Research Society* 65 (7), 983–1000.
- Lei, H., Laporte, G., Guo, B., 2012. Districting for routing with stochastic customers. *EURO Journal on Transportation and Logistics* 1 (1-2), 67–85.
- Luo, Z., Qin, H., Che, C., Lim, A., Jun. 2015. On service consistency in multi-period vehicle routing. *European Journal of Operational Research* 243 (3), 731–744.
- Luxen, D., Vetter, C., 2011. Real-time routing with openstreetmap data. In: *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. GIS'11. ACM, New York, NY, USA, pp. 513–516.

- Madsen, O. B., Ravn, H. F., Rygaard, J. M., 1995. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of operations Research* 60 (1), 193–208.
- Mahmoudi, M., Zhou, X., Jul. 2016. Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state–space–time network representations. *Transportation Research Part B: Methodological* 89, 19–42.
- Masson, R., Lehuédé, F., Péton, O., 2014. The Dial-A-Ride Problem with Transfers. *Computers & Operations Research* 41, 12–23.
- Milburn, A. B., Spicer, J., 2013. Multi-objective home health nurse routing with remote monitoring devices. *International Journal of Planning and Scheduling* 1 (4), 242–263.
- Mo, D. Y., Wang, Y., Lee, Y., Tseng, M. M., 2018. Mass customizing paratransit services with a ridesharing option. *IEEE Transactions on Engineering Management*.
- Molenbruch, Y., Braekers, K., Caris, A., Berghe, G. V., 2017. Multi-directional local search for a bi-objective dial-a-ride problem in patient transportation. *Computers & Operations Research* 77, 58–71.
- Ogryczak, W., Luss, H., Pióro, M., Nace, D., Tomaszewski, A., 2014. Fair optimization and networks: A survey. *Journal of Applied Mathematics* 2014.
- Paquette, J., Bellavance, F., Cordeau, J.-F., Laporte, G., 2012. Measuring quality of service in dial-a-ride operations: the case of a canadian city. *Transportation* 39 (3), 539–564.
- Paquette, J., Cordeau, J.-F., Laporte, G., Pascoal, M. M., 2013. Combining multicriteria analysis and tabu search for dial-a-ride problems. *Transportation Research Part B: Methodological* 52, 1–16.
- Parragh, S. N., 2011. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. *Transportation Research Part C: Emerging Technologies* 19 (5), 912–930.
- Parragh, S. N., Cordeau, J.-F., Doerner, K. F., Hartl, R. F., 2012. Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints. *OR spectrum* 34 (3), 593–633.
- Parragh, S. N., Doerner, K. F., Hartl, R. F., 2008. A survey on pickup and delivery problems. *Journal für Betriebswirtschaft* 58 (2), 81–117.
- Parragh, S. N., Doerner, K. F., Hartl, R. F., 2010. Variable neighborhood search for the dial-a-ride problem. *Computers & Operations Research* 37 (6), 1129–1138.

- Parragh, S. N., Schmid, V., 2013. Hybrid column generation and large neighborhood search for the dial-a-ride problem. *Computers & Operations Research* 40 (1), 490–497.
- Pillac, V., Gendreau, M., Guéret, C., Medaglia, A. L., 2013. A review of dynamic vehicle routing problems. *European Journal of Operational Research* 225 (1), 1 – 11.
- Pisinger, D., Ropke, S., 2007. A general heuristic for vehicle routing problems. *Computers & Operations Research* 34 (8), 2403–2435.
- Prescott-Gagnon, E., Desaulniers, G., Rousseau, L.-M., 2009. A branch-and-price-based large neighborhood search algorithm for the vehicle routing problem with time windows. *Networks: An International Journal* 54 (4), 190–204.
- Qu, Y., Bard, J. F., 2013. The heterogeneous pickup and delivery problem with configurable vehicle capacity. *Transportation Research Part C: Emerging Technologies* 32, 1–20.
- Qu, Y., Bard, J. F., 2015. A Branch-and-Price-and-Cut Algorithm for Heterogeneous Pickup and Delivery Problems with Configurable Vehicle Capacity. *Transportation Science* 49 (2), 254–270.
- Rasmussen, M. S., Justesen, T., Dohn, A., Larsen, J., 2012. The home care crew scheduling problem: Preference-based visit clustering and temporal dependencies. *European Journal of Operational Research* 219 (3), 598–610.
- Rekiek, B., Delchambre, A., Saleh, H. A., Aug. 2006. Handicapped Person Transportation: An application of the Grouping Genetic Algorithm. *Engineering Applications of Artificial Intelligence* 19 (5), 511–520.
- Ritzinger, U., Puchinger, J., Hartl, R. F., 2016. A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research* 54 (1), 215–231.
- Robinson, V., Goel, V., MacDonald, R., Manuel, D., 2009. Inter-facility patient transfers in ontario: Do you know what your local ambulance is being used for? *Healthcare Policy | Politiques de Santé* 4 (3), 53–66.
- Rochat, Y., Taillard, É. D., 1995. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of heuristics* 1 (1), 147–167.
- Ropke, S., Cordeau, J.-F., 2009. Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science* 43 (3), 267–286.
- Ropke, S., Cordeau, J.-F., Laporte, G., 2007. Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks* 49 (4), 258–272.

- Ropke, S., Pisinger, D., Nov. 2006. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science* 40 (4), 455–472.
- Rouquet, A., Vauché, L., 2015. A typology of logistics pooling in supply chains. In: *Supply Chain Forum: An International Journal*. Vol. 16. Taylor & Francis, pp. 2–12.
- Ruland, K. S., 1999. A model for aeromedical routing and scheduling. *International Transactions in Operational Research* 6 (1), 57–73.
- Saghafian, S., Austin, G., Traub, S. J., 2015. Operations research/management contributions to emergency department patient flow optimization: Review and research prospects. *IIE Transactions on Healthcare Systems Engineering* 5 (2), 101–123.
- Santini, A., Ropke, S., Hvattum, L. M., 2016. A comparison of acceptance criteria for the adaptive large neighbourhood search metaheuristic. working paper.
- Schilde, M., Doerner, K. F., Hartl, R. F., 2011. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. *Computers & operations research* 38 (12), 1719–1730.
- Schneider, M., Stenger, A., Schwahn, F., Vigo, D., Nov. 2015. Territory-Based Vehicle Routing in the Presence of Time-Window Constraints. *Transportation Science* 49 (4), 732–751.
- Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., Dueck, G., 2000. Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics* 159 (2), 139–171.
- Shaw, P., 1998. Using constraint programming and local search methods to solve vehicle routing problems. In: *International Conference on Principles and Practice of Constraint Programming*. Springer, pp. 417–431.
- Smilowitz, K., Nowak, M., Jiang, T., May 2013. Workforce Management in Periodic Delivery Operations. *Transportation Science* 47 (2), 214–230.
- Spliet, R., Desaulniers, G., 2015. The discrete time window assignment vehicle routing problem. *European Journal of Operational Research* 244 (2), 379–391.
- Spliet, R., Gabor, A. F., Nov. 2015. The Time Window Assignment Vehicle Routing Problem. *Transportation Science* 49 (4), 721–731.
- Spliet, R., Gabor, A. F., Dekker, R., 2014. The vehicle rescheduling problem. *Computers & Operations Research* 43, 129–136.
- Subramanian, A., Uchoa, E., Ochi, L. S., 2013. A hybrid algorithm for a class of vehicle routing problems. *Computers & Operations Research* 40 (10), 2519 – 2531.

- Subramanyam, A., Gounaris, C. E., 2016. A branch-and-cut framework for the consistent traveling salesman problem. *European Journal of Operational Research* 248 (2), 384–395.
- Sungur, I., Ren, Y., Ordóñez, F., Dessouky, M., Zhong, H., May 2010. A Model and Algorithm for the Courier Delivery Problem with Uncertainty. *Transportation Science* 44 (2), 193–205.
- Taillard, E. D., 1999. A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO-Operations Research* 33 (1), 1–14.
- Tang, J., Kong, Y., Lau, H., Ip, A. W. H., 2010. A note on “Efficient feasibility testing for dial-a-ride problems”. *Operations Research Letters* 38 (5), 405–407.
- Tarantilis, C., Stavropoulou, F., Repoussis, P., Mar. 2012. A template-based Tabu Search algorithm for the Consistent Vehicle Routing Problem. *Expert Systems with Applications* 39 (4), 4233–4239.
- Tellez, O., Daguet, L., Lehuédé, F., Monteiro, T., Montoya, G. O., Péton, O., Vercraene, S., 27-29 August 2018a. Étude pour l’analyse et l’optimisation du transport des personnes en situation de handicap. In: *Conférence Francophone en Gestion et Ingénierie des Systèmes Hospitaliers (GISEH)*. Geneva, Switzerland.
- Tellez, O., Daguet, L., Lehuédé, F., Monteiro, T., Montoya, G. O., Péton, O., Vercraene, S., 2019. A stakeholder oriented approach to the optimization of transports of people with disabilities. *Supply Chain Forum: An International Journal* (Submitted).
- Tellez, O., Vercraene, S., Lehuédé, F., Péton, O., Monteiro, T., 2018b. The fleet size and mix dial-a-ride problem with reconfigurable vehicle capacity. *Transportation Research Part C: Emerging Technologies* 91, 99 – 123.
- Toth, P., Vigo, D., 1997. Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science* 31 (1), 60–71.
- Turan, B., Schmid, V., Doerner, K. F., 2011. Models for intra-hospital patient routing. In: *3rd IEEE International Symposium on Logistics and Industrial Informatics*. IEEE, pp. 51–60.
- ur Rahman, S., Smith, D. K., 2000. Use of location-allocation models in health service development planning in developing nations. *European Journal of Operational Research* 123 (3), 437 – 452.
- Uthayakumar, R., Priyan, S., 2013. Pharmaceutical supply chain and inventory management strategies: Optimization for a pharmaceutical company and a hospital. *Operations Research for Health Care* 2 (3), 52–64.

- Wahl, T., Lesteven, P., Paul-Emmanuel, G., Eve, R., 2016. Les transports sanitaires - revue de dépenses 2016. Tech. rep., Inspection Générale des Finances & Inspection Générale des Affaires Sociales.
- Xiang, Z., Chu, C., Chen, H., 2006a. A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints. *European Journal of Operational Research* 174 (2), 1117 – 1139.
- Xiang, Z., Chu, C., Chen, H., Oct. 2006b. A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints. *European Journal of Operational Research* 174 (2), 1117–1139.
- Xiang, Z., Chu, C., Chen, H., 2008. The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments. *European Journal of Operational Research* 185 (2), 534 – 551.
- Xu, Z., Cai, Y., Dec. 2018. Variable neighborhood search for consistent vehicle routing problem. *Expert Systems with Applications* 113, 66–76.
- Zhang, Z., Liu, M., Lim, A., 2015. A memetic algorithm for the patient transportation problem. *Omega* 54, 60–71.
- Zhong, H., Hall, R. W., Dessouky, M., Feb. 2007. Territory Planning and Vehicle Dispatching with Driver Learning. *Transportation Science* 41 (1), 74–89.
- Zhu, Q., Johnson, S., Sarkis, J., 2018. Lean six sigma and environmental sustainability: a hospital perspective. In: *Supply Chain Forum: An International Journal*. Vol. 19. Taylor & Francis, pp. 25–41.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., Da Fonseca Grunert, V., 2002. Performance assessment of multiobjective optimizers: An analysis and review. *TIK-Report* 139.



FOLIO ADMINISTRATIF

THESE DE L'UNIVERSITE DE LYON OPEREE AU SEIN DE L'INSA LYON

NOM : TELLEZ SANCHEZ

DATE de SOUTENANCE : 4/10/2019

(avec précision du nom de jeune fille, le cas échéant)

Prénoms : Oscar Augusto

TITRE : Optimisation du transport quotidien des personnes en situation de handicap

NATURE : Doctorat

Numéro d'ordre : AAAALYSEIXXXX

Ecole doctorale : École Doctorale Informatique et Mathématiques (N° ED512)

Spécialité : Génie Industriel

RESUME :

En France, le coût du transport médico-social a été estimé à 400 millions d'euros en 2012 et augmente à un taux annuel de 8% depuis. Pour un établissement médico-social (ESMS), ce coût représente souvent la deuxième dépense en importance après celle du personnel.

Dans ce contexte, cette thèse appartenant au projet «Numérique et Optimisation pour une Mobilité Adaptée» (NOMAd) vise à améliorer le transport quotidien des personnes handicapées entre leur domicile et les ESMS. Pour cela, nous proposons la mutualisation du transport entre plusieurs ESMS ainsi qu'une stratégie d'optimisation globale de la planification annuelle de transport.

La construction de la planification annuelle des transports est un problème complexe encore non complètement résolu dans la littérature scientifique. Conformément à la pratique actuelle, nous supposons que le plan de transport annuel est composé d'horaires hebdomadaires identiques divisés en demi-journées.

Le problème d'optimisation sur une demi-journée, appelé "fleet size and mix dial-a-ride problem" considère plusieurs types de passagers et un parc hétérogène de véhicules reconfigurables qui peuvent être modifiés en route. La matheuristique LNS-RSCP est développée pour ce problème.

L'optimisation du problème hebdomadaire, nommé time-consistent DARP, comprend simultanément des exigences de régularité de service et des objectifs de coût de transport. La matheuristique MSPP est développée pour ce problème.

Actuellement, la gestion du transport est principalement réalisée intra-établissements, sans mutualisation. La mutualisation du transport, en plus de l'augmentation de la complexité du problème, suppose d'avoir un intérêt économique et un mécanisme commun de gestion et communication efficace entre les parties. Au niveau stratégique cette étude vise à évaluer l'impact de la mutualisation sur les coûts et temps de trajet. Opérationnellement, ce problème correspond au problème de clustering pour la résolution des problèmes de grande taille.

MOTS-CLÉS : Optimisation, Recherche opérationnelle, transport, personnes en situation de handicap, mutualisation

Laboratoire (s) de recherche : Laboratoire DISP

Directeur de thèse: Thibaud MONTEIRO

Président de jury : Maria DI MASCOLO

Composition du jury :

NORRE, Sylvie

Professeur Université Clermont-Auvergne

Rapporteur

JUSSIEN GUERET, Christelle

Professeur Université d'Angers

Rapporteur

DI MASCOLO, Maria

Directrice de recherche Laboratoire G-SCOP

Examinatrice

GARAIX, Thierry

Maître Assistant École des Mines St. Étienne

Examineur

MONTEIRO, Thibaud

Professeur INSA Lyon

Directeur

VERCRAENE, Samuel

Maître de conférences INSA Lyon

Co-directeur

LEHUÉDÉ, Fabien

Professeur IMT Atlantique

Co-directeur

PÉTON, Olivier

Professeur IMT Atlantique

Co-directeur