Planification proactive et réactive des réseaux d'ambulances.

Mohammed Skiredj

▶ **To cite this version:**

Mohammed Skiredj. Planification proactive et réactive des réseaux d'ambulances.. Autre. Université de Lyon, 2021. Français. NNT : 2021LYSEM022 . tel-03888192

HAL Id: tel-03888192
https://theses.hal.science/tel-03888192

Submitted on 7 Dec 2022

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.

N°d'ordre NNT : 2021LYSEM022

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
l'Ecole des Mines de Saint-Etienne

Ecole Doctorale N° 488
Sciences, Ingénierie, Santé

Spécialité de doctorat : Génie industriel
Discipline : Recherche opérationnelle

Soutenue publiquement/à huis clos le 29/06/2021, par :
Mohammed SKIREDJ

Planification proactive et réactive des réseaux d'ambulances

Devant le jury composé de :

Mme SANTOS Andréa Cynthia, Professeur, Université Le Havre Normandie, Rapporteur
Mme LABADI Nacima, Professeur, Université de technologie de Troyes, Rapporteur
Mr SEVAUX Marc, Professeur, Université de Bretagne-Sud, Examinateur
Mr KERGOSIEN Yannick, Maître de conférences(HDR), Université de Tours, Examinateur
Mr Xie Xiaolan, Professeur, Mines Saint Etienne, Directeur de thèse
Mr GARAIX Thierry, Maître de conférences, Mines Saint Etienne, Co-encadrant

Mr ISSART Romain, Directeur des opérations, LOMACO, Invité
Mr VENZAC Jérôme, Chef de projet, LOMACO, Invité
Mr MAZIER Alexandre, Consultant(PhD), Apis Consulting, Invité

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

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

ABSI	Nabil	MR	Génie industriel	CMP
AUGUSTO	Vincent	CR	Image, Vision, Signal	CIS
AVRIL	Stéphane	PR2	Mécanique et ingénierie	CIS
BADEL	Pierre	MA(MDC)	Mécanique et ingénierie	CIS
BALBO	Flavien	PR2	Informatique	FAYOL
BASSEREAU	Jean-François	PR	Sciences et génie des matériaux	SMS
BATTON-HUBERT	Mireille	PR2	Sciences et génie de l'environnement	FAYOL
BEIGBEDER	Michel	MA(MDC)	Informatique	FAYOL
BLAYAC	Sylvain	MA(MDC)	Microélectronique	CMP
BOISSIER	Olivier	PR1	Informatique	FAYOL
BONNEFOY	Olivier	PR	Génie des Procédés	SPIN
BORBELY	Andras	MR(DR2)	Sciences et génie des matériaux	SMS
BOUCHER	Xavier	PR2	Génie Industriel	FAYOL
BRODHAG	Christian	DR	Sciences et génie de l'environnement	FAYOL
BRUCHON	Julien	MA(MDC)	Mécanique et ingénierie	SMS
CAMEIRAO	Ana	MA(MDC)	Génie des Procédés	SPIN
CHRISTIEN	Frédéric	PR	Science et génie des matériaux	SMS
DAUZERE-PERES	Stéphane	PR1	Génie Industriel	CMP
DEBAYLE	Johan	MR	Sciences des Images et des Formes	SPIN
DEGEORGE	Jean-Michel	MA(MDC)	Génie industriel	Fayol
DELAFOSSE	David	PR0	Sciences et génie des matériaux	SMS
DELORME	Xavier	MA(MDC)	Génie industriel	FAYOL
DESRAYAUD	Christophe	PR1	Mécanique et ingénierie	SMS
DJENIZIAN	Thierry	PR	Science et génie des matériaux	CMP
BERGER-DOUCE	Sandrine	PR1	Sciences de gestion	FAYOL
DRAPIER	Sylvain	PR1	Mécanique et ingénierie	SMS
DUTERTRE	Jean-Max	MA(MDC)		CMP
EL MRABET	Nadia	MA(MDC)		CMP
FAUCHEU	Jenny	MA(MDC)	Sciences et génie des matériaux	SMS
FAVERGEON	Loïc	CR	Génie des Procédés	SPIN
FEILLET	Dominique	PR1	Génie Industriel	CMP
FOREST	Valérie	MA(MDC)	Génie des Procédés	CIS
FRACZKIEWICZ	Anna	DR	Sciences et génie des matériaux	SMS
GARCIA	Daniel	MR(DR2)	Sciences de la Terre	SPIN
GAVET	Yann	MA(MDC)	Sciences des Images et des Formes	SPIN
GERINGER	Jean	MA(MDC)	Sciences et génie des matériaux	CIS
GOEURIOT	Dominique	DR	Sciences et génie des matériaux	SMS
GONDRAN	Natacha	MA(MDC)	Sciences et génie de l'environnement	FAYOL
GONZALEZ FELIU	Jesus	MA(MDC)	Sciences économiques	FAYOL
GRAILLOT	Didier	DR	Sciences et génie de l'environnement	SPIN
GROSSEAU	Philippe	DR	Génie des Procédés	SPIN
GRUY	Frédéric	PR1	Génie des Procédés	SPIN
HAN	Woo-Suck	MR	Mécanique et ingénierie	SMS
HERRI	Jean Michel	PR1	Génie des Procédés	SPIN
KERMOUCHE	Guillaume	PR2	Mécanique et Ingénierie	SMS
KLOCKER	Helmut	DR	Sciences et génie des matériaux	SMS
LAFOREST	Valérie	MR(DR2)	Sciences et génie de l'environnement	FAYOL
LERICHE	Rodolphe	CR	Mécanique et ingénierie	FAYOL
MALLIARAS	Georges	PR1	Microélectronique	CMP
MOLIMARD	Jérôme	PR2	Mécanique et ingénierie	CIS
MOUTTE	Jacques	CR	Génie des Procédés	SPIN
NAVARRO	Laurent	CR		CIS
NEUBERT	Gilles			FAYOL
NIKOLOVSKI	Jean-Pierre	Ingénieur de recherche	Mécanique et ingénierie	CMP
NORTIER	Patrice	PR1	Génie des Procédés	SPIN
O CONNOR	Rodney Philip	MA(MDC)	Microélectronique	CMP
PICARD	Gauthier	MA(MDC)	Informatique	FAYOL
PINOLI	Jean Charles	PR0	Sciences des Images et des Formes	SPIN
POURCHEZ	Jérémy	MR	Génie des Procédés	CIS
ROUSSY	Agnès	MA(MDC)	Microélectronique	CMP
ROUSTANT	Olivier	MA(MDC)	Mathématiques appliquées	FAYOL
SANAUR	Sébastien	MA(MDC)	Microélectronique	CMP
SERRIS	Eric	IRD		FAYOL
STOLARZ	Jacques	CR	Sciences et génie des matériaux	SMS
TRIA	Assia	Ingénieur de recherche	Microélectronique	CMP
VALDIVIESO	François	PR2	Sciences et génie des matériaux	SMS
VIRICELLE	Jean Paul	DR	Génie des Procédés	SPIN
WOLSKI	Krzystof	DR	Sciences et génie des matériaux	SMS
XIE	Xiaolan	PR0	Génie industriel	CIS
YUGMA	Gallian	CR	Génie industriel	CMP

Mise à jour : 30/08/2018

Document édité le : 11 octobre 2021

Remerciements

La rédaction de cette thèse est l'aboutissement de nombreux défis et marque la fin d'un épisode et le début d'un nouvel chapitre dans ma vie. Ce travail de longue haleine n'aurait jamais pu être achevé sans l'aide précieuse de plusieurs personnes qui ont contribué, chacune à sa façon, à bien mener ce projet.

D'abord, j'adresse tous mes remerciements à tous les organismes impliqués dans ce projet et qui ont permis à ce travail de voir le jour (Lomaco, Fidéciel, ANRT, Ecole Nationale Supérieure des Mines de Saint-Etienne et LIMOS). Ensuite, j'exprime ma gratitude à mes encadrants pour leur présence, leur soutien et leur support indéfectible. Je tiens à remercier aussi mon directeur de thèse Professeur Xiaolan Xie pour son appui, son professionnalisme et pour les nombreux encouragements qu'il m'a prodigué. Je voudrais exprimer mes remerciements les plus vifs et les plus sincères à Thierry Garaix pour sa disponibilité permanente, sa générosité d'esprit et son aide précieuse. Mes remerciements vont aussi à Jérôme Venzac et Serge Rouzières qui m'ont aidé et encadré dans la partie industrielle des algorithmes développés et qui n'ont pas hésité à partager avec moi les ficelles du métier du transport sanitaire français. Je remercie également tout le personnel Lomaco et tout particulièrement le directeur monsieur Philippe Issart pour sa confiance, et son intérêt porté au projet. Il m'est impossible d'oublier de remercier Alexandre Mazier pour son aide précieuse dans le suivi des travaux et la coordination entre l'école des Mines et l'entreprise Lomaco du début jusqu'à la fin de cette thèse.

Je remercie également madame Andrea Cyntia Santos Duhamel et madame Nacima Labadie d'avoir accepté d'être les rapporteurs de ce travail, monsieur Marc Sevaux et monsieur Yannick Kergosien qui m'ont fait l'honneur d'être examinateurs de ma thèse. Il me sera très difficile de remercier tout ceux qui ont collaboré de près ou de loin dans la réalisation de ce travail.

D'un autre coté, je remercie particulièrement mes parents, mon épouse, mon frère et mes sœurs pour leur amour infini et inconditionnel. Je remercie aussi mes amis, à la fois ceux de longue date et ceux rencontrés pendant ces années de thèse et qui ont aussi à leur manière rendu ce travail possible.

J'espère que ce travail sera utile et servira à démêler une problématique à la fois complexe et critique.

Table des matières

Table des matières v

Table des figures vii

Liste des tableaux x

1 Introduction Générale 1
- 1.1 Introduction . . . 2
- 1.2 Transport sanitaire . . . 3
  - 1.2.1 Définition du transport sanitaire . . . 3
  - 1.2.2 Les organismes impliqués dans le transport sanitaire . . . 3
  - 1.2.3 La croissance du secteur de transport sanitaire et ses enjeux . . . 4
  - 1.2.4 La réglementation . . . 5
  - 1.2.5 Difficulté du métier . . . 9
- 1.3 La société Lomaco . . . 9
  - 1.3.1 Présentation de la société Lomaco . . . 9
  - 1.3.2 Lomaco en chiffres . . . 10
  - 1.3.3 Les objectifs de Lomaco . . . 10
- 1.4 Processus de planification . . . 11
  - 1.4.1 Régulation des demandes de transports des patients . . . 11
- 1.5 Problèmes de décision . . . 14
  - 1.5.1 La gestion de temps des employés . . . 14
  - 1.5.2 La planification de la veille pour le lendemain (partie proactive) . 14
  - 1.5.3 La régulation dynamique de la flotte de véhicules (partie réactive) 15
- 1.6 Organisation de la Thèse . . . 15

2 Planification proactive des réseaux d'ambulance 17
- 2.1 Introduction . . . 19
- 2.2 Literature review . . . 21
  - 2.2.1 Scientific position of the crewdarp . . . 22

2.2.2 A detailed review of a close problem – the rdarp . . . 23
2.3 Problem description and modeling . . . 24
2.3.1 Underlying graph . . . 24
2.3.2 Drivers and depots . . . 25
2.3.3 Vehicles . . . 26
2.3.4 Requests . . . 27
2.3.5 Objective function . . . 28
2.4 Neighborhood evaluation . . . 29
2.4.1 Request removal and insertion in single trips . . . 29
2.4.2 Request insertion and removal in synchronized multitrips . . . 33
2.5 Adaptive large neighborhood search . . . 35
2.5.1 Destruction operators . . . 35
2.5.2 Insertion operators . . . 37
2.6 Constructive heuristics . . . 38
2.6.1 hla – Human-Like Algorithm . . . 38
2.6.2 hla+ – Improved constructive heuristic . . . 39
2.7 Real-world data set . . . 40
2.8 Numerical experiments . . . 41
2.8.1 ALNS implementation . . . 41
2.8.2 CrewDARP . . . 43
2.8.3 Benchmark on related problems . . . 45
2.9 Conclusion . . . 47

3 Pilotage réactif d'une flotte d'ambulances 61
3.1 Introduction . . . 62
3.2 État de l'art . . . 67
3.2.1 Types d'approches . . . 67
3.2.2 Synthèse . . . 70
3.3 Description générale du problème . . . 71
3.3.1 Régulation du transport sanitaire . . . 71
3.3.2 Différentes approches de régulations . . . 74
3.4 Définition du Problème de Transport Médicalisé Dynamique (DMTP) . . . 76
3.4.1 Réseau de transport . . . 76
3.4.2 Chauffeurs et dépôts . . . 77
3.4.3 Véhicule . . . 78
3.4.4 Demande . . . 79
3.4.5 Objectifs et contraintes . . . 79
3.4.6 Hypothèses et simplifications par rapport au problème réel . . . 81
3.5 Problèmes intermédiaires considérés . . . 82
3.5.1 Problème simplifié P_1 – **DARP** . . . 82

3.5.2 Problème P_2 – **DARP avec temps de travail** 82
3.5.3 Problème P_3 – **DARP avec pauses** 82
3.5.4 Problème P_4 – **DARP avec plusieurs types de véhicules** . . . 83
3.6 Stratégies de planification 83
3.6.1 Fonctionnement des stratégies et pré-traitements communs 84
3.6.2 Stratégie basée sur les activités S_1 88
3.6.3 Stratégie basée sur les ressources puis les activités S_2 91
3.6.4 Stratégie basée sur les activités puis les ressources S_3 92
3.7 Expérimentation numérique 93
3.7.1 Plateforme de tests 93
3.7.2 Génération de données 96
3.7.3 Résultats numériques 98
3.8 Conclusion 104

4 Implémentation industrielle **121**
4.1 Introduction 122
4.2 Lomaco Online (LOL) 122
4.2.1 Fonctionnalités 123
4.2.2 Interface Régulation / Planning 124
4.3 Intégration des demandes des clients 127
4.3.1 Gestion des différents points d'embauche et de débauche 127
4.3.2 Critère tarifaire des choix de type de véhicule 128
4.3.3 Minimiser le nombre de ressources utilisées dans le planning de la veille pour le lendemain 129
4.3.4 Donner la priorité aux véhicules fortement kilométrés 129
4.4 Conclusion 130

5 Synthèse et Conclusion **135**

Bibliographie **138**

A Lunch break counterexample **143**

B Results tables **145**

Table des figures

1.1 La répartition du nombre de transport par type de véhicule (2018). . . . 4
1.2 Interaction entre les différents acteurs du transport sanitaire ; Source : [Santé-Mobilité-Services, 2017] 5
1.3 La société Lomaco en chiffres 11

2.1 Request deletion and insertion with sequence concatenation 30
2.2 Comparison of two temperature control management strategies 42

3.1 Approches d'optimisation 69
3.2 Coût associé aux heures supplémentaires 78
3.3 Exemple plan prévu 84
3.4 Description des stratégies 86
3.5 Description de l'outil de simulation 95
3.6 Temps de calculs 100

4.1 Application LOL 123
4.2 Capture LOL Interface régulation 124
4.3 Barre de commande de l'interface régulation 131
4.4 Saisie des données d'un patient 131
4.5 Affectation manuelle d'une requête de transport 132
4.6 Types d'exigence 132
4.7 Paramétrage de la planification automatique 133
4.8 Paramétrage de la plage horaire de calcul 133
4.9 Affectation automatique 134
4.10 Capture LOL aides aux simultanés 134

Liste des tableaux

1.1 Statistiques sur le marché de transport sanitaire (année 2018) ; Source : [LivreBlanc, 2018] 6
1.2 Tarifs conventionnels applicables aux transports sanitaires par ambulance (à compter du 1er février 2013) ; Source : [Ameli.fr, a] 6
1.3 Tarifs conventionnels applicables aux transports sanitaires par VSL (à compter du 1er janvier 2015) ; Source : [Ameli.fr, b] * La liste des départements qui composent les zones A, B, C et D, et la liste des communes concernées par l'application de la prise en charge sont disponibles dans [Ameli.fr, b] 7

2.1 Break settings 25
2.2 Vehicle attributes 27
2.3 Criteria definition 28
2.4 Fields of the objective function 28
2.5 17 original instances 40
2.6 Objective function setting 40
2.7 ALNS Parameters 41
2.8 Removal and insertion operator utilities over 17 A-instances 43
2.9 ALNS under $C1$ objective function 49
2.10 ALNS under $C2$ objective function 52
2.11 ALNS under $C3$ objective function 53
2.12 ALNSperformance on A-instances 54
2.13 HLA+ and HLA performances on A-instances 55
2.14 ALNS vs 1-Trip-ALNS 56
2.15 Break at the depot (ALNS-BAD) versus break on the way (ALNS-BOW) 57
2.16 ALNS with customers preferences 58
2.17 ALNSversus SOA [Lim et al., 2017] on RDARP 58
2.18 Optimal and [Lim et al., 2017] results on MTVRPTW small (10 runs) instances 59

2.19 ALNS results on MTVRPTW small (10 runs) instances 59
2.20 ALNS vs Azi et al. on MTVRPTW instances 60

3.1 paramètres des pauses 83
3.2 Paramètres de la stratégie $S1$ 88
3.3 Paramètres de la stratégie $S2$ 91
3.4 informations sur les instances utilisées 96
3.5 Valeurs des paramètres 98
3.6 Comparaisons des stratégies sur chaque problème 109
3.7 Influence du paramètre H_{libre} sur la qualité de la solution 110
3.8 Influence du paramètre H_{Fix} sur la qualité de la solution avec H_{libre} fixé . 110
3.9 Influence du paramètre F sur la qualité de la solution avec H_{libre} et H_{Fix} fixés 110
3.10 Résultats avec une seule réplication de la stratégie 1 sur 4 paramètres . . 111
3.11 Résultats avec une seule réplication de l'algorithme ALNS sur 4 paramètres 111
3.12 Résultats avec une seule réplication de la stratégie 1 sur 4 paramètres avec informations parfaites 112
3.13 Gap entre solution stratégie 1 et stratégie 1 avec information parfaite . . 112
3.14 Gap entre la solution de la stratégie 1 et VPL 113
3.15 Comparaison entre HV=720 et HV=1200 sur grandes instances 114
3.16 Comparaison entre HV=720 et HV=1200 sur moyennes instances 115
3.17 Résultats avec une seule réplication de la stratégie 1 pour 4 paramètres avec information parfaite sur les grandes instances 116
3.18 Résultats avec une seule réplication de la stratégie 1 sur 4 paramètres avec information parfaite sur les instances moyennes 117
3.19 Comparaison entre la minimisation et sans minimisation du nombre de chauffeurs utilisés dans VPL 118
3.20 Comparaison entre la minimisation et sans minimisation de chauffeurs utilisés dans VPL 119

A.1 Computation details of the counterexample 144

B.1 ALNS-Best 146
B.2 ALNS avg 147
B.3 HLA 148
B.4 HLA+ 149
B.5 ALNS 1-trip 150
B.6 ALNS BOW 151
B.7 ALNS with customers preferences 152

Acronymes

AD Assistant diploma.

ALNS Adaptative large neighborhood search.

BAD CrewDARP with breaks at the depot.

BOW CrewDARP with breaks on the way.

CA Chiffre d'affaire.

CEI Norme internationale concernant la sécurité de l'information.

CI Certification logiciel comptabilité informatisée.

CrewDARP Crew scheduling problem+ dial a ride problem.

CSS Code de la sécurité sociale.

DARP Dial a ride problem.

DEA Diplôme d'état d'ambulancier.

DMTP Dynamic medical transportation problem.

DOM-TOM Départements d'outre-mer et territoires d'outre-mer.

HDS Certification hébergement des données de santé.

HLA Human-Like Algorithm.

HLA+ Improved Human-Like Algorithm.

ISO International organization for standardization.

LHV Light health vehicles.

LOL Lomaco onLine.

LS Local search.

MP Master problem.

MTVRPTW Multitrip vehicle routing problems with time windows.

PDPTW Pickup and delivery problem with Time windows.

PLNE Programmation linéaire en nombres entiers.

RDARP Resource-constrained dial a ride problem.

SaaS Software as a service.

SAMU Service d'aide médicale urgente.

SD State diploma.

SMUR Service mobile d'urgence et de réanimation.

TAP Transport assis professionnalisé.

TL Taxi licence.

U.R Uncovered request.

VA Valeur ajoutée.

VPL Veille pour le lendemain.

VRP Vehicle routing poblem.

VRPTW Vehicle routing problems with time windows.

VSL Véhicule sanitaire léger.

Chapitre 1

Introduction Générale

Sommaire

1.1 Introduction 2
1.2 Transport sanitaire 3
1.2.1 Définition du transport sanitaire 3
1.2.2 Les organismes impliqués dans le transport sanitaire 3
1.2.3 La croissance du secteur de transport sanitaire et ses enjeux . . 4
1.2.4 La réglementation 5
1.2.5 Difficulté du métier 9
1.3 La société Lomaco 9
1.3.1 Présentation de la société Lomaco 9
1.3.2 Lomaco en chiffres 10
1.3.3 Les objectifs de Lomaco 10
1.4 Processus de planification 11
1.4.1 Régulation des demandes de transports des patients 11
1.5 Problèmes de décision 14
1.5.1 La gestion de temps des employés 14
1.5.2 La planification de la veille pour le lendemain (partie proactive) 14
1.5.3 La régulation dynamique de la flotte de véhicules (partie réactive) 15
1.6 Organisation de la Thèse 15

1.1 Introduction

Au cours des dernières années, le secteur du transport ambulancier a connu une profonde mutation au niveau du tissu économique. Les petites sociétés familiales, souvent de proximité, laissent de plus en plus place à des grands groupes qui ont l'avantage d'avoir une flotte importante et reconfigurable en fonction des besoins.

La société Lomaco commercialise des progiciels essentiellement dédiés à la gestion des sociétés ambulancières. Parmi l'ensemble des logiciels proposés, nous nous intéresserons plus particulièrement au module dédié à la régulation des tournées de véhicules. Ce module permet d'affecter les demandes de transport sanitaire (ambulance, VSL (Véhicule Sanitaire Léger) , taxi...) aux équipages de la société (véhicules et ambulanciers). La société prend connaissance d'une partie des requêtes la veille. Le reste de l'activité est gérée au fil de la journée.

Cette thèse se focalise sur le développement des méthodes d'optimisation pour la planification proactive des plans de routes d'un grand réseau d'ambulances et la planification réactive pour leur mise à jour durant la journée. Les solutions proactives correspondent au plan de routes du matin qui planifie les heures de début de travail des chauffeurs et leurs premières visites. Ce plan doit être robuste pour permettre une meilleure gestion des requêtes apparues au cours de journée. Lorsque, de nouvelles requêtes sont soumises au système, les solutions réactives doivent proposer leur intégration dans le plan de route courant, tout en garantissant son efficacité et le potentiel d'acceptabilité de futures requêtes.

Les motivations principales d'un point de vue applicatif sont :

1. Permettre la gestion des problèmes de taille grandissante : Une flotte allant jusqu'à 800 véhicules et qui réalise jusqu'à 1500 voyages par jour

2. Intégration d'un nombre important de contraintes métier réelles notamment la gestion de plusieurs types de véhicules et de multi-dépôts, la constitution des équipages d'ambulanciers, la gestion du temps de travail, l'affectation des pauses, ...

3. Réalisation d'une application commerciale qui sera utilisée par les entreprises de transport sanitaire

Dans ce chapitre, nous allons mettre l'accent dans la première section 1.2 sur le transport sanitaire, ses acteurs, sa croissance et ses enjeux. Nous nous intéresserons ensuite à la réglementation légale et aux différentes difficultés liées au métier. Dans la section suivante 1.3, nous présenterons la société Lomaco et nous focaliserons sur ses objectifs et son implication dans le sujet de thèse. Dans une troisième section 1.4, nous aborderons le processus de planification des régulateurs du transport sanitaire. Enfin, nous ferons un état des problèmes de décisions dans la cinquième section 1.5.

1.2 Transport sanitaire

1.2.1 Définition du transport sanitaire

Le transport sanitaire consiste à transporter depuis/vers des pôles de santés des patients, pour des raisons de soins ou de diagnostic, qui sont en difficulté de se déplacer de manière autonome.

L'ambulancier est un professionnel de santé qui exerce son activité au sein d'une entreprise de transport sanitaire privée ou d'un établissement de santé. Il travaille uniquement sur prescription médicale et intervient à la demande des hôpitaux pour réaliser des transferts de patients entre établissements ou pour réaliser des interventions SAMU. Dans ce dernier cas, il participe ponctuellement aux secours publics. Toutefois, il n'intervient en aucun cas sur la voie publique.

Les ambulanciers effectuent près de 40 millions de voyages pour transporter 5 millions de malades chaque année. Ils facilitent et assurent une parfaite égalité dans l'accès aux soins.

Les ambulanciers ont trois différents statuts :

— Ambulanciers du secteur privé : Ils travaillent dans des entreprises privées et ils dépendent du ministère des transports ;

— Ambulanciers du secteur public : Ils sont fonctionnaires et dépendent du ministère de la santé ;

— Ambulanciers du secteur associatif : ils travaillent pour des organismes spécifiques (Croix rouge française, protection civile...).

Concernant le transport sanitaire terrestre, il peut être assuré par :

— Une ambulance, si l'état de santé du patient nécessite un allongement ou une surveillance médicale ;

— Un véhicule sanitaire léger (VSL), si le patient est transporté assis et a besoin d'aide pour marcher ou pour accomplir des tâches administratives ;

— Un Taxi conventionné, si le patient n'a besoin d'aucune assistance particulière et peut se déplacer seul.

Comme mentionné dans [LivreBlanc, 2018], en 2018, La répartition des transports sanitaires terrestres est de 41% pour les taxis, 40% pour les ambulances et 19% pour les VSL (Cf. figure 1.1).

1.2.2 Les organismes impliqués dans le transport sanitaire

Nous distinguons quatre catégories d'acteurs du transport sanitaire : l'établissement de santé, le transport sanitaire, l'assurance maladie et le patient (voir Figure 1.2)

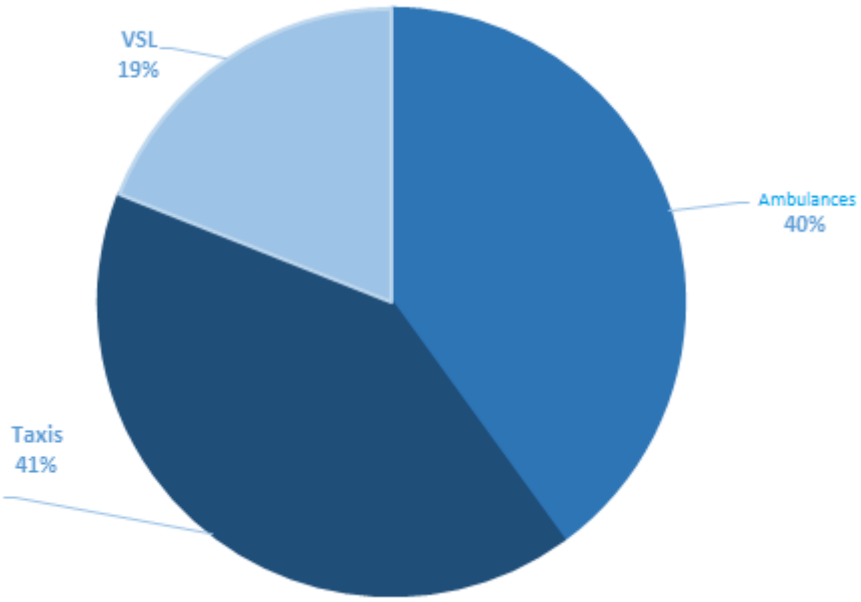

FIGURE 1.1 – La répartition du nombre de transport par type de véhicule (2018).

1. Le payeur, Assurance maladie : L'assurance maladie vise à contrôler les dépenses et faire des économies en favorisant le partage d'un trajet par plusieurs patients dans le même véhicule (transport partagé) et en utilisant les justes moyens.
2. Le prescripteur, Établissement de santé : Les établissements de santé ont pour objectif de gagner du temps en optimisant les flux des entrées et des sorties et d'avoir un meilleur taux de disponibilité des lits.
3. Le prestataire, Le transporteur sanitaire : Le transporteur sanitaire vise à optimiser l'utilisation des ressources et des moyens disponibles (ambulanciers, véhicules, ...) ainsi qu'améliorer la productivité afin d'avoir une activité fluide, une équité du traitement, une télé-transmission des documents et une bonne communication avec l'établissement de santé.
4. Le bénéficiaire, Le patient : Le patient cherche à avoir un libre choix, une plus grande capacité à obtenir un transport à l'heure souhaitée et une meilleure prise en charge.

1.2.3 La croissance du secteur de transport sanitaire et ses enjeux

Le transport sanitaire est un secteur qui emploie environ 50 000 salariés (dont 96% en CDI) chez 6000 entreprises. C'est un secteur en pleine croissance puisqu'on note une progression annuelle de +3,5%. Cette croissance d'activité est liée au vieillissement de la population et à l'augmentation des maladies chroniques (entre 25 et 75% de l'augmentation des dépenses de santé) (Voir Tableau 1.1).

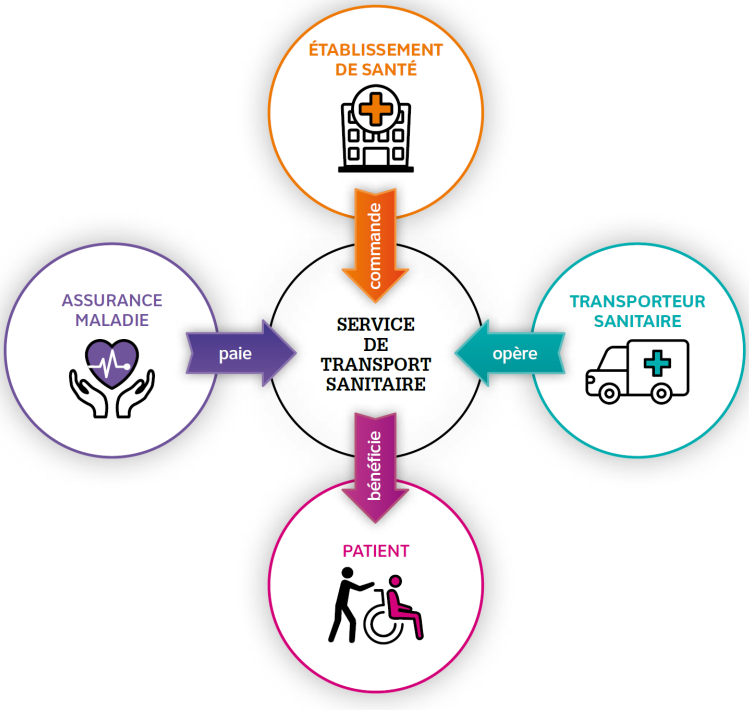

FIGURE 1.2 – Interaction entre les différents acteurs du transport sanitaire ; Source : [Santé-Mobilité-Services, 2017]

1.2.4 La réglementation

Pour suivre la croissance des dépenses de santé, plusieurs réglementations ont été déployées par l'assurance maladie notamment les tarifs conventionnels, l'article 80 et le transport partagé.

1.2.4.1 Les tarifs conventionnels

Tarifs conventionnels des ambulances La tarification du transport sanitaire par ambulance a été révisée le 1er janvier 2013. Entrée en vigueur le 1er février de la même année, la nouvelle grille tarifaire comporte trois volets (Voir Tableau 1.2) :

— Un forfait : Selon le lieu du siège de l'entreprise de transport sanitaire, nous distinguons un forfait département, un forfait agglomération et un forfait prise en charge ;

— Un tarif kilométrique basé sur la distance entre le lieu de prise en charge et de dépôt, une déduction est faite des 3 premiers kilomètres inclus dans le forfait départemental et dans le forfait agglomération, ou dès le 1er kilomètre parcouru dans le cas de la facturation de la prise en charge ;

— Une valorisation trajet court calculée en fonction de la distance parcourue et qui s'applique de manière à diminuer progressivement jusqu'au 19ème kilomètre parcouru.

Nombre d'entreprise	5316
Nombre de salariés	50 000
Nombre de patients	5 millions
Nombre de transports effectués	40 millions
Valeur ajoutée (% du CA)	78%
Frais de personnel (% du VA)	62-63 %
Excédent brut d'exploitation (% du CA)	13%
Endettement (% des fonds propres)	58%

TABLE 1.1 – Statistiques sur le marché de transport sanitaire (année 2018) ; Source : [LivreBlanc, 2018]

Élément de tarif	Montant
Forfait départemental	51,30 €
Forfait agglomération	57,37 €
Prise en charge	64,30 €
Tarif kilométrique	2,19 €
Valorisation trajet court 5 km parcourus	7,00 €
Valorisation trajet court $>$ 5 et $\leq$ 10 km parcourus	5,50 €
Valorisation trajet court $>$ 10 et $\leq$ 15 km parcourus	4,00 €
Valorisation trajet court $>$ 15 et $\leq$ 19 km parcourus	2,50 €

TABLE 1.2 – Tarifs conventionnels applicables aux transports sanitaires par ambulance (à compter du 1er février 2013) ; Source : [Ameli.fr, a]

tarifs conventionnels des VSL La tarification des transports sanitaires par véhicule sanitaire léger (VSL) comporte trois volets (Voir Tableau 1.3) :

— Un forfait : Forfait départemental qui varie selon le lieu du siège de l'entreprise de transports sanitaires ;

— Une valorisation trajet court : Est en fonction du nombre de kilomètres parcourus. La valorisation s'applique de manière dégressive, jusqu'au 18e km parcouru ;

— Un tarif kilométrique : Applicable à la distance parcourue en charge du lieu de départ au lieu d'arrivée, déduction faite des 3 premiers km inclus dans le forfait départemental, ou dès le 1er km parcouru dans le cas de la facturation de la prise en charge.

Élément de tarif	Montant
Forfait départemental zone A*	13,85 €
Forfait départemental zone B*	13,45 €
Forfait départemental zone C*	12,60 €
Forfait départemental zone D*	11,97€
Prise en charge	15,85€
Tarif kilométrique	0,89 €/ km
Valorisation trajet court ≤7 km parcourus	+ 6,26 €
Valorisation trajet court > 7 et ≤ 8 km parcourus	+ 6,05 €
Valorisation trajet court > 8 et ≤ 9 km parcourus	+ 5,53 €
Valorisation trajet court > 9 et ≤ 10 km parcourus	+ 5,00 €
Valorisation trajet court > 10 et ≤ 11 km parcourus	+ 4,48 €
Valorisation trajet court > 11 et ≤ 12 km parcourus	+ 3,96 €
Valorisation trajet court > 12 et ≤ 13 km parcourus	+ 3,44 €
Valorisation trajet court > 13 et ≤ 14 km parcourus	+ 2,92 €
Valorisation trajet court > 14 et ≤ 15 km parcourus	+ 2,40 €
Valorisation trajet court > 15 et ≤ 16 km parcourus	+ 1,88 €
Valorisation trajet court > 16 et ≤ 17 km parcourus	+ 1,36 €
Valorisation trajet court > 17 et ≤ 18 km parcourus	+ 0,83 €

TABLE 1.3 – Tarifs conventionnels applicables aux transports sanitaires par VSL (à compter du 1er janvier 2015) ; Source : [Ameli.fr, b]
* La liste des départements qui composent les zones A, B, C et D, et la liste des communes concernées par l'application de la prise en charge sont disponibles dans [Ameli.fr, b]

1.2.4.2 La réglementation : Article 80

L'article 80 est entré en vigueur le 1er octobre 2018 afin de préciser les règles de prise en charge des transports des patients et des dépenses des transports inter et intra hospitaliers.

Aujourd'hui, les transports de patients entre établissements sont financés d'une part par l'enveloppe des soins de ville (transferts provisoires pour la réalisation d'une séance de chimiothérapie, de dialyse ou de radiothérapie / transferts définitifs entre deux entités juridiques) et d'autre part par le budget des établissements de santé (transferts provisoires hors séances / transferts entre deux entités géographiques d'une même entité juridique).

Cette mesure prévoit de confier la responsabilité du financement de l'ensemble des transports, inter et intra hospitaliers, aux établissements de santé pour les inciter à respecter les prescriptions médicales et favoriser une meilleure adéquation entre le mode de transport et l'état de santé du patient. Plusieurs principes et orientations ont été déclinés dans

des conventions négociées au niveau départemental. Actuellement, le protocole national met en œuvre des dispositifs de régulation cohérents sur l'ensemble du territoire tout en prenant en compte les spécificités locales.

Les mesures à porté nationale sont les suivantes :

— Définition des tarifs de référence : Le protocole permet de ne plus prendre en compte systématiquement l'augmentation des tarifs préfectoraux et d'en limiter l'impact par un dispositif de désindexation totale ou partielle en fonction de cette hausse sur une durée de 5 ans ;

— Fixation des taux de remise : Le protocole prévoit que le taux de remise (article L.322-5 du CSS), applicable aux tarifs issus de la réglementation des taxis, est majoré chaque année et modulé en fonction des résultats annuels d'évolution des dépenses.

D'après [Assurance-Maladie, 2020], En cas de non prise en compte de l'article 80, la croissance des dépenses de taxis est de +3,0% à fin novembre 2019. L'application de l'article 80 a abouti à un net ralentissement de la croissance, soit +1,6% à fin novembre 2019.

Ce protocole a réalisé d'importantes économies. Hors effet désindexation, l'économie réalisée est estimée à 78.6 M€, alors qu'elle atteint 33.4 M€avec l'effet de désindexation (en 2019).

1.2.4.3 Transport partagé

L'article L. 322-5 du code de la sécurité sociale stipule que :

— Le transport partagé est prescrit par principe sauf opposition du prescripteur ;

— Dans le respect de la prescription, les frais de transport prennent en compte les conditions de transport et le nombre de patients transportés et ils sont pris en charge sur la base du trajet le moins onéreux.

— Un système d'incitation financière avec l'application d'un malus de 5% sur le CA des transporteurs sanitaires dont le taux de transport partagé serait inférieur à 5% et un reversement à ces professionnels, d'une part croissante des économies générées, à mesure que le taux de transport partagé croit.

Le transport partagé permet de réduire les dépenses de l'assurance maladie, de renforcer les marges et les taux d'occupation des transporteurs, de réduire le temps d'attente des véhicules pour les patients et de diminuer l'empreinte écologique des transports sanitaires. La montée en charge du transport partagé à 30% des transports réalisés permettrait de réaliser une économie de 77 millions d'euros. En moyenne, le transport partagé représente 15% du montant total remboursable des transports en VSL : En 2019, 3 800 entreprises de transport sanitaire ont réalisé du transport partagé avec 2 ou 3 patients.

1.2.5 Difficulté du métier

L'article 80 a engendré certaines difficultés de gestion aux autres acteurs :

— Établissements de santé : en plus de son activité principale qui est la gestion opérationnelle des soins quotidiens, les hôpitaux doivent aussi gérer les transports de santé.

— Les transporteurs : les petites entreprises du transport sanitaire rencontrent une difficulté économique liée à la procédure d'appels d'offres, à la monopolisation des grandes entreprises et à la concurrence des prix.

Les entreprises qui opèrent dans le transport des patients sont dans des situations très difficiles. Ces complications sont généralement dues aux charges sociales, et à l'accroissement des coûts qui est d'environ 2% (si nous comparons la même période de l'année 2018 et 2019) ([KPMG, 2019]). Ces coûts comprennent entre autres, les coûts immobiliers (sites logistiques), les frais d'entretien, les coûts d'assurance et le coût du carburant qui a augmenté de 10%.

De plus, nous identifions également des problèmes liés à la mauvaise gestion qui s'explique par un manque d'anticipation et une faiblesse du taux de remplissage des journées d'ambulanciers (en moyenne à 44% du temps de travail des ambulanciers).

Pour répondre aux changements de la réglementation et afin de faire face aux difficultés du métier, plusieurs éditeurs de logiciel sont entrés en jeu afin de proposer des solutions d'optimisation et de gestion d'activités pour les transporteurs.

Parmi ces sociétés, Lomaco a orienté son choix vers un sujet de thèse.

1.3 La société Lomaco

1.3.1 Présentation de la société Lomaco

Lomaco fait partie de la French Tech, un mouvement français des startups qui vise à promouvoir l'excellence française dans le domaine de l'innovation. Lors de sa création en 1966, Lomaco commence par la saisie des données comptables du cabinet Rossignol, ensuite elle se lance dans l'édition des logiciels de gestion. Après quelques années, Lomaco devient une entreprise de service numérique qui héberge des logiciels à distance. Aujourd'hui, l'entreprise est devenu un créateur d'énergie qui conçoit et commercialise des logiciels, des solutions de géolocalisation, du cloud computing et du matériel informatique. La société détient trois certifications :

— NF203 CI : Certification Logiciel Comptabilité Informatisée.

— ISO/CEI 27001 : Certification internationale relative à la sécurité des systèmes d'information de l'ISO et la CEI.

— HDS : Certification Hébergement des données de santé.

1.3.2 Lomaco en chiffres

Filiale du groupe FIDECIEL, Lomaco emploie aujourd'hui 112 collaborateurs dans la France métropolitaine et les DOM-TOM. En 2019, l'entreprise a réalisé quelques 23 Millions d'euros de chiffre d'affaire. Elle a un portefeuille de 6700 clients sous les contrats de maintenance et compte 9500 utilisateurs au niveau de LOMOACO OnLine, le logiciel full-web dédié à la gestion des transports sanitaires, utilisable à distance, partout, tout le temps et sur tous supports. (voir figure 1.3)

Lomaco agit actuellement sur différents métiers (voir figure 1.3) :

- Fournisseurs de biens médicaux : 1120 contrats.
- Funéraire : 1200 entreprises équipées.
- Transport sanitaire : 60% des professionnels équipés, 1400 entreprises sous contrat de maintenance et des Taxis conventionnés.
- Formation et e-learning.
- Matériels et impression 3D.
- Assistance et maintenance.

1.3.3 Les objectifs de Lomaco

Lomaco s'est engagée et s'engage toujours à :

- Avoir une stratégie claire est forte concernant l'information et sa protection.
- Avoir des processus structurés et méthodiques.
- Avoir une vision sur les points faibles et les menaces.
- Avoir une anticipation efficace.
- Avoir une haute protection des données personnelles de ses clients.
- Avoir des équipes mobilisées et sensibilisées sur la sécurité.
- Avoir des indicateurs facilitant le pilotage de la sécurité
- Avoir des services améliorés et travailler sur leur amélioration continue.

Lomaco s'est investi dans ce sujet de thèse, qui est la planification proactive et réactive des réseaux d'ambulance, afin d'élargir ses services pour aider ses clients à améliorer leurs flux, optimiser leurs ressources et réduire leurs dépenses.

Afin de mener cette thèse, nous nous intéresserons au métier du transporteur et nous décrypterons sa façon de réguler des demandes de transport afin d'en tirer les problèmes de décision. Dans les chapitres qui suivent, nous proposerons les solutions adéquates.

FIGURE 1.3 – La société Lomaco en chiffres

1.4 Processus de planification

Dans cette partie, nous nous focalisons sur le métier du régulateur qui a pour mission la régulation des demandes de transport des patients et la planification prévisionnelle de la disponibilité des ambulanciers.

Ces informations sont spécifiques aux sociétés de transport sanitaire que nous avons visitées.

Les demandes en transport sanitaire proviennent de deux types de clients par courrier, fax ou téléphone :

- Des centres hospitaliers ;
- Des patients qui disposent d'une prescription médicale demandant un transport sanitaire pour se déplacer.

1.4.1 Régulation des demandes de transports des patients

Dans la régulation, ils existent deux catégories de transport :

- Les transports qui peuvent être préalablement programmés : Transports réguliers et prévisibles qui sont déjà définis au début de la journée ;
- Les transports dont la demande émane au fur à mesure au cours de la journée nécessitant ainsi une réponse et une prise en charge dans les minutes ou les heures qui suivent l'appel.

Le régulateur est tenu alors de répondre au mieux à la demande du client en lui proposant le voyage adapté à ses besoins et ses exigences dans le meilleur délai.

Afin de bien répondre aux besoins des patients, le régulateur, assiste, ordonne et gère les ressources (ambulances et véhicules).
Lors de la planification le régulateur doit prendre en compte aussi certaines contraintes légales :

- Les pauses de 20 minutes minimum accordées aux ambulanciers, avec la contrainte : 6 heures de travail maximum en continu (sans aucune pause) et au moins une heure déjà travaillée. Ces pauses peuvent être prises généralement n'importe où.
- La pause déjeuner (resp. La pause dîner) qui dure une heure et concerne les ambulanciers ayant démarré leur service avant 11 heures (rep. 18 heures). Cette pause doit avoir lieu entre 11H15 et 14H (resp. 18H et 20H). La pause est prise généralement au dépôt, elle peut être raccourcie à 30 min ou être prise hors dépôt avec une indemnité financière octroyée à l'ambulancier en question.
- Le type de véhicule ambulance doit être affecté à un binôme d'ambulanciers formant une équipe dont un diplômé d'ambulancier d'état (DEA) et l'autre auxiliaire.
- Le type de véhicule VSL doit être affecté à un seul ambulancier avec la certification d'auxiliaire, tandis que les Taxis doivent être conduits par un ambulancier qui dispose d'un diplôme de TAXI sanitaire.
- Les transports partagés ne sont autorisés que pour le VSL et les Taxis.

Le régulateur dispose d'un outil sur la plateforme de Lomaco qui lui permet de visualiser en temps réel la position des véhicules et des chauffeurs, ce qui lui permet d'avoir une meilleure visibilité et l'aide dans le processus d'affectation des nouvelles missions. Le régulateur reçoit la demande du client dans laquelle il précise :

- L'emplacement de prise en charge et de dépose ;
- Le type de véhicule prescrit : Ambulance (Couché) ou VSL / Taxi (Assis) ;
- Les autres exigences du client : Chauffeur qui parle en anglais, ambulance bariatrique ...

A partir des informations fournies et en fonction de la disponibilité des ressources, le régulateur négocie une fenêtre de temps de passage pour récupérer le patient. S'il n'arrive pas à trouver un ambulancier libre, le régulateur refuse la mission.
Les personnes impliquées dans la planification sont :

- L'opérateur qui reçoit les appels des clients. Il fixe les modalités de prise en charge et les horaires. Lorsque la course concerne le jour courant, il valide avec le régulateur la faisabilité.
- Le régulateur qui affecte les courses aux ambulanciers. Il démarre la journée avec un nombre de requêtes connues et les postes d'ambulanciers ouverts. Il traite les appels d'autres services de régulation (association hospitalière, autres ambulanciers), les urgences.

— Le responsable opérationnel qui définit les postes de travail et y affecte le personnel.

Nous distinguons 4 étapes dans le processus de planification :

1.4.1.1 Plan trimestriel

Les entreprises de transport sanitaire travaillent pendant toute l'année sans arrêt. Les horaires de travail varient entre 5h du matin et 20h en fonction des demandes de transport. Durant les week-ends, un nombre minimal de personnes est chargé d'assurer les transports plantés et chaque salarié est tenu de travailler un samedi sur deux. La planification bi-mensuelle doit prendre en considération les absences pour maladie, pour formation, vacances ou les récupérations prévues durant le mois suivant pour prévoir un grand nombre d'ambulanciers disponibles. La préparation du planning se fait en concertation avec les salariés.

Ce plan garantit le volume d'heures contractuel de chaque employé (152 h/mois). Les heures supplémentaires sont calculées sur une base hebdomadaire. La variable d'ajustement se trouve alors dans la définition des amplitudes horaires dans les plans à plus court terme.

1.4.1.2 Plan hebdomadaire

L'amplitude horaire et l'heure de prise de service est mise à jour pour chaque équipage en prenant en considération le critère de minimisation des risques d'heures supplémentaires.

Ces décisions sont basées sur une estimation de la demande (en consultant l'historique et via l'expérience du planificateur) et les heures déjà travaillées par chaque employé. Notons que les heures de travail de nuit ne sont pas comptabilisées de la même manière.

1.4.1.3 Plan de la veille pour lendemain

Sur la base des données plus précises connues (30% à 70% des requêtes à servir sont déjà prédéfinis), les régulateurs ajustent les heures du début du service et les amplitudes horaire de travail des ambulanciers, le régulateur affecte les premières demandes connues de la journée à des équipages et véhicules, à cette étape, les ambulances ne sont pas avertis des affectations planifiées sauf dans le cas de longs trajets pour lesquels les équipages sont prévenus pour des raisons d'organisation personnelle.

1.4.1.4 Régulation en temps réel

Les distances séparant les centres de toutes les ruelles du secteur couvert sont connues. Le régulateur estime les autres distances sur la base de son expérience. Il affecte les courses de façon dynamique en prévenant un petit peu à l'avance les ambulanciers (10-30 min.). Il peut recomposer des équipages, notamment pour passer d'ambulance à VSL avec un seul chauffeur. Les pauses sont décidées par le régulateur. Il est chargé aussi

de décider du re-positionnement des véhicules inactifs (ceux qui sont restés à l'hôpital, retour à la base).

1.5 Problèmes de décision

Durant nos visites aux différentes entreprises d'ambulances, nous avons relevé trois grandes problématiques :

1.5.1 La gestion de temps des employés

Sur un horizon de plusieurs jours, un planning prévisionnel est fourni aux chauffeurs. Il s'agit de définir les jours et les horaires supposés d'embauche et de débauche. Pour la compagnie, l'objectif est de maximiser le rendement de ses employés tout en respectant lois, conventions et conditions de travail (équité, rythme). Chaque société a sa propre stratégie qui confère plus ou moins de la flexibilité à ses employés. Le processus de décision peut aussi varier avec un planning mensuel et des mises à jour hebdomadaires voire même quotidiennes. Le management dépend du statut des employés. Ainsi, les ambulanciers sont salariés avec un volume horaire forfaitaire mensuel de (152 h.). Les contraintes à prendre en considération concernent les périodes de repos avec les postes de nuit et de jour et la formation d'équipes conformément à la réglementation en vigueur. Les objectifs sont l'adéquation avec la demande prévisionnelle, la minimisation d'heures supplémentaires à payer, la formation d'équipes complémentaires et la satisfaction des employés (préférences matin-soir, week-end ou certains jours , équité). L'outil doit donc être robuste en fonction du processus de décision de l'entreprise et l'évolution des lois et conventions.

1.5.2 La planification de la veille pour le lendemain (partie proactive)

En J-1, un premier plan est élaboré à partir des demandes prédéfinis. Selon la nature des activités, un pourcentage plus ou moins faible des requêtes sont connues la veille (généralement entre 30 à 70%). Ce plan comprend une mise à jour des heures de début de service, d'amplitude horaires des ambulanciers et l'affectation des premières demandes connues aux équipes et véhicules. Ce plan doit être robuste dans le sens d'être capable d'assurer une meilleure gestion de l'acceptation des requêtes qui vont apparaître en cours de journée tout en minimisant les déviations des horaires de travail prévues dans le planning hebdomadaires prévisionnel.

1.5.3 La régulation dynamique de la flotte de véhicules (partie réactive)

La principale question est l'insertion de requêtes au plan courant de la journée. La régulation doit satisfaire toutes les contraintes de transport, en garantissant une bonne qualité de service et de travail, ainsi qu'une meilleure gestion et planification des demandes à venir. D'autres décisions sont aussi prises de façon dynamique : l'attribution des pause et la relocalisation de véhicules (retour base, ou centre de santé, fin ou prolongement (heures supplémentaires) de la période d'activité).

Pour tous les clients de Lomaco, les localisations des véhicules et les temps de transport sont connus en temps réel.

Les stratégies de planification dépendent des délais durant lesquels les directives sont transmises aux chauffeurs. Dans le cas où des pré-affectations sont transmises aux chauffeurs, la pénalisation de modifications est à intégrer dans les objectifs.

Les décisions prises par le régulateur ne sont pas toujours en fonction de la survenance d'un événement comme le besoin d'affecter un véhicule à une demande. La question de prise de pause ou relocalisation dépend d'une analyse de l'état actuel et prévisionnel du système.

Dans cette thèse, nous nous intéressons particulièrement aux problèmes cités ci-dessous :

— La planification de la veille pour le lendemain (partie proactive)

— La régulation dynamique de la flotte de véhicules (partie réactive)

Le problème de gestion du temps de travail n'est pas pris en considération dans cette thèse, il sera étudié en perspectives.

1.6 Organisation de la Thèse

Le Chapitre 2 présente une étude détaillée du problème de la planification de la veille pour le lendemain (partie proactive). Dans ce chapitre, nous proposerons une description détaillée, une méthode de résolution qui simule une planification manuelle et une Métaheuristique (Adaptive Large Neighberhood Search). La méthode est comparée avec la planification manuelle et autres méthodes connues dans la littérature pour la gestion des problèmes proches.

Le Chapitre 3 traite la problématique de planification en temps réel (partie réactive). Trois stratégies d'optimisation ont été proposées. Ces stratégies ont été comparées selon plusieurs problèmes DARP dynamique , DARP Dynamique avec temps de travail des ambulanciers, DARP Dynamique avec pauses (déjeuner, diner, repos) aux chauffeur, DARP avec plusieurs types de véhicules et finalement le problème complet de planification en temps réel des ambulances. La dernière partie de l'expérimentation numérique du chapitre, étudie l'impact de la planification de la veille pour le lendemain dans les

décisions dans le même. Plusieurs utilisations du planning de la veille pour le lendemain ont été proposées et comparées.

Le dernier chapitre 4, montre l'implémentation industrielle des méthodes développées durant cette thèse dans une application industrielle (Lomaco-Online(LOL)). Dans ce chapitre, nous présenterons les fonctionnalités de l'application LOL, comment les méthodes sont implémentées dans LOL et finalement, nous mettrons l'accent sur les demandes des clients qui ont nécessité des modifications sur les méthodes développées.

Chapitre 2

Planification proactive des réseaux d'ambulance

Résumé du chapitre

Ce chapitre traite un problème auquel est confronté le transport sanitaire : la planification de la veille pour le lendemain. Ce processus consiste à élaborer un premier planning de travail aux ambulanciers sur la base des demandes de transport préalablement connues. Le planning doit mettre à jour les horaires de travail des ambulanciers ainsi que l'affectation de leurs premières missions.

Le plan calculé la veille n'est pas exactement exécuté, ce plan est utilisé par les régulateurs pour appuyer leurs décisions. Cependant, nous définissons ce travail dans un contexte déterministe. En effet, pour de nombreuses entreprises de transport sanitaire, peu d'événements imprévus se produisent en début de journée. Ainsi, le plan du jour suivant est mis à jour progressivement au cours de la journée. La deuxième raison est d'ordre technique. Des demandes imprévues peuvent surgir dans différents lieux. Les pires scénarios ne sont pas faciles à définir, et par conséquent, des approches d'optimisation robustes sont également difficiles à concevoir. Certaines approches stochastiques basées sur l'échantillonnage de scénarios semblent plus faciles à développer dans ce contexte. Et un solveur déterministe tel que nous le proposons ici est souvent utile pour ce genre de situations.

Nous abordons ce problème de « planification du lendemain » comme un problème d'optimisation combinatoire dans la famille des tournées de véhicules (VRP). Nous avons nommé ce problème CrewDARP comme une extension de deux problèmes NP-difficiles connus dans la littérature (Dial-A-Ride-Problem et Crew Scheduling Problem), Dans cette étude, nous proposons une métaheuristique de grand voisinage « Adaptive Large Neighberhood Search » avec de nouveaux mécanismes et opérateurs adaptés spécialement à notre problème.

Nous proposons aussi deux autres heuristiques d'insertion, la première imite une planifi-

cation manuelle (human-like approach), et la deuxième représente une version améliorée de la première en utilisant une recherche Locale. L'efficacité de l'algorithme proposé a été testée sur des instances du monde réel. Les résultats ont montré l'efficacité de l'algorithme par rapport aux méthodes manuelles .

D'autres expérimentations ont été aussi réalisées et comparées dans ce travail : Pauses au dépôt ou n'importe où et la comparaison avec l'utilisation classique de ALNS.

De plus, les performances de l'algorithme développé ont été évaluées sur deux autres problèmes simplifiés connus dans la littérature : le Multi-Trip Vehicle Routing Problem with Manpower Planning et le Multi-Trip Vehicle Routing Problem. Les résultats montrent que l'algorithme développé réalise des solutions très compétitives par rapport aux méthodes récentes qui existent dans la littérature.

Ce chapitre à été soumis à l'éditeur pour publication sous forme d'article de journal : Skiredj, M., Garaix,T., Xie X. à Transportation Science

Sommaire

2.1 Introduction 19
2.2 Literature review 21
2.2.1 Scientific position of the crewdarp 22
2.2.2 A detailed review of a close problem – the rdarp 23
2.3 Problem description and modeling 24
2.3.1 Underlying graph 24
2.3.2 Drivers and depots 25
2.3.3 Vehicles 26
2.3.4 Requests 27
2.3.5 Objective function 28
2.4 Neighborhood evaluation 29
2.4.1 Request removal and insertion in single trips 29
2.4.2 Request insertion and removal in synchronized multitrips . . . 33
2.5 Adaptive large neighborhood search 35
2.5.1 Destruction operators 35
2.5.2 Insertion operators 37
2.6 Constructive heuristics 38
2.6.1 hla – Human-Like Algorithm 38
2.6.2 hla+ – Improved constructive heuristic 39
2.7 Real-world data set 40
2.8 Numerical experiments 41
2.8.1 ALNS implementation 41
2.8.2 CrewDARP 43
2.8.3 Benchmark on related problems 45
2.9 Conclusion 47

Absctract

This chapter addresses a real-life medical transportation problem where routing plans of the ambulance crews have to be built for a single day. The main originality of this problem is the possibility of decomposing and recomposing crews throughout the day, since customers may require vehicles with one or two paramedics and the working hours of paramedics may differ. Therefore, trips and breaks of different paramedics have to be synchronized. The problem is modeled as a dial-a-ride problem with crew composition, named CrewDARP for short, that shares some common terminology with the multitrip pick-up and delivery problem with time windows, breaks and crew composition, which can be seen as an extension of several NP-hard vehicle routing problems with time windows (VRPTW).

An Adaptive Large Neighborhood Search (ALNS) is designed to solve this difficult problem on large instances (thousands of customers and hundreds of vehicles and drivers). Several algorithmic locks are addressed to control the combinatorial complexity and the computing time of the procedures embedded in the neighborhood exploration.

ALNS efficiency is assessed by means of extensive experiments on real-world instances. Our algorithm is also shown to be very competitive against the state-of-the-art solution for the MTVRPTW and the resource-constrained DARP. In addition, the relevance of the setting with crew modifications is also verified numerically.

2.1 Introduction

Medical transportation services are facing increasing demand these recent years, as stated in [Andrew et al., 2020]. The aging population and home care service expansion contribute the most to this increase.

The dispatchers who assign transportation requests to ambulances are also subject to an increase in the number of demands and the size of the fleet of vehicles to manage. Actually, leading companies buy smaller ones, and independent companies join their dispatching services together ([Maligorne and Marcadé, 2018]). Some companies deal with thousands of requests each day in the same region. A centralized dispatching service with a complete view of the demand allows operating cost savings and better coverage of the demands.

The need for an intelligent decision support system becomes critical when dealing with a large number of requests with complex specifications on the transportation mode (car-sharing, lying down, seating), vehicle equipment (bariatric accommodations, wheelchairs) and driver skills (medical qualifications, strength, spoken languages). Manpower management is also involved in the decision-making process, as the dispatcher also manages drivers' working time management and crew composition/recomposition.

Medical transportation companies are interested in supporting dispatchers' activity,

standardizing dispatching abilities among junior and senior dispatchers, minimizing operational costs, minimizing delays and minimizing the rejection rate of demands.

This work is supported by Lomaco company, which is the French leading software publisher for ambulance dispatching. They have approximately 3,000 customers with different profiles, with 50 to 3,000 requests to serve every day. Their covered territory is rural, urban or a mix of both. The company strategy is to capture their market share by maximizing three different objectives : demand coverage, operational costs and the quality of service.

Three decision problems are identified : (i) the planning drivers' working days for a several-week horizon, (ii) the transportation planning of the next day and (iii) the real-time dispatching. This research aims to propose an algorithmic solution that computes the transportation plan for the next day on the basis of the already known demands.

New requests, cancellations and delays rise during the day. Therefore, the plan computed the day before is not exactly executed. This plan is used by the dispatcher to support their decisions. However, we define this research in a deterministic context. The main reason is that few unforeseen events occur during the beginning of the day for many medical transportation companies. Thus, the next-day plan is incrementally updated during the day. The second reason is technical. Unforeseen requests may appear at a very large number of locations. Worst-case scenarios are not easy to define ; therefore, robust optimization approaches are also difficult to design. Some stochastic approaches based on scenario sampling seem easier to develop in this context, and a deterministic solver, as we propose herein, is often useful in such a framework.

We address industrial issues of the 'next-day planning' as a combinatorial optimization problem in the family of VRPs for vehicle routing problems. We named this problem CrewDARP, an extension of the already defined dial-a-ride problem. As discussed in the review of the literature (Section 2.2), CrewDARP challenges state-of-the-art algorithms.

The main challenge raised in this chapter is related to crew management. Indeed, these last have to be composed and recomposed at any time.

In France, ambulances are dedicated to transportation with the passenger lying down and require two drivers. LHVs (Light Health Vehicles) and taxis require a single driver with different qualifications. For instance, at the beginning of the afternoon, several LHVs and taxi drivers are grouped to form ambulance crews since many patients are admitted to the hospital with a lying-down transportation prescription.

Working time and break management cause the second planning difficulty. The drivers know their provisional working times. However, these working times are often modified the day before. The next-day plan has to avoid extensive working time changes. Breaks are ruled by laws, and companies have to satisfy different working condition agreements for breaks. Breaks usually make vehicle routing problems harder to solve. Combined with the crew recomposition possibility, the problem becomes even harder, and classical solution approaches such as local search or dynamic programming have to be deeply

revisited.

The main contributions of this chapter are the definition of real-world VRP with non-standard additional constraints and an ALNS(Adaptative Large Neighborhood Search) algorithm that meets operational constraints, *i.e.,* able to solve instances with thousands of demands and hundreds of drivers in less than fifteen minutes.

In the next section, CrewDARP is discussed with respect to the classification of VRPs with a focus on more closely related problems that have been recently published. In Section 2.3, the CrewDARP is described formally and notations are defined.

In Section 2.4.1, we present how to adapt state-of-the-art preprocessing procedures used to quickly compute neighborhoods in local search frameworks. In this section, we also show that the quadratic complexity approach to deal with lunch breaks presented in [Vidal et al., 2014] is erroneous. We propose a cube complexity approach to address this problem.

We present two heuristics to construct a solution in Section 2.6. The first is a human-like approach, and the second is based on a regret insertion scheme. Our ALNSalgorithm is presented in Section 2.5.

An extensive series of experiments are presented in Section 2.8. The first series of experiments assesses the efficiency of our ALNSsetting. The second series compares our ALNSagainst state-of-the-art algorithms on close VRP variants, the MTVRPTW (multitrip VRP with time windows) and the PDPTW (pickup and delivery problem with time windows). Finally, our ALNSis compared with our constructive heuristics on a real-world data set. We also show that complicating constraints, such as crew changes, allow us to reach significantly better solutions.

In our conclusion section, we mainly discuss future lines of research opened with the CrewDARP definition. We also give some close problems that could benefit from some developments on the CrewDARP.

2.2 Literature review

CrewDARP can be considered a RichVRP, as defined in [Crainic et al., 2009], similar to a VRP with several additional constraints. CrewDARP combines DARP and MT-VRPTW characteristics. We explain in Section 2.2.1 why the algorithms designed for these two problems cannot be directly applied to CrewDARP. In Section 2.2.2, we comment in detail on the works of one team [Lim et al., 2017, Liu et al., 2015, Luo et al., 2019] who proposed heuristic and exact solution approaches for a problem similar to the CrewDARP derived from a medical transportation system in Hong Kong. We show the main differences between the two problems and discuss their solution approaches.

2.2.1 Scientific position of the crewdarp

The DARP is an extension of the PDPTW where transportation requests define a pick-up and a drop-off node to visit in a time window. In the DARP, a maximal ride time is imposed for each request. In CrewDARP, the maximal ride time is modeled through time window constraints as described in Section 2.3. According to the application, the authors add some additional constraints to the DARP. Most of these constraints, such as driver skills or vehicle equipment, necessitate slight adjustments to most of the solution algorithms.

In addition to the VRP with time windows (VRPTW), the MTVRPTW imposes a maximal route length. Then, a vehicle has to return to the depot when this limit is reached. These travel events from depot to depot are called trips. The maximal length of a trip can be measured as a load, a distance, a duration, an energy consumption, etc. In the most studied cases, a maximal duration is imposed from the depot to the last customer visited by the trip. A CrewDARP solution may also be modeled with multiple trips, but three additional decisions make the CrewDARP harder, namely, drivers to trip assignments, break schedules and start/end working times.

When time window constraints are involved in VRPs, exact approaches are usually based on an extended linear formulation since the polyhedral structure of the compact linear formulation is difficult to exploit. The route of each vehicle/driver is defined through a variable in MP (Master Problem), which is a set partitioning integer linear program. The exponential number of variables is managed by a column generation procedure. The most efficient algorithm was proposed by [Baldacci et al., 2008] ; several bound improvements are included and allow us to solve instances of VRPTW with up to 100 requests to the optimum.

When multiple trips are considered, the decomposition scheme is adapted, and the MP variables become trips. Therefore, trip synchronization is controlled in MP and deteriorates the quality of its linear relaxation bound. Unfortunately, the quality of this bound is closely related to the efficiency of these branch-and-price algorithms. In [Hernandez et al., 2016, Cattaruzza et al., 2016], such a method was implemented, and instances with 25 to 40 customers were solved to the optimum. When additional constraints on trip synchronization among crews are added, one can expect to decrease the performance of the column generation-based algorithm for CrewDARP.

Almost all metaheuristics have been applied to VRP and its variants. The hybrid algorithm of [Vidal et al., 2014] based on giant tours (a fictitious route where all nodes are visited) combined with an efficient splitting procedure reaches very competitive results on many variants of these problems. The splitting procedure is run on a graph model of the giant tour where arcs correspond to individual routes that satisfy the total ordering of nodes given by the giant tour. A path in this graph gives a solution. Efficient dynamic programming algorithms are used to obtain the shortest paths and optimized

sets of routes. In the CrewDARP, routes (from depot to depot) have to be assigned to drivers, and the assignment has to satisfy lunch breaks. In this context, the dominance rules used in the dynamic programming algorithm that splits the giant tour are not as efficient. Such approaches have to be deeply revisited to address the CrewDARP.

LS (local search) algorithms such as tabu search are often applied to VRP standalones in [Laporte and Semet, 2002] or as an intensification procedure in some metaheuristics in [Vidal et al., 2014, Sörensen, 2013]. LS is based on operators that locally modify a solution (node exchange, subsequence twist). The algorithm performs well when the neighborhood is quickly evaluated and allows us to visit many solutions. For CrewDARP, the classical operators based on movements involving few nodes and/or arcs, such as 2-Opt, cannot be easily evaluated through a very fast procedure. A small modification on a trip may imply chains of changes to reach a feasible solution that satisfies break consistencies and trip synchronizations.

ALNS was proposed in [Shaw, 1997] and has several successful applications in vehicle routing problems ([Pisinger and Ropke, 2005]). The core procedures of ALNS are destructor and constructor operators. In contrast to LS, a large part of the solution is modified at each iteration – one destruction followed by a construction. The constructor operator is comparable to a chain of changes in LS, but it is related to several simultaneous modifications of the solution. In the DARP case and when side constraints make the splitting procedure less efficient, ALNSis shown as a good alternative ([Pisinger and Ropke, 2005, Cattaruzza et al., 2016]). Crew recomposition and break and tight time window constraints seem to be a relevant context for ALNSapplications.

Synchronization constraints in VRPs have been introduced by a series of works of [Drexl, 2012]. The synchronizations already studied involve some routes to synchronize at determined nodes or arcs. A well-studied case is found in home health care services when two caregivers have to perform a task together ([Kergosien et al., 2013]). The algorithms dedicated to these problems involve repair mechanisms to satisfy synchronizations. In CrewDARP, the synchronization of trips is even more complex to check since each trip involves several drivers with their own working time.

Some applications in medical transportation involve reconfigurable vehicles such as vans, which can accept different types of wheel chairs (see [Téllez et al., 2018]). The decision on vehicle reconfiguration depends on the sequence of visited nodes, such as vehicle changes in the CrewDARP. However, crew recomposition is not necessary, and the problem seems easier to address.

2.2.2 A detailed review of a close problem – the rdarp

A series of works ([Zhang et al., 2015,Lim et al., 2017,Luo et al., 2019]) have been published by a team on a similar problem under different names ; the latter is the Resource-constrained DARP (RDARP). The underlying ambulance planning problem is located

in Hong Kong. As in the classical DARP definition, customers have to be transported from a pick-up node to a drop-off node within time window constraints.

In the RDARP, two types of operator drivers and assistants (drivers are also assistants) are considered with a fixed working time window for each operator.

The number of operators in the vehicle for a trip depends on the maximal number of assistants required over all the transportation requests of the trip. At least one driver has to be assigned to each vehicle. After at most two hours, the vehicle has to return to the depot for a thirty-minute break. The vehicle has to be cleaned for at least thirty minutes. After the break, operators may retire or be assigned to new crews.

The authors proposed a large neighborhood search heuristic in [Lim et al., 2017] to solve 365 realistic daily instances from a medical transportation company in Hong Kong with up to 200 requests per day. Branch-and-Cut-and-Price algorithm, published in [Luo et al., 2018], solves optimally twenty seven artificial instances with up to 36 requests within four hours. For this exact approach, the setting of instances is simplified with a fixed assignment between drivers and vehicles along the day.

The CrewDARP definition given in the following section is a generalization of the RDARP, where the working times of drivers are decisions, break scheduling rules are more flexible and request requirements on vehicle types and skills of the staff are more complex.

The ALNS algorithm designed to solve large instances of the CrewDARP is described in Section 2.5.

2.3 Problem description and modeling

In the CrewDARP, six types of decisions have to be made : (i) the requests to serve selection ; (ii) the ambulance driver working times ; (iii) the crew compositions/recompositions ; (iv, v) the assignment of the selected requests to crews and to vehicles ; and (vi) the schedule of tasks including breaks.

2.3.1 Underlying graph

Three node sets $P = \{1, ..., |P|\}$, $D = \{|P|+1, ..., 2|P|\}$ and $O = \{2|P|+1, ..., 2|P|+|O|\}$ represent the pick-up, drop-off and depot nodes, respectively. CrewDARP is defined on a directed graph $G = (V, A)$, where $V = (1, ..., 2|P| + |O|) = P \cup D \cup O$ is the node set and $A = \{(i, j)|i, j \in V\}$ is the arc set. A request consists of picking up a patient at node $i \in P$ and dropping off them at node $i + |P| \in D$. Therefore, the sets of requests and patients are identified by P.

Deterministic distances c_{ij} and travel times t_{ij} are defined for each arc $(i, j) \in A$. Deterministic values are assumed in many published VRP models and comply with medical transports where tight time constraints are imposed on the visiting time at each

node. However, the travel times t_{ij} and t_{uv} may have different values with u (resp. v) located at the same place as i (resp. j). Almost all commercial mapping tools provide reliable travel times on a specific date (day of the week and hour).

In contrast to many models in the literature, costs and travel times may violate triangular inequality since these weights are not proportional to real-world road networks. Moreover, in practice, driver habits do not follow the optimization of one criterion, and triangular inequality may be violated for any weight. The triangular inequality assumption allows us to decrease the complexity of several algorithmic procedures. We highlight in their description the algorithmic features that require or exploit this property.

2.3.2 Drivers and depots

A driver e is assigned to a single depot $o_e \in O$. Therefore, crew composition/recomposition can be performed among drivers and vehicles assigned to this depot. The set E of all drivers is a partition of the sets $\{E_o : o \in O\}$ of the drivers assigned to each depot.

A working time window $[ew_e, lw_e]$ is defined for each driver e. Starting earlier than ew_e and ending later than lw_e is penalized by linear costs in the objective function (2.1).

After T_{max} minutes out of the depot, each driver must return to their assigned depot. Note that this setting is often discarded ($T_{max} = \infty$) since it is dominated by break satisfaction constraints.

Each break is defined by seven attributes, namely, driver e, duration δ, location loc, starting time window $[eb, lb]$, trigger time window $]et, lt[$, working time limit Δ and binary indicator pb which is equal to 1 if patients on board are allowed during the break.

The meaning of these attributes is explained through the two types of breaks relevant for ambulance drivers in France. The values of the attributes are listed in Table 2.1.

TABLE 2.1 – Break settings

Attribute	Lunch Break	Work Break	Trip Break
δ	1 hour	20 minutes	30 minutes
$[eb, lb]$	$[11:00, 14:00]$	$]-\infty, +\infty[$	$]-\infty, +\infty[$
$]et, lt[$	$]11:00, 11:15[$	$]-\infty, +\infty[$	$]-\infty, +\infty[$
Δ	$+\infty$	6 hours	6 hours
pb	True/False	False	False
loc	Depot/On the way	On the way	Depot

The lunch break is triggered if the driver starts to work before 11 :00 and ends after 11 :15. The 'Lunch Break' must start between 11 :00 and 14 :00 and lasts 1 hour. The lunch break is usually taken at the depot, and no patient is allowed on board. These two constraints are relaxed for long trips. In our setting, the location and the on-board patient parameters are fixed independently for each break.

A driver cannot work continuously for more than 6 hours without a break of at least 20 minutes. In practice, the 'work break' is never scheduled because of idle times during the day and lunch (or diner) break.
In [Lim et al., 2017], the drivers must return to the depot after at most 6 hours for 30 minutes of 'Trip Break'.
A break that is scheduled at the beginning or the end of the working day of a driver is not included in their working time.
Several binary attributes, named qualifications, are assigned to each driver. The set $Q_e \subset Q$ of qualifications allows us to define constraints on vehicle driving allowances and skill requirements to serve some requests.
Drivers who operate in health vehicles may have different certifications. With a state diploma (SD), the driver is qualified as a paramedic and can provide some care and communicate with practitioners in health care facilities. An assistant diploma (AD) allows one to drive the LHV and to assist a paramedic in an ambulance. A taxi license (TL) is required to drive a taxi.
Strength and specific spoken languages are the most common preferences requested by customers.

— State Diploma (SD) : An ambulance driver with a state diploma must provide to patients optimal transport conditions (patient lying down), taking into account their state of health. Having drawn up an assessment of the patient's needs, it is up to him or her to situate the patient as best as possible, to monitor the patient during transport and, if necessary, to implement the appropriate rescue and emergency measures. On arrival at the hospital, clinic or other health center, the ambulance driver must know how to transmit to the nursing staff the information that he or she has regarding the patient's condition. He or she must ensure the maintenance of administrative documents (hospital, social security, etc.), the establishment of files, and the collection of the receipt.

— Auxiliary Diploma (AD) : An ambulance driver with an auxiliary diploma assists the ambulance driver with duties related to the SD and drives the vehicle when transporting a recumbent patient in an ambulance. In the case of transporting a seated patient, the ambulance driver with an AD is solely responsible for transporting seated patients (three at most) in LHVs.

— Aaxi License (TL) : An ambulance driver with a taxi diploma is approved by social security, which differentiates them from traditional taxis drivers. Thus, such drivers devote part of their activities to medical seated transport.

2.3.3 Vehicles

Each vehicle $k \in K$ is attached to a single reference depot $o_k \in O$. Each trip must start and end at the reference depot. Vehicles differ by their crew requirement M_k, the

transportation mode (sitting or lying down) sol_k, the maximal number of passengers (without drivers) κ_k, the car-sharing possibility cs_k and cost per kilometer α_k. The cost per kilometer α_k depends on fuel consumption, maintenance and highway tolls.

All possible crew compositions allowed for vehicle k are in set M_k. The usual requirements are related to diplomas, as described in Table 2.2.

TABLE 2.2 – Vehicle attributes

Attribute	Ambulance	LHV	Taxi
M_k	$\{SD \wedge AD\} \vee \{SD \wedge SD\}$	$\{AD\} \vee \{SD\}$	$\{TL\}$
sol_k	lie	sit	sit
κ_k	1	3	5
cs_k	$False$	$True$	$True$
α_k	2.19	0.89	0.89

A cleaning time CT can be scheduled at the depot after each arrival to the depot. In theory, the cleaning time necessity depends on the sequence of persons transported. In practice, we found only two behaviors : cleaning at the end of the day and a fixed cleaning time after each trip.

2.3.4 Requests

Each request $i \in P$ defines a service time at the pick-up location s_i (resp. $s_{i+|P|}$ at the drop-off location), a starting service time window $[es_i, ls_i]$ (resp. $[es_{i+|P|}, ls_{i+|P|}]$), the number of places required q_i (some companions can be transported and some wheelchairs can be used), the minimal staff member required sf_i, car-sharing acceptance $\rho_i \in \{0, 1\}$, a financial profit pr_i and a set of requirements about the crew and the vehicle. A set of suitable vehicles B_i is defined according to hard constraints in the prescription and the characteristics of each vehicle (bariatric accommodations, lying-down transport, capacity). The patient may require a list of binary characteristics on crews described by Q_e sets. Two behaviors are allowed : soft and hard constraints. The corresponding lists are denoted Φ_i and Ψ_i. Strength and spoken languages are the most common soft constraints. In practice, hard constraints are not common but may be activated by the planner to avoid some assignments.

The starting time windows are often computed in practice from the latest drop-off time $ls_{i+|P|}$, a maximal delay and $t_{i,i+|P|}$ for an outbound request i (outbound means the way to an appointment). Inbound requests (back to the home or hospital) are symmetrically built from the release time es_i instead of the latest drop-off time.

The maximal delay allows us to absorb lateness on travel and service duration, enables car sharing and forbids drivers' lunch breaks with patients on board. This delay differs from the maximal ride time used in several DARP formulations since it is not limited to

driving time.

2.3.5 Objective function

The objective function written in Equation 2.1 is split into six fields, which are described in Table 2.3 with the coefficients of Table 2.4. All the criteria are evaluated in economical loss or profit. Variable X_i means that request i is not covered ; $H_{i,\phi}$ indicates the preference of i is satisfied ; Y_{ijk} means that arc (i,j) is used by vehicle k ; A_e is the starting time of driver e ; and B_e is the finishing time. Function $(.)^+$ equals $\max(0,.)$.

$$\begin{aligned} \min Z = & \sum_{i\in P}(\theta_{1,i}+\theta_{3,i})X_i + \sum_{i\in P}\sum_{\phi\in\Phi_i}\theta_{2,i,\phi}H_{i,\phi} + \sum_{(i,j)\in A}\sum_{k\in K}\theta_{4,i,j,k}Y_{ijk} \\ & + \sum_{e\in E}(\theta_{5,e}(ew_e - A_e)^+ + \theta_{6,e}(B_e - lw_e)^+ + \theta_{7,e}(B_e - A_e - lw_e + ew_e)^+) \end{aligned} \tag{2.1}$$

TABLE 2.3 – Criteria definition

Criteria	Description	Class
$\sum_{i\in P} X_i$	Uncovered Request (U.R)	quality of service
$\sum_{\phi\in\Phi_i} \theta_{2,i,\phi}H_{i,\phi}$	cost of unsatisfied preferences (preferences)	quality of service
$\sum_{(i,j)\in A}\sum_{k\in K} c_{ij}Y_{ijk}$	Total distance costs traveled by vehicles (Distance)	economical
$\sum_{i\in P} pr_i X_i$	Sum of the financial profit of uncovered requests (Loss)	economical
$\sum_{e\in E}(ew_e - A_e)^+$	the sum of earlier working time (Earlier)	economical
$\sum_{e\in E}(B_e - lw_e)^+$	the sum of the later working time (Later)	working conditions
$\sum_{e\in E}(B_e - A_e - lw_e + ew_e)^+$	the sum of each of the extra working time (Longer)	working conditions

TABLE 2.4 – Fields of the objective function

Coefficient	Description
$\theta_{1,i}$	the cost related to uncovered request i
$\theta_{2,i,\phi}$	the cost if the preference ϕ is not satisfied for request i
$\theta_{3,i}$	$= pr_i$ the financial profit of uncovered request i
$\theta_{4,i,j,k}$	$= \alpha_k c_{ij}$, the routing cost of (i,j) used by vehicle k
$\theta_{5,e}$	the cost of each minute of early working time for driver e
$\theta_{6,e}$	the cost of each minute of late working time for driver e
$\theta_{7,e}$	the cost of each minute of extra working time for driver e

The criteria related to the quality of service and the working conditions are long-term objectives ; their conversion to economic costs is an estimation set by the user.

Note that a financial bonus might be paid to drivers for extra working time. In our study cases, the driver's working time was managed monthly, and such a bonus was difficult to

evaluate for a single day. From a mathematical point of view, the bonus can be included in $\theta_{5,e}$, $\theta_{6,e}$ or $\theta_{7,e}$.

2.4 Neighborhood evaluation

The procedure that evaluates some neighbors of a given solution is the core of our heuristic.

A solution is made of routes for drivers and vehicles. Each route is divided into trips from depot to depot for one crew and one vehicle.

A trip is a sequence of services and breaks that starts and ends at the depot. As the crew and the vehicle are a consistent pair along a trip, the entity trip is useful to describe and model several constraints of the CrewDARP. In addition, several efficient optimization procedures like the unified framework published in [Vidal et al., 2014], can be directly applied to trips, since crew and vehicle changes are not considered inside a trip.

Our solution approach, based on ALNS, is working on complete solutions, not on trips.

In ALNS operators insert or remove requests incrementally. For each request to insert, each position in the trips is considered. A position for a request is a position for the pick-up and one for the drop-off.

2.4.1 Request removal and insertion in single trips

As shown in Figure 2.1, several local operators can be modeled through successive sequence concatenations. A sequence of nodes v_i, v_j, v_k is denoted $v_i - v_j - v_k$ and the concatenation operator is denoted $\oplus$. Therefore, if the concatenation operator is applied in constant time, each neighbor is evaluated in constant time. The input is the evaluation of all the sequences of nodes of a solution. Since the concatenation operator can be used incrementally to evaluate any sequence, the preprocessing computation time is quadratic in the number of nodes ; *i.e.*, the number of sequences.

In [Vidal et al., 2014], a preprocessing treatment on a sequence of nodes is proposed to compute concatenations of sequences of nodes in constant time. Several attribute values of each sequence are maintained through the concatenation operator of two sequences. These attributes are the variables involved in the expression of the objective function and the constraints of many VRP variants. The key sequence attributes and the constant time concatenation operator are presented in Section 2.4.1.1 without lunch break constraints. In Section 2.4.1.2, we present what is wrong with lunch break constraints in [Vidal et al., 2014]'s formula, and we propose a way to overcome this issue. Finally, we explain how to exploit our preprocessing procedure in an optimization algorithm.

FIGURE 2.1 – Request deletion and insertion with sequence concatenation

Input : $\sigma = v_1 - v_2 - v_3 - v_4 - v_5 - v_6$; $\sigma' = v_7 - v_8$; $i \in P : i = v_3$ (pick-up), $i + |P| = v_5$ (delivery)

1. Preprocessing on all subsequences $v_i - v_{i+1} - ... - v_{i+n} : v_i - v_{i+n}$ in σ and σ' :

v_1 ; v_2 ; v_3 ; v_4 ; v_5 ; v_6 ; v_7 ; v_8
$v_1 \oplus v_2$; $v_2 \oplus v_3$; $v_3 \oplus v_4$; $v_4 \oplus v_5$; $v_5 \oplus v_6$; $v_7 \oplus v_8$
$(v_1 \oplus v_2) \oplus v_3$; $(v_2 \oplus v_3) \oplus v_4$; $(v_3 \oplus v_4) \oplus v_5$; $(v_4 \oplus v_5) \oplus v_6$
$((v_1 \oplus v_2) \oplus v_3) \oplus v_4$; $((v_2 \oplus v_3) \oplus v_4) \oplus v_5$; $((v_3 \oplus v_4) \oplus v_5) \oplus v_6$
$(((v_1 \oplus v_2) \oplus v_3) \oplus v_4) \oplus v_5$; $(((v_2) \oplus v_3) \oplus v_4) \oplus v_5) \oplus v_6$
$((((v_1 \oplus v_2) \oplus v_3) \oplus v_4) \oplus v_5) \oplus v_6$

2. Evaluate the move of request $i = \{v_3, v_5\}$ from σ to σ' :
   (a) Remove request $\{v_3, v_5\}$ from σ : Compute $(v_1 - v_2 \oplus v_4) \oplus v_6$.
   (b) Insert $\{v_3, v_5\}$ into σ' : Compute all combinations where the pick-up v_3 precedes the delivery v_5.

$(v_3 \oplus v_5) \oplus v_7 - v_8$;	$((v_3 \oplus v_7) \oplus v_5) \oplus v_8$;	$(v_3 \oplus v_7 - v_8) \oplus v_5$;
$((v_7 \oplus v_3) \oplus v_5) \oplus v_8$;	$((v_7 \oplus v_3) \oplus v_8) \oplus v_5$;	$(v_7 - v_8 \oplus v_3) \oplus v_5$

2.4.1.1 Sequence concatenation operator

For a given sequence of nodes $\sigma = (\sigma_1, ..., \sigma_{|\sigma|})$, we expect to obtain the following indicators defined in [Vidal et al., 2014] : $\mathcal{T}$ the sum of travel and service times, $\mathcal{E}$ the earliest completion time, $\mathcal{L}$ the latest starting time, $\mathcal{F}^{|TW}$ the feasibility statement regarding time windows, $\mathcal{C}$ the total distance, $\mathcal{Q}$ the final load, and Dur the minimum duration. To address the CrewDARP, additional attributes are defined : $\mathcal{P}$ the peak load, $\mathcal{V}$ the set of vehicle types, SF the minimized size of the crew, $\mathcal{S}$ the car-sharing indicator, CA the car-sharing feasibility and $\mathcal{F}$ the feasibility statement regarding all constraints.

Note that the complexity of the procedure for CrewDARP is V, which is assumed to be bounded.

The concatenation operator $\oplus$ is given by Equations (2.2)-(2.14) for two sequences σ and σ'.

$$\mathcal{T}(\sigma \oplus \sigma') = \mathcal{T}(\sigma) + t_{\sigma_{|\sigma|},\sigma'_1} + \mathcal{T}(\sigma') \quad (2.2)$$

$$\mathcal{E}(\sigma \oplus \sigma') = max\{\mathcal{E}(\sigma) + t_{\sigma_{|\sigma|},\sigma'_1} + \mathcal{T}(\sigma'), \mathcal{E}(\sigma')\} \quad (2.3)$$

$$\mathcal{L}(\sigma \oplus \sigma') = min\{\mathcal{L}(\sigma), \mathcal{L}(\sigma') - t_{\sigma_{|\sigma|},\sigma'_1} - \mathcal{T}(\sigma)\} \quad (2.4)$$

$$\mathcal{F}^{|TW}(\sigma \oplus \sigma') = F^{|TW}(\sigma) \wedge \mathcal{F}^{|TW}(\sigma') \wedge (\mathcal{E}(\sigma) + t_{\sigma_{|\sigma|},\sigma'_1} \leq \mathcal{L}(\sigma')) \quad (2.5)$$

$$\mathcal{C}(\sigma \oplus \sigma') = \mathcal{C}(\sigma) + c_{\sigma_{|\sigma|},\sigma'_1} + \mathcal{C}(\sigma') \quad (2.6)$$

$$\mathcal{Q}(\sigma \oplus \sigma') = \mathcal{Q}(\sigma) + \mathcal{Q}(\sigma') \quad (2.7)$$

$$\mathcal{P}(\sigma \oplus \sigma') = max\{\mathcal{P}(\sigma), \mathcal{Q}(\sigma) + \mathcal{P}(\sigma')\} \quad (2.8)$$

$$\mathcal{S}(\sigma \oplus \sigma') \equiv \mathcal{S}(\sigma) \vee \mathcal{S}(\sigma') \vee (\sigma_{|\sigma|} \text{ is a pick-up and } \sigma'_1 \text{ is not its delivery}) \quad (2.9)$$

$$\mathcal{V}(\sigma \oplus \sigma') = \{k \in \mathcal{V}(\sigma) \cap \mathcal{V}(\sigma') : \kappa_k \geq \mathcal{P}(\sigma \oplus \sigma') \text{ and } ((\neg cs_k \wedge \mathcal{S}(\sigma \oplus \sigma')) \vee cs_k)\} \quad (2.10)$$

$$SF(\sigma \oplus \sigma') = \max(SF(\sigma); SF(\sigma')) \quad (2.11)$$

$$CA(\sigma \oplus \sigma') \equiv \neg(\sigma_{|\sigma|} \text{ is a pick-up which forbids car-sharing and } \sigma'_1 \text{ is not its delivery}) \quad (2.12)$$

$$Dur(\sigma \oplus \sigma') = max(\mathcal{E}(\sigma \oplus \sigma') - \mathcal{L}(\sigma \oplus \sigma'); \mathcal{T}(\sigma \oplus \sigma')) \quad (2.13)$$

$$\mathcal{F}(\sigma \oplus \sigma') \equiv \mathcal{F}(\sigma) \wedge \mathcal{F}(\sigma') \wedge \mathcal{F}^{|TW}(\sigma \oplus \sigma') \wedge \{\mathcal{V} \neq \emptyset\} \wedge CA(\sigma \oplus \sigma') \wedge Dur(\sigma \oplus \sigma') \leq T_{max} \quad (2.14)$$

The initial sequences are single nodes with the following enforced values, with $\sigma = v$: $\mathcal{T}(\sigma) = s_v, \mathcal{E}(\sigma) = es_v + s_v, \mathcal{L}(\sigma) = ls_v, \mathcal{F}^{TW}(\sigma) \equiv True, \mathcal{C}(\sigma) = 0, \mathcal{P}(\sigma) = \mathcal{Q}(\sigma) = (-q_v$ if v is a drop-off node and q_v otherwise), $\mathcal{V}(\sigma) = B_v$, $\mathcal{SF}(\sigma) = sf_v$, $\mathcal{S}(\sigma_0) \equiv False$ and $\mathcal{F}(\sigma_0) \equiv True$.

2.4.1.2 Breaks

Breaks which have to be taken at the depot are modeled as single nodes ; one for each driver with a high profit. The distance between two breaks of different drivers is set to $-\delta$ in order to schedule simultaneous breaks of the drivers of the same crew. Breaks scheduled at the beginning or the ending of a working day are discarded from the working time evaluation.

The following part of this section is dedicated to breaks taken on the way.

Original erroneous formulas (2.15)-(2.25) are given with a counterexample. A not constant time formulation of the concatenation operator is derived from the proposed formula. A second approach which does not these formula has been implemented and is described below.

Three cases are considered if the break was present : $case\,1$ if the break was in σ, $case\,2$ if the break was between σ and σ' and $case\,3$ if the break was in σ'.

$$\mathcal{E}'(\sigma \oplus \sigma') = min(\{\mathcal{E}'_{case\,i} | \mathcal{F}'_{case\,i} = true, \forall i \in \{1,2,3\}\} \cup +\infty) \quad (2.15)$$

$$\mathcal{L}'(\sigma \oplus \sigma') = max(\{\mathcal{L}'_{case\,i} | \mathcal{F}'_{case\,i} = true, \forall i \in \{1,2,3\}\} \cup -\infty) \quad (2.16)$$

$$\mathcal{F}'(\sigma \oplus \sigma') \equiv \{\mathcal{F}'_{case\,1} \wedge \mathcal{F}'_{case\,2} \wedge \mathcal{F}'_{case\,3}\} \quad (2.17)$$

$$\mathcal{E}'_{case\,1} = max\{\mathcal{E}'(\sigma) + t_{\sigma_{|\sigma|},\sigma'_1} + \mathcal{T}(\sigma'), \mathcal{E}(\sigma')\} \quad (2.18)$$

$$\mathcal{E}'_{case\,2} = max\{\mathcal{E}(\sigma) + t_{\sigma_{|\sigma|},\sigma'_1} + \delta + \mathcal{T}(\sigma'), eb + \delta + \mathcal{T}(\sigma'), \mathcal{E}(\sigma')\} \quad (2.19)$$

$$\mathcal{L}'_{case\,1} = min\{\mathcal{L}'(\sigma), \mathcal{L}(\sigma') - t_{\sigma_{|\sigma|},\sigma'_1} - \mathcal{T}(\sigma) - \delta\} \quad (2.20)$$

$$\mathcal{L}'_{case\,2} = min\{\mathcal{L}(\sigma), lb - \mathcal{T}(\sigma), \mathcal{L}(\sigma') - t_{\sigma_{|\sigma|},\sigma'_1} - \delta - \mathcal{T}(\sigma)\} \quad (2.21)$$

$$\mathcal{L}'_{case\,3} = min\{\mathcal{L}(\sigma), \mathcal{L}'(\sigma) - t_{\sigma_{|\sigma|},\sigma'_1} - \mathcal{T}(\sigma)\} \quad (2.22)$$

$$\mathcal{F}'_{case\,1} \equiv \mathcal{F}'(\sigma) \wedge \mathcal{F}(\sigma') \wedge (\mathcal{E}'(\sigma) + t_{\sigma_{|\sigma|},\sigma'_1} \leq \mathcal{L}(\sigma')) \quad (2.23)$$

$$\mathcal{F}'_{case\,2} \equiv \mathcal{F}(\sigma) \wedge \mathcal{F}(\sigma') \wedge (\mathcal{E}(\sigma) \leq lb) \wedge (\mathcal{E}(\sigma) + \delta + t_{\sigma_{|\sigma|},\sigma'_1} \leq \mathcal{L}(\sigma')) \quad (2.24)$$

$$\mathcal{F}'_{case\,3} \equiv \mathcal{F}(\sigma) \wedge \mathcal{F}'(\sigma') \wedge (\mathcal{E}(\sigma) + t_{\sigma_{|\sigma|},\sigma'_1} \leq \mathcal{L}'(\sigma')) \quad (2.25)$$

The initial state with single nodes v is set with $\mathcal{E}'(\sigma) = +\infty$, $\mathcal{L}'(\sigma) = 0$ and $\mathcal{F}'(\sigma) \equiv False$.

Let us consider $\sigma = \{v_1, v_2, v_3, v_4\}$ and $\sigma' = \{v_5\}$ with null service times and with the respective earliest and latest start times

$$\{[10, 20]; [70, 80]; [100, 110]; [150, 170]; [160, 170]\}.$$

A one-hour break is required with duration $\delta = 60$ and starting time in $[eb = -\infty, lb = +\infty]$. Travel times between all pairs of nodes are equal to 10.

$\mathcal{E}'(\sigma)$ is equal to 150 and corresponds to a starting time at 10 with the break between v_1 and v_2. $\mathcal{L}'(\sigma)$ is equal to 20 and corresponds to 170 as the completion time with a break between v_3 and v_4.

Formula (2.15)-(2.25) set $\mathcal{L}'(\sigma \oplus \sigma')$ to 20 and a feasible statement since $\mathcal{E}'(\sigma) = 150$. However, this setting is not consistent since $\mathcal{L}'(\sigma \oplus \sigma') = 20$ means a break between v_3 and v_4 and $\mathcal{E}'(\sigma) = 150$ a break between v_1 and v_2. In Appendix, Table A.1, the intermediate results are detailed.

From the counterexample, we understand that if $\mathcal{L}'$ and $\mathcal{E}'$ correspond to different break positions, the formula does not work properly. To obtain the necessary information for each feasible break position, the nondominated set of pairs $(\mathcal{E}', \mathcal{L}')$ has to be kept. The dominance of $(\mathcal{E}'_1, \mathcal{L}'_1)$ over $(\mathcal{E}'_2, \mathcal{L}'_2)$ is evaluated by Equation (2.26).

$$(\mathcal{E}'_1 \leq \mathcal{E}'_2) \wedge (\mathcal{L}'_1 \geq \mathcal{L}'_2) \wedge (\max\{\mathcal{E}'_1 - \mathcal{L}'_1, \mathcal{T}_1 + \delta\} \leq \max\{\mathcal{E}'_2 - \mathcal{L}'_2, \mathcal{T}_2 + \delta\}) \tag{2.26}$$

Of course, the maximal number of nondominated pairs depends on the length of the sequence, and such procedure loses the constant time evaluation property.

In order to keep a constant time complexity, we develop a second approach where breaks are modeled as single nodes, similarly to breaks at the depot but with a floating location. The floating location means that the break is considered located at the previous node in the sequence. This sequence dependent location implies specific definitions and concatenation procedures.

The travel time from a customer node to a break is set to zero and to $-\delta$ between two breaks. The travel time from a break and to a customer node is dynamically set depending on the running concatenation.

Let consider i the rank of the last customer node in σ and j the rank of the first customer node in σ'. These two values are included in the list of attributes of each sequence. The concatenation operator is now decomposed into the three consecutive concatenations of Equation (2.27).

$$\sigma \oplus \sigma' = (\sigma_1 - \ldots - \sigma_i) \oplus (\sigma_{i+1} - \ldots - \sigma_{|\sigma|}) \oplus (\sigma'_1 - \ldots - \sigma'_{j-1}) \oplus (\sigma'_j - \ldots - \sigma'_{|\sigma'|}) \tag{2.27}$$

Thus, only one 'dynamic travel time' has to be considered between the break node σ'_{j-1} and the customer node σ'_j at the third concatenation. The value of this travel time is set to t_{σ_i,σ'_j}. This general formula can be easily extended to the cases where i and j do not exist or are extreme nodes of σ and σ'.

2.4.2 Request insertion and removal in synchronized multitrips

In this section, we describe how to evaluate the insertion and removal of a request from a current solution. Actually, the modification of a trip can impact the feasibility and schedule of related trips that involve the same drivers.

By fixing some decisions, an optimized schedule is computed as the solution of the so-called timing problem. The decisions fixed are the crew and vehicle assignment to customers, the sequence of visits for each driver and the position of breaks. The timing problem thus defined, is formulated by the linear program (2.28)-(2.33) on a single graph (V^T, A^T) with V^T being the set of trips and A^T being the set of arcs between trips. Trip u is connected to trip v if at least one driver or the same vehicle is involved in u before v. $V^T(e)$ is the set of trips with driver e. Note that for trip t, no vehicle is assigned when (V^T, A^T) is built. From the attributes computed by the preprocess of Section 2.4.1, three parameters are computed for trip u : the minimal duration $Dur(u) = \max(\mathcal{E}(u) - \mathcal{L}(u); \mathcal{T}(u))$; the earliest starting time $Est(u) = E(u) - Dur(u)$; and the latest starting time $Lst(u) = \mathcal{L}(u)$. If an 'on-the-way' break is assigned to u, $\mathcal{E}'(u)$ and $\mathcal{L}'(u)$ are used instead of $\mathcal{E}(u)$ and $\mathcal{L}(u)$.

$$Timing(D,t): \min_{(V^T,A^T)\in\mathcal{G}(D,t)} \min\{\sum_{e\in D}(\theta_{5,e}\gamma_e + \theta_{6,e}\beta_e + \theta_{7,e}\zeta_e)\} \tag{2.28}$$

$$Est(u) - Dur(u) \leq x_u \leq Lst(u)\ ,\forall u \in V^T \tag{2.29}$$

$$x_v \geq x_u + Dur(u)\ ,\forall (u,v) \in A^T \tag{2.30}$$

$$\beta_e \geq x_u + Dur(u) - lw_e\ ,\forall e \in E, u \in V^T(e) \tag{2.31}$$

$$\gamma_e \geq ew_e - x_u\ ,\forall e \in E, u \in V^T(e) \tag{2.32}$$

$$\zeta_e \geq (x_u + Dur(u) - x_v) - (lw_e - ew_e)\ ,\forall e \in E, u, v \in V^T(e) \tag{2.33}$$

When a new solution obtained by insertions or removals raises vehicle and crew incompatibility or trip overlapping, the reassignment of crew to requests and rescheduling of the sequences of visits are investigated. The extension of the timing problem to these decisions becomes combinatorial and is addressed by Algorithm 1 (A1).

Algorithm 1 computes the best insertion position for a request in a current solution. The new trip obtained after insertion is evaluated at Line 2 (A1-L2) through the procedure described in Section 2.4.1. At this step, the crew and the vehicle are not considered.

If the size of the crew matches the trip requirement (A1-L5), then the trip with crew and vehicle is evaluated by Algorithm 2 (A2). The compatibility of the crew is checked

(A2-L3), the timing of the complete solution is revised (A2-L5) and an available vehicle is selected (A2-L7 and A2-L1).

If needed, the crew was completed through instructions A1-L9 to A1-L17. For each driver, a marginal cost is computed at A1-L10 by solving the timing problem solved at A2-L5 for each feasible graph (V^T, A^T) with the set of drivers $D = d \cup D(u)$. Every new sequence of trips obtained after the modification of t is considered. In practice, the number of graphs (V^T, A^T) is very limited. All subsets of drivers to add (A1-L13) are evaluated, and the search is stopped as soon as a feasible crew is found.

If the current crew assigned to t can be reduced, driver removal is performed from instructions A1-L18 to A1-L27. The marginal cost of removing each driver from $D(u)$ is evaluated at A1-L20 by solving the timing problem (A2-L5 for each feasible graph (V^T, A^T) with the set of drivers $D = D(u) \setminus d$. The first subset of drivers to remove that is evaluated with a feasible solution is selected at A1-L25. If this procedure fails to remove drivers and find a feasible solution, the feasibility of the solution with the current crew $D(u)$ is checked at A1-L28.

Finally, the best insertion position (trip u and position in trip pos) is **return**ed.

The neighborhood size limitation is obtained by a strict selection of the positions (u, pos) to evaluate with Algorithm 1. Let us consider the pick-up j to insert after service k in trip t. All the configurations where the pick-up j is placed just after activity k and the drop-off $j + |P|$ is placed at any position after j are evaluated by Algorithm 1 if one of the two following conditions is satisfied.

The first condition is that request j is closely related to request i of service k. The second condition is the result of a random trial ; the position is selected with a probability π_{Sel} if less than Sel_{max} positions have already been selected by this means.

In the relatedness function (2.34), $\Gamma(i, j)$ denotes the permutations of pick-up and drop-off nodes of requests i and j. The transportation time $T_{a,b,c,d}$ penalizes requests far apart. The minimal waiting time $W_{a,b,c,d}$ penalizes requests between which other tasks can be scheduled. The parameter α allows us to balance the importance of the two criteria. Note that any infeasible permutation (a, b, c, d) is evaluated by $+\infty$.

$$Rel(i, j) = \min\{W_{a,b,c,d} + \alpha T_{a,b,c,d} : (a, b, c, d) \in \Gamma(i, j)\} \tag{2.34}$$

The Rel_{max} requests with the lowest relatedness score with i allow us to evaluate the insertion position. The relatedness values are computed once as a preprocess.

Thus, the number of positions evaluated for the pick-up node is bounded by $Sel_{max} + Rel_{max}$.

2.5 Adaptive large neighborhood search

To improve an initial solution $\mathcal{S}_0$, we designed a heuristic approach based on the metaheuristic ALNS(adaptive large neighborhood search). Our ALNSimplementation for the CrewDARP is described in Algorithm 3. The lists of removal ($\mathcal{R}$) and insertion ($\mathcal{I}$) procedures are described in Sections 2.5.1 and 2.5.2. Three subsets of removal operators are defined : very large operators ($r = 1$) ; large operators ($r = 2$) ; and local operators ($r = 3$).

At Line 6 of Algorithm 3, a new solution $\mathcal{S}_{new}$ is obtained from the current $\mathcal{S}$ by an insertion procedure (P_4) followed by a removal operator (P_{r}).

Removal and insertion procedures are selected at Lines 4-5 with a probability mass function (pmf). The weights of the last used removal and insertion procedures (P_r and P_4) are updated through an adaptive mechanism at Lines 9, 13. At Lines 27-34, a similar update is performed for the last used large and very large operators P_2 and P_1, respectively, when the best current solution is improved. At each iteration, the score $\pi_{\mathcal{P}}$ is increased by σ according to the quality of the solution $\mathcal{S}_{new}$; $(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ are positive parameters. Every n iterations, the weights Π_P that define the selection probabilities w_P of every procedure are updated with Lines 27-34, where $\gamma \in [0, 1[$ is a parameter. The probabilities w_P define the probability mass functions used for the procedure selection. Therefore, large values of γ increase the impact of the increment in the selection probability of every operator.

After NN consecutive local removals without any improvement of the current solution $\mathcal{S}$ (Lines 19-25), a large removal operator is applied. After DN descents without any improvement of the best-known solution $\mathcal{S}_{best}$, a very large removal operator is applied.

In several state-of-the-art ALNS algorithms, a diversification mechanism similar to simulated annealing is implemented [Azi et al., 2014,Pisinger and Ropke, 2005]. On our set of instances of the CrewDARP, this setting did not provide good results. Due to possible very large cost variations, the main difficulty in the classical framework and Metropolis function was to define a robust value for the parameters that control the temperature variations. The proposed implementation with descents and large (and very large) destruction operators gave better average results over all the instances than different settings of the classical approach (see Figure 2.2).

2.5.1 Destruction operators

We used seven operators, the three first ones : Random, Related and Worst removals are direct adaptations of already published operators. Related-trip and worst-trip operators apply the same ideas to trips instead of nodes. The driver and vehicle operators remove some activities assigned to the same driver or the same vehicle. Usually, the last four removals involve a larger part of the solution compared to the other operators. All these

operators are described in the following sections.

For each operator, some elements are removed. The number of elements to remove is bounded by a random number generated at each iteration and controlled by a fixed range.

The values for these parameters are given in Section 2.8, and the efficiency of each operator is also validated in this section.

2.5.1.1 Local Operators

The random-removal operator is mainly used for diversification, as r requests are randomly removed from the current solution.

The related-removal operator was proposed in [Shaw, 1997] and improved in [Pisinger and Ropke, 2005] for the PDPTW and is often called Shaw removal. The idea is to remove requests that have relatedness and can be easily exchanged or scheduled in the same trip.

From a reference request i, another request is randomly selected according to its relatedness measure from i as described by Equation (2.34). This procedure is repeated to select a fixed number of requests. At each iteration, the reference request is uniformly selected among the requests already selected for removal (in the first iteration, all requests can be selected).

In addition to the set of requests that could be sequenced, we aim to select requests that could be exchanged. Good candidates for exchanges are permutations that are unfeasible because of time window constraints ; *i.e.*, at least two service times have to overlap. Therefore, the evaluation of such permutations is set to 0 in the relatedness function.

The related-removal operator is described in Algorithm 4). In the common implementation of the relatedness function, the value depends on the current solution and is updated for every request after each removal (Line 7). Motivated by tight time windows in practice, our implementation considers fixed $Rel(i, j)$ values that are computed as preprocessing.

The worst-removal operator removes the requests with the greatest marginal cost. For each request, the marginal cost is the difference between the objective value function and the value after deletion.

The procedure is similar to the related-removal operator, where the list H of requests is sorted in nondecreasing order of marginal costs.

After request removal, the marginal costs of the requests served by the same driver or vehicle are updated.

2.5.1.2 Large operators

The related-trip removal removes $Trip_{remove}$ trips from the current solution. The procedure is similar to the related-removal operator but instead of the related-request j selected at Line 10 in Algorithm 4, the complete trip that contains j is removed. The number of trips removed is bounded by the parameter $Trip_{remove}$ instead of Req_{remove}.

The worst-trip removal is inspired by the worst-removal operator, but at each iteration, the complete trip is removed instead of only the selected request.

2.5.1.3 Very large operators

The driver trip-removal operator randomly removes (uniform selection) $Driv_{remove}$ driver activities. i.e., remove all trips where the selected drivers are assigned.

The vehicle trip-removal operator randomly removes Veh_{remove} vehicle activities. i.e., remove all trips where the selected vehicles are assigned.

2.5.2 Insertion operators

The reconstruction operators usually insert requests previously removed by a destruction operator. In addition to these requests, our reconstruction procedures also attempt to insert requests (three requests or less in our experiments) that were not covered in the original solution.

Insertion into trips is evaluated by the procedure of Section 2.4.

Three insertion operators have been implemented.

The list order-insertion operator is an iterative procedure that inserts requests one by one picked from a list. Several lists are used to generate different solutions. The first list of requests follows the increasing order of earliest pick-up times. Other lists are randomly generated. The number of lists has been set, after some preliminary experiments, to the number of requests to insert divided by three.

The best-insertion operator is an iterative procedure that inserts at each iteration the request with the best trip insertion evaluation.

The 2-regret-insertion operator inserts at each iteration the request with the greatest gap between its best=trip insertion evaluation and its second-best-trip insertion evaluation. Trip insertion is evaluated as the evaluation of the best insertion position in the trip. The insertion with the best evaluation of the selected request is performed.

2.6 Constructive heuristics

Constructive heuristics that imitate those of the human planner are used to find an initial solution. The second heuristic is a basic improvement of the first heuristic and is used to benchmark ALNS.

2.6.1 hla – Human-Like Algorithm

In this approach, the solution is built through two phases. In the first phase, some feasible trips are built for each crew and vehicle for the day. Soft constraints on the latest drivers' working time, lw_e, are considered hard constraints. In the second phase, these constraints are relaxed, and the remaining requests are inserted in existing trips.

2.6.1.1 First phase

The first phase is formally described in Algorithm 5. At each iteration, a new crew (drivers and vehicle) is composed of Function find_Crew_Hard at Line 5 according to a request selected for insertion. As described in Section 2.6.1.3, the crew created implies little extra working time for drivers and a short trip to reach the request from the depot. In our implementation on Lines 2 and 4, the remaining request with the earliest starting pick-up time is selected as the first request of the new trip. Function make_Trip at Line 7 builds the single-request trip with the selected crew. Function add_Hard extends the trip (Lines 8- 14) until no request can be added at the end of the trip without any working time window violation. This function is detailed in Section 2.6.1.4.

Note that drivers and vehicles that are already assigned to a trip can be selected for another trip. The requests are considered for insertion in increasing order of their earliest starting pick-up time. This order has been empirically shown to be the most efficient, as it copes with the push-back insertion of requests in trips.

2.6.1.2 Second phase

The second procedure tries to insert each request that has not been inserted during the previous phase. Both options are considered ; create a trip with a new crew or insert at the end of an existing trip. The procedures used in Algorithm 6 are similar to the first phase, but the working time bound lw_e is imposed as a soft constraint as described in Sections 2.6.1.3 and 2.6.1.4.

2.6.1.3 Compose a crew for a request

The functions find_Crew_Hard/Soft select the crew with the best score S to serve a request starting from the depot. As the sequence is fixed (variables $x = x_0$ define : depot $\rightarrow$ pick-up $\rightarrow$ drop-off $\rightarrow$ depot), the evaluation of $Z(w, x, y, z)$ depends on timing. In Equation (2.35), the second member measures the maximal gap between working time

limits of the crew members to compose persistent crews. The parameter ζ balances the weight of both indicators.

$$S(\{e_1, \ldots, e_n\}) = \max_{(w,x_0,y,z)\in\mathcal{P}} Z(w, x_0, y, z) + \zeta \max_{1\leq i,j\leq n} |lw_i - lw_j| \tag{2.35}$$

In the 'Hard' version of the procedure, the set of feasible solutions $\mathcal{P}$ is reduced since all the latest working times lw_e are imposed as hard constraints.

At the depot, a vehicle compatible with the request requirements has to be available from the beginning of the trip until the end of the day. Otherwise, the crew cannot be selected.

Note that no driver can be selected if he or she already has some activities already scheduled after the request considered for evaluation. This limitation is acceptable, as the requests are considered in their increasing earliest starting time, and tight service time window constraints are expected.

If a driver has finished a trip before, his or her release time is fixed ; when a trip is closed (back to the depot), its timing is fixed to the best solution.

Note also that breaks are inserted if required by the new insertion.

The function makes_Trip applies the best solution evaluated by find_Crew_Hard/Soft for the request and the crew.

2.6.1.4 Request insertion

A request i is inserted at the end of a trip with last request j. All insertion positions for the pick-up and the drop-off of i are evaluated such that the pick-up of j precedes the pick-up of i. The relative order of the original trip is not modified. Thus, car sharing among several requests can be accepted. The feasible permutation with lower marginal travel time is selected.

Note that breaks are inserted if required. The functions add_Hard and add_Soft are similar, but the parameter lw_e (latest ending working time of a driver) is considered a hard or soft bound.

2.6.2 hla+ – Improved constructive heuristic

We implemented a deterministic multistart version of HLA where a local search procedure is applied on the list of requests. Twist operators – operators that reverse a subsequence in the sequence – are applied to all positions. The length of the subsequences visited ranges in a limited set of values ; $[2, 6]$ in our experiments. The length is reinitialized at the minimal value when no improvement is obtained with the maximum value. The first improvement strategy guides the local search.

2.7 Real-world data set

Data have been provided by Lomaco and come from four ambulance companies, with 10, 6, 25 and 61 depots. The main characteristics of the 17 original daily instances provided are described in Table 2.5 and denoted set A.

TABLE 2.5 – 17 original instances

name	requests	drivers	vehicles	depots
A-10-1	128	20	16	10
A-10-2	97	20	13	10
A-10-3	118	24	17	10
A-10-4	88	21	14	10
A-10-5	117	21	14	10
A-06-1	189	27	22	6
A-06-2	183	23	20	6
A-06-3	185	28	21	6
A-06-4	154	22	17	6
A-25-1	144	26	21	25
A-25-2	199	29	25	25
A-25-3	312	42	34	25
A-61-1	1178	196	165	61
A-61-2	1628	267	200	61
A-61-3	1903	300	230	61
A-61-4	941	177	144	61
A-61-5	1178	201	165	61

Time windows are set to 40 minutes following the procedure described in Section 2.3.4. Three types of vehicles are available : ambulances, LHVs and taxis with different parameters. Cleaning time is never required.

In this set, the lunch break is taken at the depot, and sets of constraints Φ_i and Ψ_i of request preferences are empty (this option is not included in the current version of the software).

The default weights of the objective function are defined in Table 2.6.

TABLE 2.6 – Objective function setting

quality of service $\theta_{1,i}$	operational cost $\theta_{3,i}$; $\theta_{4,i,j,k}$	quality of working life $\theta_{5,e} = \theta_{6,e} = \theta_{7,e}$	customer preferences $\theta_{2,i,\phi}$
5,000	pr_i ; $\alpha_k c_{ij}$	1	1

To assess the relevance of our algorithm in different contexts, these instances are slightly

modified in some sections.

2.8 Numerical experiments

Several computational experiments have been conducted following three objectives. The best setting of our algorithm is evaluated in Section 2.8.1. In Section 2.8.2.2, we measure the performance of our ALNSagainst the human-like algorithm on the set of real data described in Section 2.7. The robustness of our algorithm and setting is assessed in Section 2.8.3 by a comparison against recently published algorithms designed for similar problems : the RDARP ([Lim et al., 2017]) and the MTVRPTW ([Azi et al., 2014]).

All the algorithms have been implemented in JAVA to be directly integrated in the full-web platform of the company. Experiments have been conducted on 2.4 GhZ processors running under CentOS.

Instances and detailed results are available at `https://portail.emse.fr/hpc-cis-projet/i4s_public/rootpage.html`.

The algorithm parameters of Table 2.7 have been optimized on set A of instances and kept for all the experiments.

TABLE 2.7 – ALNS Parameters

i_{max}	γ	NN	DN
$30,000$	$0,8$	$3.94\|P\|^{0.13}$	$1.57\|P\|^{0.24}$
α	ζ	$Trip_{rem}$	Req_{rem}
0.02	0.02	$0.09\|T\|$	$Uniform(1,9)$
$Driv_{rem}$	Veh_{rem}	$(\sigma_1,\sigma_2,\sigma_3,\sigma_4)$	Rel_{max}
$0.2\|E\|$	$0.2\|K\|$	$(30,60,200,200)$	$48.81\|P\|^{-0.35}$
π_{Sel}	Sel_{max}	π_{P_j}	Π_P
$3.33\|P\|^{-0.4}$	$0.94\|P\|^{0.4}$	$0 \ \forall j$	$1000 \ \forall P$

2.8.1 ALNS implementation

In this section, we empirically study the relevance of our ALNS implementation. We do not detail how we set the values of all the parameters. First, we justify here why we did not implement classical temperature management in ALNS. We also study the impact of each removal/insertion operator.

2.8.1.1 Metaheuristic setting

From Figure 2.2, one can state that ALNS clearly outperforms ALNS(T) with the classical temperature control (similar to simulated annealing) used by several authors [Azi

et al., 2014,Lim et al., 2017,Pisinger and Ropke, 2005].

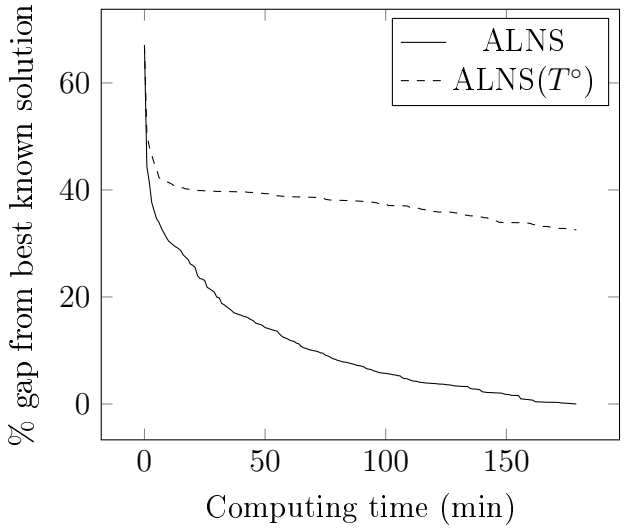

FIGURE 2.2 – Comparison of two temperature control management strategies

The average gap from the best-known solution over the 17 A-instances is reported every minute for both implementations.

The main difficulty with the classical formulation is to deal with a very wide range of solution costs. In addition, solution costs can jump quickly after only one move, since the request covering benefit is usually set to a very high value. This context makes the value of the temperature and its variation difficult to set. This difficulty is increased when the algorithm is designed to solve instances with different characteristics in size and constraint tightness.

2.8.1.2 Neighborhood operator relevance

In each run, one removal or insertion operator was deactivated. The average, maximum and minimum (over the 17 A-instances) percentage deviations compared to the run with all operators are given in Table 2.8. The deviation is computed as the cost of the solution obtained by ALNSwithout one operator minus the cost with ALNS, divided by the cost with ALNS.

These values show that each operator improves the performance of the algorithm on average between 11.4% and 24.5%. Moreover, each operator is critical for at least one instance since the maximum improvement on a single instance ranges from 75.8% to 164.0%. However, removing one operator is sometimes better for 7 instances, compared to 17 in the 'Best' column with VehicleTrip. The improvement without the operator can reach 51.4% for RandomRequest on one instance. However, the loss is greater than the benefit for any removal of the operator.

TABLE 2.8 – Removal and insertion operator utilities over 17 A-instances

Discarded operator	avg %	max %	min %	best /17	time(s)
RelatedRequest	16.8	80.5	-4.8	4	3882
RandomRequest	12.3	122.0	-51.4	6	3901
WorstRequest	15.9	98.6	-11.4	4	3869
RelatedTrip	24.3	164.0	-42.2	3	3783
WorstTrip	13.6	78.5	-5.7	4	3869
DriverTrip	11.4	81.9	-6.7	5	3794
VehicleTrip	21.6	159.1	-6.7	7	3867
Basic	24.5	75.8	-0.5	2	3901
2-regret	15.4	80.7	-19.3	5	3821
Best	13.0	86.3	-16.5	5	3813

2.8.2 CrewDARP

First, we check that ALNS is able to solve problems under different weights on objective criteria and that the solutions obtained are consistent with these weights. Then, ALNS is compared against HLA and HLA+.

Finally, ALNS is applied under different settings of the problem, and the relevance of the crew reconfiguration on real-world instances is shown.

2.8.2.1 ALNS under different optimization settings

ALNS is run on each A-instance with three objective functions : (i) with $C1$ the default objective function setting of Table 2.6 ; (ii) with $C2$ where the operational cost is multiplied by 600 ; and (iii) with $C3$ where the quality of working life criteria are multiplied by 600.

The results in Tables 2.9-2.11 show that ALNSis able to follow the modifications in the weights of the different criteria. The objective value (Ob. V), and the value for each criterion is given for each instance.

Settings $C2$ and $C3$ provide bad results in terms of request coverage, which is usually the most important criterion. Therefore, $C1$ seems to be a good default setting, and its performance on 'Distance' suggests that this criterion is well optimized. Nevertheless, the working quality of life is more penalized. However, the values reported are in minutes and have to be dispatched over the set of drivers. In practice, the working time of drivers is often modified during the day.

Values displayed are rounded to the unit for minutes ('earlier', 'later' and 'longer' columns) and to the first decimal place for columns 'distance', 'loss' and 'objective value'.

2.8.2.2 ALNS versus human-like heuristics

In Table 2.12, the best and average results after three runs of ALNS on A-instances are reported. All the criteria except 'U. R', are aggregated in 'Cost'.

For each instance, the results are stable if we consider the variation in 'U. R' independently. Few unserved requests greatly increase (5000 for each request) the objective value.

Even for instances in the same series, large discrepancies are observed. Best and average evaluations are almost doubled from A-25-2 to A-25-3. However, this difference is explained by significantly different quantities of requests, vehicles and drivers.

The average computing times 'T(s)' in seconds over three runs for 30,000 iterations show that the running time is linearly correlated with the number of requests with a factor close to 3.

For more details of each term of the objective function, see B.2 and B.1.

The results obtained by a single run of HLA and HLA+ which are both deterministic algorithms, are displayed in Table 2.13. Even if HLA+ strongly improves HLA on every instance, HLA+ is outperformed by ALNS.

In addition, long computing times on the 'A-61' series show that the complete HLA+ descent is not relevant as a preprocess of ALNS.

As ALNS is initialized with the solution of HLA, we can state that ALNS is able to improve bad initial solutions and therefore that the impact of the initial solution is low after 30,000 iterations of ALNS.

For more details of each term of the objective function, see B.3 and B.4.

2.8.2.3 Modified real data

In order to show the interest of crew recomposition, ALNS has been executed with a limitation to a single trip for each driver and vehicle. This algorithm is denoted 1-Trip-ALNS.

The average of three runs of ALNS and one run of 1-Trip-ALNS are compared over all instances in Table 2.14. The average and maximal number of trips 'Avg_{tr}' and 'max_{tr}' indicate that fewer than two trips are performed by drivers on average, but some drivers perform five trips in larger instances. Note that the average number of trips is computed on the working drivers, i.e., the workers with at least one trip.

On smaller instances, 1-Trip-ALNS is able to obtain results close to ALNS, but on larger instances (A-61 series), more requests are unserved by 1-Trip-ALNS. The gap is small in the number of covered requests and 1-Trip-ALNSis sometimes able to find better solutions (instances A-10-5 and A-25-3). However, unserved requests are a very unlikely situation, and planners may strongly deteriorate the plan to satisfy requests. In the case of several planners, one of them may also drive a vehicle to serve the request themselves.

For more details of each term of the objective function, see B.5.

Two modes for lunch breaks have been implemented, at the depot – the default mode – and on the way. Breaks on the way dominate breaks at the depot, since breaks at the depot are allowed with this setting. In practice, several companies enforce breaks at the depot because of the working quality condition. Additionally, the depot is usually close to a hospital, which generates several demands.

The comparison of one run with each mode in Table 2.15 shows that breaks 'on the way' slightly dominate breaks at the depot except on instance A-06-4. For this instance, the difficulty for ALNS to optimize breaks on the way is raised.

For more details of each term of the objective function, see B.6.

In the original instances, preference sets Φ_i and Ψ_i are empty. To assess whether ALNS works with such parameters and to provide a benchmark for future research, Table 2.16 compares the results obtained without consideration of customer preferences. A list of preferences was randomly generated according to proportions given by some Lomaco customers.

The results show that some customer preferences have to be balanced with the request coverage criterion.

For more details of each term of the objective function, see B.7.

2.8.3 Benchmark on related problems

ALNS is compared to state-of-the-art algorithms designed for RDARP and MTVRPTW, which are the two closest special cases of CrewDARP.

2.8.3.1 RDARP

The RDARP instances represent one year of sanitary transportation in Hong Kong. Each instance is one day. The 116 'Holidays' instances contain fewer requests than the 249 'Non-Holidays' instances, respectively, 53.8 and 168.6 requests on average.

In RDARP, three criteria are evaluated : the number of uncovered requests (multiplied by 10,000), the traveled distance (multiplied by 100) and the workload equity among drivers. The last criterion counts for about one thousand of the objective value and is discarded from our comparison.

Additionally, assistants in RDARP occupy the same set of places as patients. Therefore, we modified the way to compute the number of available places in our algorithm by replacing Equation (2.10) with (2.36).

$$\mathcal{V}(\sigma \oplus \sigma') = \{k \in \mathcal{V}(\sigma) \cap \mathcal{V}(\sigma') : \kappa_k \geq \mathcal{P}(\sigma \oplus \sigma') + SF(\sigma \oplus \sigma') - 1 \text{ and } ((\neg cs_k \wedge \mathcal{S}(\sigma \oplus \sigma')) \vee cs_k)\} \tag{2.36}$$

ALNS results are compared to [Lim et al., 2017]'s (SOA algorithm) results through ten runs described in Table 2.17. The number of uncovered requests is the first objective, and the distance traveled is the second objective.

Since machines are similar for our and [Lim et al., 2017] experiments, the time limits for each run are equal for both algorithms : 1800 seconds if $|P| \leq 50$, 3600 s if $50 < |P| \leq 100$, and 7200 s for $|P| > 100$. ALNS dominates SOA on average over all instances and on larger instances (non-holidays). Additionally, ALNS was able to improve several best-known solutions, 158 non-holidays and 116 holidays, from line % dominance, which counts the percentage of instances for which one algorithm determines a strictly better solution (or average solution) than the other algorithm. As ALNShas been designed for large instances of a generalization of SOA, the results obtained on RDARP are satisfying.

SOA performs better on smaller instances since, in our opinion, the assignment of drivers to trips is optimized for all neighbor solutions. These approaches provide better intensification but are time consuming on large instances.

2.8.3.2 MTVRPTW

The MTVRPTW is a special case of the CrewDARP, where pick-up nodes are at the depot. The key feature is T_{GT}, which bounds the length of trips until the last customer and not back to the depot. The concatenation equations for two sequences of nodes of Section 2.4.1 are adapted to this case with an additional attribute $LT(\sigma)$, which measures the accumulated loading time over σ.

Small and large instances of the MTVRPTW are used. The first set is provided by [Lim et al., 2017], and the second set is described by [Azi et al., 2014]. Three classes of instances are considered with customers clustered 'C', uniformly distributed 'R' and mixed 'RC'.

Tables 2.18 and 2.19 show that ALNSworks slightly better than the algorithm published in [Lim et al., 2017] over ten runs. ALNSis often able to find optimal known solutions within a maximal computing time set to 1.15 times the 'time to best' provided by [Azi et al., 2014].

On large instances, a single run of ALNS is compared to a single run of the algorithm from [Azi et al., 2014] in Table 2.20. The number of iterations of ALNSwas set for each class of instances to obtain a running time close to their performance on a 3.1 GHZ (16 GB of RAM) machine.

ALNS clearly outperforms the results from the literature in terms of uncovered requests and cost. Note that the instances with more than 200 customers have been modified by [Azi et al., 2014] from the original data set provided by [Gehring and Homberger, 1999] to fit in a $100\times$ 100 square. As the modified data set is not available and normalization is not described in detail, several normalization procedures have been applied, and ALNSremains dominant. The results displayed here for ALNSare the worst case, *i.e.* , with the longest distances. Each point with coordinates (X, Y) is transformed into (X', Y') with formula (2.37), where X^{min} and X^{max} are the minimum and maximum values on the X-axis, respectively. A similar formula is applied to compute Y'.

$$X' = 100 * (X - X^{min})/(X^{max} - X^{min}) \tag{2.37}$$

2.9 Conclusion

This chapter studies a transportation service provided by ambulance companies. The daily planning problem, called CrewDARP, is modeled as an extension of the MT-VRPTW in which staff scheduling is imposed. This problem is of theoretical interest because of its high level of complexity. Although routing and staff scheduling are closely related in real-life applications, little research has been presented that considers the combination of those two problems. To solve CrewDARP, we propose an ALNS algorithm that allows us to solve instances with thousands of customers.

The most interesting result from our numerical experimentation is the robustness of ALNS. Our ALNS algorithm shows a high level of performance with the same setting of parameters on a wide range of problems and instances in terms of size and side constraints. This statement also mitigates the default of our algorithm, which has many parameters.

Our results also show that considering crew reassignment allows us to reach better solutions for the service provider, drivers and customers.

We hope that this practical and challenging problem will open some line of research and can become a test bed case for methodological contributions, especially about integrated models with multiple fixed and routing resources.

Next, improvements for local search approaches may address the theoretical complexity of the evaluation of neighbor solutions. Some partial evaluations may help to quickly identify promising solutions. At some time, a complete evaluation and repair procedure has to be applied to the current solution.

To satisfy the demand of planners, the dynamic version is worthy of interest and very challenging. In the medical transport system, 30% to 70% of the demand is requested during the day.

The proposed algorithm is run by planners some days before with a mix of known requested and fictitious requests. The resulting plan is used to assign working hours to paramedics and to build the initial routing plan at the beginning of the working day. The robustness of this approach must also be investigated and might be improved.

Algorithm 1 getBestInsertion(r)

Require: Request r
Ensure: The best insertion position $(u, pos)^*$ for r

1: **for all** trips u, and insertion position pos of r into u **do**
2: t := SingleTripEvaluation(u,r)
3: **if** $F(t)$ is False **then** GOTO 1
4: **end if**
5: **if** $SF(t) = |D(u)|$ **then**
6: **if** $Check(D(u), t) = \infty$ **then** GOTO 1
7: **end if**
8: **else if** $N := SF(t) - |D(u)| > 0$ **then**
9: **for all** $d \in \mathcal{D} \setminus D(u)$ **do**
10: $s_d := Check(d \cup D(u), t)$ without Lines 7-13
11: **end for**
12: $L := Sort(d \in \mathcal{D} \setminus D(u))$ in descending s_d
13: **for all** $(d_1, .., d_N) \in L^N$ **do**
14: $D := D(u) \cup (d_1, .., d_N)$
15: **if** $Check(D, t) \neq \infty$ **then** $C_{u,pos} := Cost(D, t)$; GOTO 1
16: **end if**
17: **end for**
18: **else if** $m := |D(u)| - SF(t) > 0$ **then**
19: **for all** $d \in D(u)$ **do**
20: $s_d := Check(D(u) \setminus \{d\})$ without Lines 7-13
21: **end for**
22: $L := Sort(D(u) \setminus \{d\})$ in descending s_d
23: **for all** $\{d_1, ..., d_m\} \in D(u)^m$ **do**
24: $D := D(u) \setminus \{d_1, ..., d_m\}$
25: **if** $Check(D, t) \neq \infty$ **then** $C_{u,pos} := Cost(D, t)$; GOTO 1
26: **end if**
27: **end for**
28: **if** $Check(D(u), t) \neq \infty$ **then** $C_{u,pos} := Cost(D(u), t)$; GOTO 1
29: **end if**
30: **end if**
31: $(u, pos)^* = \text{argmin}_{u,pos} C_{u,pos}$
32: **end for**

Algorithm 2 Check(D,t,u)

Require: Trip t ; Crew D ; previous trip u

if $\{v \in V(t) : D$ is qualified to use $v\} = \emptyset$ **then return** ∞
end if
if D does not cover all request qualifications for t **then return** ∞
end if
if $R := Timing(D,t)$ is unfeasible **then return** ∞
end if
for all $v \in V(t)$ ($v(u)$ is considered first if $v(u) \in V(t)$) **do**
    if $\exists (p,q) \in A^T : v(p) = v(q) = v, R(x_p) + Dur(p) \leq R(x_t), R(x_t) + Dur(t) \leq R(x_q)$ **then**
        Assign t to v between p and q
        return R
    end if
end for
return ∞

TABLE 2.9 – ALNS under $C1$ objective function

Instance	Ob. V	U.R	Distance	Earlier	Later	Longer	Loss
A-10-1	12316.1	1	4783.9	1099	429	793	211.2
A-10-2	4286.6	0	3068.6	911	111	196	0.0
A-10-3	9756.4	1	3782.5	881	0	0	92.9
A-10-4	3882.6	0	3047.6	533	302	0	0.0
A-10-5	8750.3	0	4681.3	2971	250	848	0.0
A-06-1	10823.7	1	4977.1	156	164	241	285.6
A-06-2	7969.8	0	5628.8	1091	839	411	0.0
A-06-3	12245.0	1	4481.0	1259	441	874	190.0
A-06-4	6197.7	0	5016.7	1155	1	25	0.0
A-25-1	16328.7	1	7338.1	2322	500	1081	87.6
A-25-2	6102.4	0	5401.9	685	0	16	0.0
A-25-3	11327.6	0	8574.7	2024	2	726	0.0
A-61-1	77012.9	3	42588.5	11192	2308	5743	181.3
A-61-2	97542.3	1	56338.9	20128	5329	10666	80.4
A-61-3	112216.0	3	62879.0	18303	5191	10730	113.0
A-61-4	67076.6	1	37350.0	14787	2262	7635	42.6
A-61-5	74371.4	2	41222.4	11954	2891	8156	148.0

Algorithm 3 ALNS

Require: Initial solution $\mathcal{S}_0$

$\mathcal{S}_{\text{best}}, \mathcal{S} \leftarrow \mathcal{S}_0$
$nn = dn = 0; r = 3$
for all $i \in \{1, 2, \ldots, i^{\max}\}$ **do**
    $P_r \leftarrow pmf(\mathcal{R}^r)$
    $P_4 \leftarrow pmf(\mathcal{I})$
    $\mathcal{S}_{new} \leftarrow P_r \circ P_4(\mathcal{S})$
    if $Z(\mathcal{S}) > Z(\mathcal{S}_{new})$ **then**
        $\mathcal{S} \leftarrow \mathcal{S}_{new}$
        $\pi_{P_j} \leftarrow \pi_{P_j} + \sigma_j \ \forall j \in \{r, 4\}$
        $nn = 0$
        if $Z(\mathcal{S}_{best}) > Z(\mathcal{S}_{new})$ **then**
            $\mathcal{S}_{best} \leftarrow \mathcal{S}_{new}$
            $\pi_{P_j} \leftarrow \pi_{P_j} + \sigma_j \ \forall j \in \{1, 2\}$
            $dn = 0$
        end if
    end if
    if $nn < NN$ **then**
        $nn = nn + 1\,;\ r = 3$
    else
        $nn = 0$
        if $dn < DN$ **then**
            $dn = dn + 1\,;\ r = 2$
        else
            $dn = 0\,;\ r = 1$
        end if
    end if
    if $i \pmod n = 0$ **then**
        for all $P \in \mathcal{R} \cup \mathcal{I}$ **do** $\Pi_P = \Pi_P + \gamma \pi_P$
        end for
        for all $P \in \mathcal{R}$ **do** $w_P = \Pi_P / \sum_{U \in \mathcal{R}} \Pi_U$
        end for
        for all $P \in \mathcal{I}$ **do** $w_P = \Pi_P / \sum_{U \in \mathcal{I}} \Pi_U$
        end for
    end if
end for

Algorithm 4 Related Removal

Input : $\{H_i, \forall i \in P\}$ the list of requests j sorted in nondecreasing order of $Rel(i,j) >= 0$; Solution $\mathcal{S}$; $Req_{remove} \in \mathbb{N}$; $d \in \mathbb{R}$
Select randomly a request $i \in P$
Remove i from $\mathcal{S}$
List $\leftarrow \{i\}$
while $|List| < Req_{remove}$ **do**
    $i \leftarrow$ request uniformly randomly selected from $List$
    $H \leftarrow \{r \in P : Rel(i,r) \geq 0\}$ sorted by nondecreasing $Rel(i,r)$ values
    $x = Unif(0;1)$
    p$\leftarrow \lceil |H| \,.\, x^d \rceil$;
    Select request j at position p in H_i
    Remove j from $\mathcal{S}$
    $List \leftarrow List \cup \{j\}$
end while
return $\mathcal{S}$

Algorithm 5 Constructive algorithm – first phase

Ensure: the trips $\mathcal{T}$ and the not-inserted requests $\mathcal{M}$
$\mathcal{T} \leftarrow \emptyset$ and $\mathcal{M} \leftarrow \emptyset$
$\mathcal{L} \leftarrow P$ ordered by ascending es_{i}
while $\mathcal{L} \neq \emptyset$ **do**
    $i \leftarrow$pop_front($\mathcal{L}$)
    $\{e_1, ..., e_{\mathrm{l}}\} \leftarrow$find_Crew_Hard($i$)
    if $\{e_1, ..., e_{\mathrm{l}}\} \neq \emptyset$ **then**
        $T \leftarrow$make_Trip($i, \{e_1, ..., e_{\mathrm{l}}\}$)
        $\mathcal{L}' \leftarrow \mathcal{L}$
        while $\mathcal{L}' \neq \emptyset$ **do**
            $j \leftarrow$pop_front($\mathcal{L}'$)
            if add_Hard(T, j) is feasible **then**
                $T \leftarrow$add_Hard(T, j)
            end if
        end while
        $\mathcal{T} \leftarrow \mathcal{T} \cup \{T\}$
    else
        $\mathcal{M} \leftarrow \mathcal{M} \cup \{i\}$
    end if
end while

Algorithm 6 Constructive algorithm – second phase

Require: $\mathcal{T}, \mathcal{M}$ as outputs of the first phase

while $\mathcal{M} \neq \emptyset$ **do**
  $i \leftarrow$pop_front($\mathcal{M}$)
  $\{e_1, ..., e_l\} \leftarrow$find_Crew_Soft($i$)
  $\Delta_Z \leftarrow$ the increase in objective function when i is assigned to $\{e_1, ..., e_l\}$
  $T^* \leftarrow \varnothing$
  for all $T \in \mathcal{T}$ **do**
    if add_Soft(T, i) is feasible **then**
      $\Delta'_Z \leftarrow$ the increase in the objective function when i is inserted into T
      if $\Delta_Z > \Delta'_Z$ **then**
        $\Delta_Z \leftarrow \Delta'_Z$
        $T^* \leftarrow T$
      end if
    end if
  end for
  if $T^* \neq \varnothing$ **then**
    $T^* \leftarrow$add_Soft(T^*, i)
  else if $\{e_1, ..., e_l\} \neq \emptyset$ **then**
    $T \leftarrow$make_Trip$(i, \{e_1, ..., e_l\})$
    $\mathcal{T} := \mathcal{T} \cup \{T\}$
  end if
end while

TABLE 2.10 – ALNS under $C2$ objective function

Instance	Ob. V	U.R	Distance	Earlier	Later	Longer	Loss
A-10-1	2993348.6	23	3301.9	1595	154	829	1491.1
A-10-2	1718732.4	7	2521.7	1198	909	1729	278.1
A-10-3	2294352.2	13	3143.7	1763	184	1086	566.8
A-10-4	1683915.2	8	2383.4	1127	1677	1931	348.6
A-10-5	2771878.7	15	3613.6	2568	569	1471	873.6
A-06-1	2994332.2	29	2802.9	2042	251	1241	1940.1
A-06-2	2412595.0	16	2420.1	2175	674	1458	1460.4
A-06-3	2349140.8	18	2706.2	3204	1073	1688	1049.1
A-06-4	2797218.8	21	2855.8	1384	539	1232	1626.0
A-25-1	1731921.9	14	1644.1	2042	237	991	1120.3
A-25-2	2041337.2	10	2462.3	2646	1333.5	2851	845.2
A-25-3	3897995.1	12	5729.8	4134	1886	3862	650.3
A-61-1	18761520.3	121	21584.6	13564	5078	12564	8624.3
A-61-2	23068059.3	151	26254.5	22127	10110	22661	10842.4
A-61-3	28328919.3	177	33121.9	23517	8968	22185	12526.8
A-61-4	15659630.9	117	17237.5	10018	4859	9968	7845.4
A-61-5	19280600.1	138	21182.4	12256	3813	11461	9756.1

TABLE 2.11 – ALNS under $C3$ objective function

Instance	Ob. V	U.R	Distance	Earlier	Later	Longer	Loss
A-10-1	96784.0	18	3727.4	1	0	0	2456.6
A-10-2	124010.9	23	2602.3	8	0	0	1608.6
A-10-3	81067.2	15	3520.8	2	0	0	1346.4
A-10-4	43938.6	8	3064.3	0	0	0	874.3
A-10-5	123899.1	23	3613.9	5	0	0	2285.2
A-06-1	30424.4	5	4618.6	0	0	0	805.8
A-06-2	146485.8	28	3225.6	2	0	0	2060.3
A-06-3	115391.8	21	3834.2	9	0	0	1157.6
A-06-4	80581.4	15	4263.6	0	0	0	1317.8
A-25-1	63775.9	12	2715.1	0	0	0	1060.9
A-25-2	75189.0	13	4012.5	9	0	0	1076.5
A-25-3	152283.8	28	6803.9	5	0	0	2479.8
A-61-1	791735.4	143	39128.3	42	1	1	11207.0
A-61-2	1425706.2	267	44365.2	42	0	0	21141.0
A-61-3	1448190.8	272	59582.5	12	1	0	20808.3
A-61-4	869639.5	163	27268.9	22	4	0	11770.5
A-61-5	862496.0	157	36970.7	42	3	3	11725.3

TABLE 2.12 – ALNSperformance on A-instances

Instance	ALNSBest			ALNSavg			
	Ob.V	U.R	Cost	Ob.V	U.R	Cost	T(s)
A-10-1	11729.4	1.0	6729.4	13562.8	1.3	6896.1	620.9
A-10-2	4286.6	0.0	4286.6	4415.7	0.0	4415.7	549.7
A-10-3	9756.4	1.0	4756.4	9917.4	1.0	4917.4	653.8
A-10-4	3882.6	0.0	3882.6	3954.7	0.0	3954.7	555.5
A-10-5	7477.2	0.0	7477.2	8183.1	0.0	8183.1	651.0
A-06-1	5208.6	0.0	5208.6	8850.7	0.7	5517.3	830.2
A-06-2	7969.8	0.0	7969.8	8058.3	0.0	8058.3	673.5
A-06-3	11821.2	1.0	6821.2	12032.1	1.0	7032.1	831.0
A-06-4	6197.7	0.0	6197.7	9738.9	0.7	6405.6	668.3
A-25-1	16253.6	1.0	11253.6	16319.5	1.0	11319.5	750.5
A-25-2	5763.6	0.0	5763.6	5947.5	0.0	5947.5	934.7
A-25-3	11312.3	0.0	11312.3	12886.5	0.3	11173.6	1177.0
A-61-1	72466.6	2.0	62466.6	75444.5	2.3	63777.8	2943.2
A-61-2	97542.3	1.0	92542.3	103783.6	2.3	92117.0	4892.7
A-61-3	112216.0	3.0	97216.0	117713.5	3.3	101046.8	6399.2
A-61-4	67020.9	1.0	62020.9	67471.5	1.3	60804.8	2488.6
A-61-5	62867.3	1.0	57867.3	66815.7	1.3	60149.0	3852.4

TABLE 2.13 – HLA+ and HLA performances on A-instances

Instance	HLA+				HLA			
	Ob.V	U.R	Cost	T(s)	Ob.V	U.R	Cost	T(s)
A-10-1	39602.3	6.0	9602.3	4.4	74414.7	13.0	9414.7	0.1
A-10-2	11307.3	1.0	6307.3	1.6	32139.2	5.0	7139.2	0.1
A-10-3	18309.0	2.0	8309.0	3.1	59459.4	10.0	9459.4	0.2
A-10-4	10496.2	1.0	5496.2	1.3	30765.1	5.0	5765.1	0.1
A-10-5	39162.4	6.0	9162.4	3.0	69285.3	12.0	9285.3	0.1
A-06-1	30202.6	4.0	10202.6	16.9	70820.7	12.0	10820.7	0.1
A-06-2	22770.2	3.0	7770.2	5.7	29200.3	4.0	9200.3	0.1
A-06-3	49467.0	8.0	9467.0	21.0	111758.9	20.0	11758.9	0.1
A-06-4	57673.0	10.0	7673.0	13.9	73995.7	13.0	8995.7	0.1
A-25-1	21010.0	2.0	11010.0	9.7	21977.2	2.0	11977.2	0.1
A-25-2	7595.1	0.0	7595.1	8.1	8365.6	0.0	8365.6	0.1
A-25-3	20425.8	1.0	15425.8	52.3	45705.9	6.0	15705.9	0.2
A-61-1	152596.9	12.0	92596.9	2535.6	201746.0	21.0	96746.0	1.2
A-61-2	203168.1	14.0	133168.1	4919.4	248580.9	23.0	133580.9	2.3
A-61-3	274819.6	26.0	404819.6	10011.0	395711.3	48.0	155711.3	3.6
A-61-4	108297.6	4.0	88297.6	1537.7	160841.1	13.0	95841.1	3.7
A-61-5	159576.0	12.0	99576.0	2719.7	225313.4	25.0	100313.4	2.3

TABLE 2.14 – ALNS vs 1-Trip-ALNS

	ALNS					1-Trip-ALNS		
Instance	Ob.V	U.R	Cost	Avg_{tr}	max_{tr}	Ob.V	U.R	Cost
A-10-1	13562.8	1.3	6896.1	1.3	4.0	17330.8	2.0	7330.8
A-10-2	4415.7	0.0	4415.7	1.1	3.0	4875.9	0.0	4875.9
A-10-3	9917.4	1.0	4917.4	1.3	3.0	10244.1	1.0	5244.1
A-10-4	3954.7	0.0	3954.7	1.2	3.0	4960.1	0.0	4960.1
A-10-5	8183.1	0.0	8183.1	1.3	3.0	7803	0.0	7803.0
A-06-1	8850.7	0.7	5517.3	1.3	3.0	11447.2	1.0	6447.2
A-06-2	8058.3	0.0	8058.3	1.4	3.0	8466.7	0.0	8466.7
A-06-3	12032.1	1.0	7032.1	1.4	3.0	14323.4	1.0	9323.4
A-06-4	9738.9	0.7	6405.6	1.3	3.0	11067.6	1.0	6067.6
A-25-1	16319.5	1.0	11319.5	1.3	3.0	16345.6	1.0	11345.6
A-25-2	5947.5	0.0	5947.5	1.1	3.0	6273.3	0.0	6273.3
A-25-3	12886.5	0.3	11173.6	1.4	4.0	11982.1	0.0	11982.1
A-61-1	75444.5	2.3	63777.8	1.3	5.0	90578.1	5.0	65578.1
A-61-2	103783.6	2.3	92117.0	1.2	4.0	115585.8	4.0	95585.8
A-61-3	117713.5	3.3	101046.8	1.2	5.0	143926.3	7.0	108926.3
A-61-4	67471.5	1.3	60804.8	1.2	4.0	75874.6	3.0	60874.6
A-61-5	66815.7	1.3	60149.0	1.3	5.0	74108.8	3.0	59108.8

TABLE 2.15 – Break at the depot (ALNS-BAD) versus break on the way (ALNS-BOW)

	ALNS-BAD			ALNS-BOW		
Instance	Ob.V	U.R	Cost	Ob.V	U.R	Cost
A-10-1	12316.1	1.0	7104.9	11168.9	1.0	6168.9
A-10-2	4286.6	0.0	4286.6	4241.5	0.0	4241.5
A-10-3	9756.4	1.0	4663.5	9622.4	1.0	4622.4
A-10-4	3882.7	0.0	3882.7	3837.5	0.0	3837.5
A-10-5	8750.4	0.0	8750.4	7556.3	0.0	7556.3
A-06-1	10823.7	1.0	5538.1	10172.5	1.0	5172.5
A-06-2	7969.8	0.0	7969.8	7892.6	0.0	7892.6
A-06-3	12245.0	1.0	7055.0	6829.3	0.0	6829.3
A-06-4	6197.7	0.0	6197.7	10689.3	1.0	5689.3
A-25-1	16328.7	1.0	11241.1	15906.4	1.0	10906.4
A-25-2	6102.5	0.0	6102.5	5615.6	0.0	5615.6
A-25-3	11327.6	0.0	11327.6	11225.1	0.0	11225.1
A-61-1	77012.9	3.0	61831.5	65471.5	1.0	60471.5
A-61-2	97542.3	1.0	92461.9	91414.3	1.0	86414.3
A-61-3	112216.0	3.0	97103.0	104917.1	1.0	99917.1
A-61-4	67076.6	1.0	62034.0	61916.2	0.0	61916.2
A-61-5	74371.4	2.0	64223.4	54871.6	0.0	54871.6

TABLE 2.16 – ALNS with customers preferences

	ALNS			ALNS-$\Phi_i\Psi_i$		
Instance	Ob.V	U.R	Cost	Ob.V	U.R	Cost
A-10-1	12316.1	1.0	7104.9	27279.0	4.0	7279.0
A-10-2	4286.6	0.0	4286.6	9800.6	1.0	4800.6
A-10-3	9756.4	1.0	4663.5	10641.4	1.0	5641.4
A-10-4	3882.7	0.0	3882.7	4186.5	0.0	4186.5
A-10-5	8750.4	0.0	8750.4	8968.1	0.0	8968.1
A-06-1	10823.7	1.0	5538.1	17423.5	2.0	7423.5
A-06-2	7969.8	0.0	7969.8	8866.6	0.0	8866.6
A-06-3	12245.0	1.0	7055.0	22900.5	3.0	7900.5
A-06-4	6197.7	0.0	6197.7	16242.4	2.0	6242.4
A-25-1	16328.7	1.0	11241.1	16977.6	1.0	11977.6
A-25-2	6102.5	0.0	6102.5	5985.1	0.0	5985.1
A-25-3	11327.6	0.0	11327.6	14421.9	0.0	14421.9
A-61-1	77012.9	3.0	61831.5	94293.9	5.0	69293.9
A-61-2	97542.3	1.0	92461.9	143406.1	8.0	103406.1
A-61-3	112216.0	3.0	97103.0	147789.2	6.0	117789.2
A-61-4	67076.6	1.0	62034.0	94668.1	4.0	74668.1
A-61-5	74371.4	2.0	64223.4	90652.2	5.0	65652.2

TABLE 2.17 – ALNSversus SOA [Lim et al., 2017] on RDARP

		Average /10 runs		
		Holidays	Non-Holydays	All instances
hline Number of all instances		116	249	365
Number of all requests		6244	41993	48237
objective value	A	85186.20	159976.39	136068.05
	H	79190.73	177508.21	146262.10
Sum of uncovered Requests	A	53.40	45.10	98.50
	H	49.20	89.20	138.40
Distance	A	384.68	1393.13	1070.75
	H	363.73	1402.07	1072.08
Workload Deviation	A	1077.39	2551.09	2079.99
	H	403.48	1477.48	1136.16
Time CPU	A	1278.12	5527.17	4176.79
	H	1279.43	5541.26	4186.81
% best solutions	A	0.44	0.59	0.55
	H	0.56	0.41	0.45

TABLE 2.18 – Optimal and [Lim et al., 2017] results on MTVRPTW small (10 runs) instances

Instances				Optimum		[Lim et al., 2017]				
						Best		Average		
size	T_{GT}	Class	# Instances	U.R	Distance	U.R	Dist.	U.R	Distance	T_{bst}
25	220	C	8	0.00	631.43	0.00	631.43	0.00	631.46	43.00
25	75	R	11	0.00	620.13	0.00	620.13	0.00	620.13	13.00
25	75	RC	8	-	-	0.00	804.07	0.00	804.07	51.00
25	220	C	8	0.00	530.27	0.00	530.27	0.00	530.27	2.00
25	75	R	11	0.00	544.99	0.00	544.99	0.00	544.99	28.00
25	75	RC	8	-	-	0.00	629.00	0.00	629.00	8.00
40	220	C	8	0.00	1087.75	0.00	1087.75	0.00	1089.01	595.00
40	75	R	11	-	-	0.18	956.64	0.18	962.80	812.00
40	75	RC	8	-	-	4.00	1306.00	4.00	1315.00	864.00
40	220	C	8	-	-	0.00	922.43	0.00	922.94	680.00
40	75	R	11	-	-	0.09	822.66	0.09	824.03	477.00
40	75	RC	8	-	-	1.37	1125.74	1.37	1149.68	862.00

TABLE 2.19 – ALNS results on MTVRPTW small (10 runs) instances

Instances				ALNS				
				Best		Average		
size	T_{GT}	Class	# Instances	U.R	Dist.	U.R	Dist.	T_{bst}
25	220	C	8	0.00	631.43	0.00	631.48	44.00
25	75	R	11	0.00	620.13	0.00	620.13	11.00
25	75	RC	8	0.00	804.07	0.00	804.07	56.00
25	220	C	8	0.00	530.27	0.00	530.27	3.00
25	75	R	11	0.00	544.99	0.00	544.99	22.00
25	75	RC	8	0.00	629.00	0.00	629.30	7.00
40	220	C	8	0.00	1087.75	0.00	1088.32	584.00
40	75	R	11	0.18	955.75	0.18	958.15	925.00
40	75	RC	8	3.00	1341.00	3.49	1329.52	787.00
40	220	C	8	0.00	922.17	0.00	922.33	694.00
40	75	R	11	0.09	821.78	0.09	823.67	478.00
40	75	RC	8	1.25	1131.36	1.34	1121.96	851.00

TABLE 2.20 – ALNS vs Azi et al. on MTVRPTW instances

				Azi et al.			ALNS		
Size	T_{GT}	Class	#instances	U.R.	Cost	$T(s)$	U.R.	Cost	$T(s)$
100	100	RC	8	25.7	1899.2	28.1	18.0	1733.5	30.1
	100	R	11	10.9	1828.6	33.5	7.2	1603.6	35.4
	100	C	8	0.0	2232.9	43.2	0.0	1777.6	55.5
200	100	RC	10	12.4	9218.3	140.3	1.9	5028.6	157.0
	100	R	10	6.1	11103.7	126.9	0.2	5915.8	91.2
	100	C	10	0.0	9730.3	126.2	0.0	4832.0	126.1
400	100	RC	10	22.4	10128.0	460.4	2.0	9817.0	470.5
	100	R	10	6.2	12657.7	427.7	0.0	11934.2	404.3
	100	C	10	0.0	10937.2	340.0	0.0	8227.4	493.1
600	100	RC	10	20.4	15577.9	1114.2	12.3	14788.6	1374.3
	100	R	10	6.4	19089.2	1009.1	6.5	20692.9	1219.6
	100	C	10	9.9	14626.0	1028.5	0.0	15477.0	1137.5
800	100	RC	10	24.3	20858.5	1842.8	6.7	25136.6	1938.8
	100	R	10	7.9	26136.9	1678.3	3.0	31597.8	2111.9
	100	C	10	24.9	14441.1	1747.4	0.0	14482.9	1953.5
1000	100	RC	10	26.3	25368.3	2855.2	4.7	26713.2	3409.3
	100	R	10	10.3	30732.2	2718.8	8.3	31668.3	3645.8
	100	C	10	36.5	14587.6	2602.4	5.5	20161.4	2753.9

Chapitre 3

Pilotage réactif d'une flotte d'ambulances

Résumé du chapitre

Ce chapitre est axé sur le traitement du problème de "Planification des réseaux d'ambulance" dans sa version dynamique, noté DMTP pour *Dynamic Medical Transportation Problem*. Au début de la journée un pourcentage de demandes est connu à l'avance environ de 30 à 70 % selon les ambulanciers, le reste des demandes surgit de manière dynamique durant la journée. En plus de ces nouvelles demandes, les régulateurs doivent faire face à des imprévus dans le déroulement des opérations. Il s'agit essentiellement du retard pris en transport ou en attente de disponibilité de passagers.

Les nombreuses contraintes à intégrer et la taille des flottes d'ambulances à gérer rendent ces décisions particulièrement complexes à prendre. Le régulateur doit tenir compte du plan prévu, des heures d'embauche et débauche des chauffeurs, du respect des pauses, du respect des horaires de prise charge, de la disponibilité des véhicules, ...

Ce chapitre vise la mise en place d'outils pour automatiser ou supporter les décisions de planification ou replanification des régulateurs en cours de journée. Une approche optimisée est comparée avec deux stratégies similaires à une planification manuelle à travers un large plan d'expérience exécuté sur une plateforme de simulation développée pour une journée d'activité. Les différentes contraintes telles que la gestion des pauses, la gestion des horaires de travail et la possibilité de réorganiser les équipages sont intégrées à un modèle du type ramassage et livraison dynamique de façon indépendante. Ainsi, la comparaison entre les différentes stratégies est établie pour chacune de ces catégories de contraintes. La stratégie optimisée dominant les deux autres est évaluée en prenant en compte l'ensemble des caractéristiques du problème de gestion des ambulances.

Un plan d'expérience supplémentaire fait le lien pratique avec le chapitre précédent. L'impact de différentes utilisations d'un plan de route calculé avec les demandes connues la veille pour le lendemain est évalué.

Sommaire

3.1 Introduction 62
3.2 État de l'art 67
3.2.1 Types d'approches 67
3.2.2 Synthèse 70
3.3 Description générale du problème 71
3.3.1 Régulation du transport sanitaire 71
3.3.2 Différentes approches de régulations 74
3.4 Définition du Problème de Transport Médicalisé Dynamique (DMTP) 76
3.4.1 Réseau de transport 76
3.4.2 Chauffeurs et dépôts 77
3.4.3 Véhicule 78
3.4.4 Demande 79
3.4.5 Objectifs et contraintes 79
3.4.6 Hypothèses et simplifications par rapport au problème réel . . 81
3.5 Problèmes intermédiaires considérés 82
3.5.1 Problème simplifié P_1 – **DARP** 82
3.5.2 Problème P_2 – **DARP avec temps de travail** 82
3.5.3 Problème P_3 – **DARP avec pauses** 82
3.5.4 Problème P_4 – **DARP avec plusieurs types de véhicules** . 83
3.6 Stratégies de planification 83
3.6.1 Fonctionnement des stratégies et pré-traitements communs . . 84
3.6.2 Stratégie basée sur les activités S_1 88
3.6.3 Stratégie basée sur les ressources puis les activités S_2 91
3.6.4 Stratégie basée sur les activités puis les ressources S_3 92
3.7 Expérimentation numérique 93
3.7.1 Plateforme de tests 93
3.7.2 Génération de données 96
3.7.3 Résultats numériques 98
3.8 Conclusion 104

3.1 Introduction

Le bureau central de régulation est au cœur du processus de la gestion du transport sanitaire. C'est là où les nouvelles demandes sont reçues tout au long de la journée pendant que les ambulanciers sont en route pour la prise en charge des usagers. Les régulateurs attribuent les demandes entrantes aux ambulanciers actifs et définissent leur plan de route en indiquant l'heure à laquelle chaque lieu de ramassage ou de dépose doit être atteint.

Les décisions de routage et d'ordonnancement ont un caractère dynamique car elles sont prises et exécutées au fil du temps. De plus, le plan prévu est régulièrement remis en question à cause d'aléas qui perturbent un déroulement acceptable.

Les régulateurs sont tenus de répondre à des objectifs contradictoires : anticiper les plans de route de la journée et prendre des décisions de planification ou replanification en temps réel. L'anticipation permet d'accepter ou refuser rapidement les nouvelles demandes, et donne la possibilité aux chauffeurs de travailler sereinement puisqu'ils sont avisés à l'avance de leurs courses et surtout du temps de leur pause et l'heure de débauche. Les décisions en temps réel permettent d'intégrer des demandes de dernière minute et de gérer des retards à cause du trafic ou de l'indisponibilité des usagers.

Chaque régulateur a sa propre capacité pour assurer un équilibre entre l'anticipation et la réactivité. En général, les régulateurs sont discrets quant aux informations transmises au chauffeur et horaires de passages communiqués à l'usager. C'est en fonction du moment de communication de ces détails que le régulateur gère son niveau de réactivité.

Avec une telle dynamique, il devient alors extrêmement difficile de réaliser un planning efficace manuellement, ce qui pousse les entreprises à chercher de plus en plus des profils de régulateurs expérimentés dans le but d'optimiser la gestion des ressources. Ces régulateurs ont besoin d'un temps d'adaptation avant de faire preuve d'efficacité sur tout le territoire. Dès lors, il devient difficile de les remplacer en cas d'absence ou de départ.

Les dernières évolutions des logiciels de régulation fournissent des outils d'aide à la décision basés sur de la visualisation avancée (cartographie dynamique sur système d'information géographique), des filtres (sélection de véhicules disponibles pour une course, ou de courses proches d'un véhicule), etc. Cependant ces outils montrent rapidement leurs limites lorsque les régulateurs sont confrontés à la nécessité de prendre plusieurs décisions en même temps. Ils n'ont pas les moyens adéquats leur permettant d'évaluer une affectation de plusieurs courses à plusieurs véhicules. Cette tendance croît avec le regroupement des régulations qui atteignent à présent plusieurs milliers de demandes par jour à traiter en connectant des territoires de plusieurs dizaines de milliers de kilomètres carrés. De plus, des décisions complexes comme les recompositions d'équipages restent difficilement évaluables avec les outils de visualisation uniquement.

Un système de planification automatique qui réagit en temps réel par rapport aux imprévus devient crucial et aidera les régulateurs dans leurs prises de décision. Les objectifs à court terme sont d'améliorer la performance des régulateurs, de raccourcir le temps d'apprentissage d'un régulateur sur un territoire, d'optimiser le temps au régulateur pour s'occuper d'autres tâches comme par exemple, accorder plus de temps pour répondre aux appels ou communiquer avec les chauffeurs ou le SAMU. À plus long terme, c'est l'activité du régulateur qui subira des changements : plus de contrôle et de résolution de problèmes sur une régulation plus automatisée.

Durant la journée, les événements suivants peuvent être source de perturbation pour la planification en cours :

— Les arrivées de nouvelles demandes de transport pour le jour même.

— Les perturbations au niveau du temps de trajet suite à des problématiques liée au trafic routier par exemple.

— L'indisponibilité de passagers dont le rendez-vous médical a été retardé par exemple.

— Les annulations de demande.

— L'indisponibilité d'un chauffeur en retard ou la survenance d'un incident technique.

Le problème de planification académique le plus proche est connu sous le nom de "Dial-A-Ride Problem" [Cordeau and Laporte, 2003] pour transport à la demande. Dans sa version dynamique, le "Dial-a-Ride Problem" est une variante du problème plus général de transport "Dynamic Pickup and Delivery vehicle vouting Problem with Time Windows" (DPDTW) [Berbeglia et al., 2010] avec l'intégration de contraintes spécifiques au transport de personnes et de qualité de service. Dans ce contexte, il n'y a pas de définition unanime du "Dial-A-Ride Problem" avec de nombreuses variantes selon les contextes.

La seule variante courante qui rend ce problème plus difficile par rapport à d'autres problèmes de ramassage et livraison à traiter est celle qui fixe une durée maximale sur le trajet en voiture ("maximal ride time" dans la littérature) en plus des contraintes de fenêtres de temps pour le ramassage ou la livraison.

Dans le contexte étudié on considère plutôt un temps maximum de disponibilité. Les requêtes ont soit une date souhaitée d'arrivée (pour un rendez-vous), soit une date de disponibilité (fin d'un rendez-vous ou d'une hospitalisation). Dans le premier cas, on définit une date de disponibilité au plus tôt à partir de laquelle le passager peut être récupéré, à son domicile généralement. Dans le deuxième cas, c'est une date de dépose au plus tard, au domicile en général, qui est exprimée. Combinées aux plus courts trajets entre origines et destinations, des contraintes de fenêtres de temps classiques sur chaque sommet modélisent parfaitement ces contraintes horaires.

Le temps passé dans le véhicule n'est donc pas directement contraint. Dans le contexte du DMTP avec des fenêtres horaires assez serrées (une vingtaine de minutes environ), le temps passé dans le véhicule est donc également assez limité.

Cependant comme nous le présentons par la suite, plusieurs contraintes bien plus complexes sont intégrées au transport médical et nous pensons qu'étendre notre approche à la contrainte de temps de trajet maximal ne devrait pas pénaliser les performances algorithmiques de notre approche.

à l'instar du problème statique du chapitre précédent, le problème étudié ici étend la définition des problèmes académiques de l'état de l'art avec notamment les caractéristiques suivantes :

1. Les véhicules nécessitent des équipages d'ambulanciers différents en nombre et compétences.
2. Les équipages peuvent être recomposés et changer de véhicules en cours de journée.

3. Les ambulanciers ont des compétences différentes.
4. Les ambulanciers ont leurs propres heures de disponibilité.
5. Les heures d'embauche et débauche des ambulanciers peuvent être modifiées en cours de journée. Il y a intérêt à laisser au repos un ambulancier sans activité car cela préserve son quota contractuel d'heures mensuelles. D'un autre côté ajouter des heures supplémentaires a un coût direct. Dans les deux cas modifier les heures de travail le jour même n'est pas toujours apprécié par les ambulanciers.
6. Les ambulanciers doivent prendre des pauses sous certaines conditions.
7. L'usager peut être transporté par un ou plusieurs types de véhicules différents. Ces types de transport, sont précisés par le médecin prescripteur. Par exemple un transport peut assis peut être effectué par un véhicule assurant le transport couché mais cela reste à éviter, pour des questions de sous facturation.

Nous appelons ce problème DMTP pour Problème de Transport Médicalisé Dynamique ou en anglais "Dynamic Medical Transportation Problem".

Avoir différents types de véhicules dont certains nécessitant un équipage de plusieurs ambulanciers, permet à ces derniers de changer de coéquipier et de véhicule plusieurs fois durant la journée. Cet aspect rend généralement le problème compliqué par la combinaison de changement d'équipage. Le tout oblige à bien modéliser et synchroniser finement les activités des différents chauffeurs et des véhicules.

D'après nos recherches aucun travail académique ou outil commercial ne traite l'ensemble de ces spécificités dans le cadre dynamique. Seuls quelques éléments comme la réorganisation d'équipages sont intégrés dans les travaux [Lim et al., 2017] sur des problèmes statiques.

Dans le chapitre précédent sont exposées les difficultés soulevées par les spécificités du problème pour résoudre une version statique du problème. Comme il est détaillé dans la section suivante sur l'état de l'art et dans la description des stratégies, la prise de décisions dynamiques devient plus complexes à cause de la nécessité de gérer les décisions d'embauche, débauche et de pause avec la possibilité d'exploiter la recomposition des équipages.

Afin de résoudre le DMTP, nous avons développé une stratégie d'optimisation que nous comparons avec deux autres stratégies qui déroulent la planification à la manière de deux régulateurs qui procèdent manuellement et suivant des logiques différentes. La première est guidée par les ressources et est axée sur la prochaine requête à traiter pour un équipage qui se libère. En général, cela génère des régulations avec peu d'anticipation et plus de flexibilité. La deuxième est guidée par les activités puis les ressources. La requête la plus pressée est affectée à un équipage, puis le plan de route de cet équipage est établi pour quelques requêtes supplémentaires. Dans ce cas, le régulateur tâche de construire des plans bien enchaînés pour certains équipages sur un horizon assez long afin de limiter le champ des possibilités pour les futures demandes à traiter.

Pour comparer les stratégies, nous avons tout d'abord étudié quatre versions simplifiées du DMTP. Puisqu'il y a plusieurs contraintes assez complexes à prendre en compte, les intégrer chacune séparément permet de mieux mesurer l'impact de chacune sur les différentes stratégies. L'interprétation des différents choix effectués par les différentes stratégies est ainsi plus facile car elle n'est pas perturbée par la prise en compte de la combinaison de plusieurs contraintes complexes.

Par exemple une stratégie qui gère mal la décision de débauche des ambulanciers serait sûrement pénalisée sur plusieurs critères de performance autre que ceux directement liés à la gestion des heures de travail et rendrait difficile l'analyse explicative des performances de cette stratégie sur le DMTP complet.

Les quatre cas particuliers et le problème DMTP complet se déclinent ainsi :

1. Le **DARP** : représente le problème de régulation dynamique du transport sanitaire sans considérer les contraintes spécifiques au DMTP. En effet il s'agit d'un problème de ramassage et de livraison avec fenêtres de temps. Aucune des caractéristiques spécifiques citées avant n'est introduite ; les chauffeurs et les véhicules sont tous équivalents avec des disponibilités fixes et il n'y pas de pauses.
2. **DARP avec temps de travail** : Décision sur heure de débauche. Sont intégrées les spécificités 4 et 5.
3. **DARP avec pauses** : Les pauses sont gérées pour chaque ambulancier 6.
4. **DARP avec plusieurs types de véhicules** : Les différents types de véhicules sont gérés avec des recompositions d'équipages possibles. Cela correspond à l'intégration des spécificités 1, 2, 3, 7.
5. Le **DMTP** comprend lui toutes les spécificités de 1 à 7.

Chaque stratégie se décline en plusieurs variantes définies par un cadre général. Il s'agit de déterminer la liste des événements pouvant déclencher une replanification ; quelles données sont intégrées à la replanification ; quelles parties *gelées* du plan en cours ne sont pas remises en cause ; quelles parties *consolidées* peuvent être modifiées mais avec une pénalité ; quelles parties *libres* peuvent être re-planifiées sans contraintes additionnelles.

La stratégie générique utilisée est définie à la section 3.6.1. Elle s'appuie essentiellement sur des paramètres temporels comme une fréquence de replanification et des intervalles de temps pour classer les requêtes figées, consolidées et libres.

Un environnement de simulation à événements discrets a été implémenté afin de mener un plan d'expérience permettant de répondre aux trois principales questions posées :

1. De combien peut-on améliorer les performances des stratégies manuelles actuelles avec des outils automatiques ?
2. Dans quelle mesure le plan de la veille pour le lendemain peut supporter une meilleure régulation le lendemain ?

3. Quels sont les meilleures stratégies à adopter suivant la compagnie d'ambulance, sa clientèle et son territoire ?

La simulation permet de dérouler les arrivées et les imprévus d'une journée, d'appliquer les stratégies de planification au moment défini et de récupérer les indicateurs de performance en fin de journée.

Ce chapitre est organisé comme suite : Nous allons définir dans la section suivante 3.2 un état de l'art sur les problèmes dynamiques des flottes de véhicules, avec la mise en avant des spécificités du DMTP. La section 3.3, nous allons décrire le problème de régulation dynamique du transport sanitaire. Une définition du problème d'optimisation sous-jacente sera ensuite fournie dans la section suivante 3.4. Au niveau de la section 3.5 allons mettre l'accent sur l'ensemble des problèmes intermédiaires considérés. Ensuite l'ensemble des stratégies de résolution et leur adaptation par rapport à chaque problème sera proposée dans la section 3.6. Les deux dernières sections 3.7.1 et 3.7 seront consacrées à la présentation de la plateforme de simulation à évènements discrets et les expérimentations numériques.

3.2 État de l'art

Les premières études concernant le DARP dynamique remontent aux années 80 ([Psaraftis, 1980]). Depuis, plusieurs études continuent d'être publiées fréquemment.

Dans cette section, nous présenterons tout d'abord les différents types d'approches soulevées dans la littérature afin de tenir en compte les incertitudes. Nous décrierons succinctement, par la suite, quelques méthodes d'optimisation (étant donné la taille importante de la littérature scientifique sur le sujet, nous ne pouvons pas étudier l'exhaustivité des méthodes). Ensuite, nous listerons les contraintes prises en compte dans les problèmes de littérature. Enfin, nous réaliserons une synthèse et nous positionnerons ce travail par rapport au reste de la littérature en recherche opérationnelle.

3.2.1 Types d'approches

Trois approches générales sont utilisées pour tenir compte de l'incertitude associée au DARP dynamique : l'optimisation basée sur un horizon glissant, la programmation stochastique et l'optimisation robuste ([Karami et al., 2018]).

L'optimisation basée sur un horizon glissant tient compte de l'incertitude en prenant des décisions répétées (ré-optimisation) au fil du temps. Dans chaque appel à la ré-optimisation, la prise de décision s'obtient en utilisant des méthodes d'optimisation statique.

Quant à l'optimisation stochastique, elle repose soit sur des lois de distribution de variables aléatoires qui définissent les incertitudes, soit sur des connaissances relatives à des probabilités d'événements futurs ou à des scénarios probables. Par exemple, les

arrivées de nouvelles requêtes peuvent suivre une distribution de Poisson spécifique ([Swihart and Papastavrou, 1999]).
Enfin, les méthodes d'optimisation robustes cherchent à construire une solution capable de satisfaire toute réalisation des paramètres incertains.

3.2.1.1 Méthodes d'optimisation

Afin de prendre en compte les informations dynamiques dans les problèmes du DARP, la majorité des techniques d'optimisation se concentrent sur l'élaboration des méthodes statiques. Ces dernières se relancent répétitivement dans le but de ré-optimiser et de produire des solutions de bonne qualité ([Psaraftis et al., 2016]).
Les approches les plus répondues dans la technique de ré-optimisation sont (voir Figure 3.1) :

— **Les approches réactives** faisant appel à des stratégies d'optimisation à chaque fois qu'une nouvelle demande est arrivée ([Chang et al., 2003] ; [Schyns, 2015]) ;

— **Les approches périodiques** exécutant des stratégies d'optimisation après chaque intervalle de temps prédéfini ([Kilby et al., 1998]) ;

— **Les approches continues** effectuant l'optimisation pour améliorer la solution tout au long de la journée. Lorsque des nouvelles informations arrivent, une procédure de décision agrège ces information pour re-mettre à jour la solution actuelle ([Pillac et al., 2013]) ;

— **Les approches basées sur des événements** exécutants des stratégies de ré-optimisation à chaque fois qu'un événement déclencheur est apparu. Un événement peut correspondre à l'arrivée d'une nouvelle demande, d'un retard, d'un arrivée de véhicule, ... [Coslovich et al., 2006].

Les approches réactives sont rarement utilisées dans les problèmes du monde réel. Dans la plupart des applications pratiques du DARP dynamique, les périodes inter-arrivées des nouvelles demandes sont étroites. De plus, les stratégies des approches réactives demandent une ré-optimisation complète de la solution à chaque fois qu'une nouvelle information arrive au système. Ainsi, cela engendre des temps de calcul démesurés.
Concernant les approches périodiques, elles sont les plus répondues dans le DARP dynamique car le temps disponible pour la ré-optimisation est préalablement connu. Toutefois, ces approches démontrent des limites notamment lorsque des requêtes urgentes arrivent au système et qu'il faut les servir avant la prochaine ré-optimisation.
L'avantage des approches continues repose dans le temps de calcul permanent, permettant de bénéficier de temps d'attente du système pour améliorer les solutions. Toutefois, les plans des routes sont susceptibles d'être changer à tout moment. Par conséquent, les chauffeurs ne connaissent pas à l'avance leur prochaine destination, ce qui n'est pas adapté aux situations réelles.

Quant aux approches basées sur des événements, leur grand avantage est de pouvoir établir des stratégies de ré-optimisation adaptées à chaque événement (par exemple des stratégies pour l'insertion à chaque nouvelle demande, pour la réinsertion de chaque demande en retard et pour la ré-optimisation complète par période).

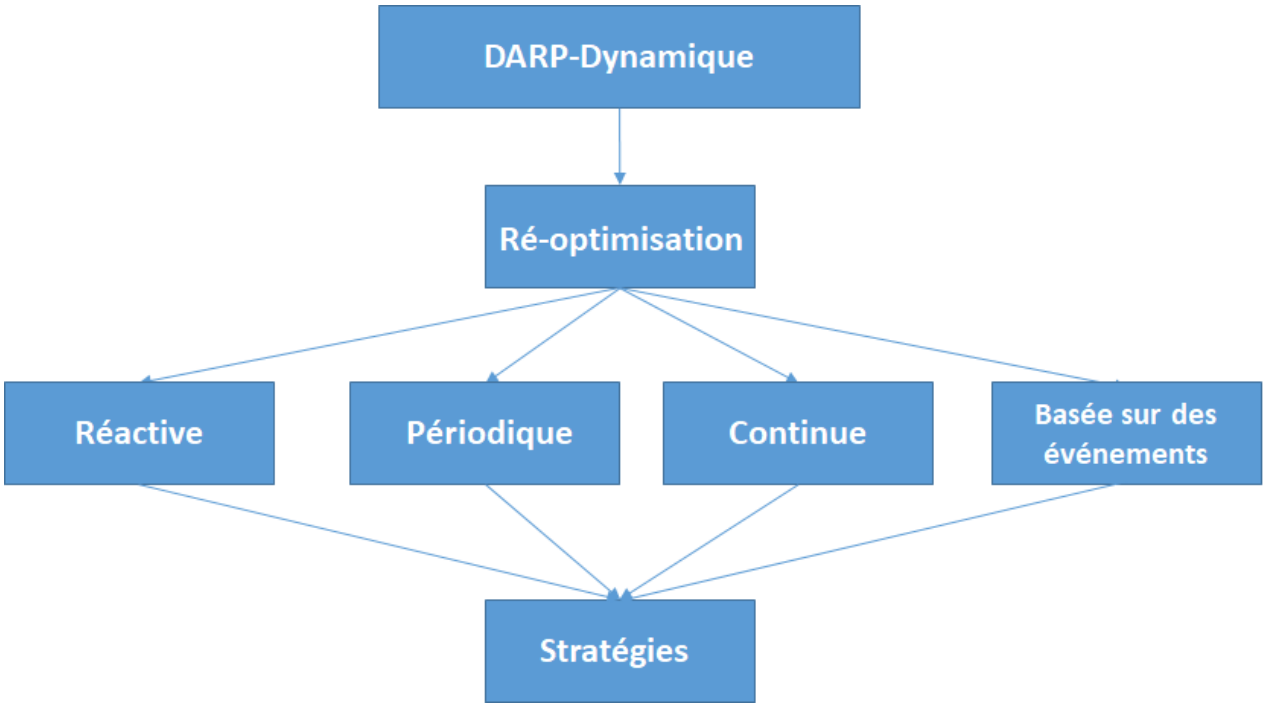

FIGURE 3.1 – Approches d'optimisation

Plusieurs stratégies d'optimisation ont été proposées dans la littérature, elles peuvent être divisées en trois catégories :

— Heuristique d'insertion : Les heuristiques d'insertion sont largement utilisées pour résoudre les DARP dynamiques surtout sur des problèmes à grandes échelles où une solution doit être fournie très rapidement [Ho et al., 2018], [Markovic et al., 2015].

— Heuristique d'insertion suppression : Dans la plupart des DARP dynamiques, une modification du plan de routes en cours est parfois nécessaire pour une meilleure prise de décision. En effet, plutôt que de simplement rejeter des nouvelles demandes, il est parfois nécessaire de déplacer d'autres demandes (déjà insérées) d'un véhicule à un autre. Généralement, le réaménagement s'effectue par une suppression des demandes à déplacer avant de les ré-insérer. Dans [Vallee et al., 2020], les auteurs ont développé trois heuristiques de suppression ré-insertion : heuristique "HDR" basée sur le concept de destruction/réparation, heuristique "GH" basée sur le principe de la chaîne d'éjection permettant de modéliser le problème de réinsertion comme un problème de CSP (plus court chemin sous contrainte de ressource) et une heuristique "IGH" qui est une version améliorée de l'heuristique "GH".

— Heuristiques à deux phases : Ce type de stratégie est plus adapté aux approches basées sur des événements. La première phase consiste à l'éxécution d'une heuristique d'insertion rapide lorsque une nouvelle demande arrive. Tandis que la deuxième est exécutée après chaque événement déclencheur. dans [Coslovich et al., 2006] les auteurs ont proposé dans la deuxième phase une recherche locale effectuée à chaque arrêt de véhicule. Dans [Beaudry et al., 2010], [Mitrovic-Minic et al., 2004] les auteurs ont proposé une deuxième phase basée sur une recherche tabou après chaque nouvelle demande ou annulation.

— Métha-heuristique continue : Ce type de stratégie est plus adapté aux approches continues. La plupart des méthodes proposées dans la littérature sont basées sur des métaheuristiques telles que la recherche tabou parallèle [Attanasio et al., 2004], algorithme génétique [Benyahia and Potvin, 1998] [Cheung et al., 2008], Variable Neighborhood Search (VNS) [Schilde et al., 2011], ...

3.2.1.2 Contraintes considérées dans la littérature

La plupart des études se concentrent sur la prise de décision opérationnelle dans le but de minimiser les coûts de transport et les retards. Les coûts associés aux heures supplémentaires relatives aux chauffeurs sont entièrement négligés, bien que celle-ci puissent entraîner des coûts importants pour les entreprises.

La majorité des problèmes de DARP qui tiennent compte des besoins en main-d'œuvre concernent que la version statique. Dans ces problèmes, les DARP comprennent des requêtes nécessitant des équipages différents ([Parragh et al., 2009] ; [Zhang et al., 2015] ; [Lim et al., 2017]), les horaires des pauses-repas préfixer à l'avance ([Parragh et al., 2009] ; [Liu et al., 2015] ; [Zhang et al., 2015] ; [Lim et al., 2017]), et la synchronisation des véhicules et des assistants (par exemple, [Lim et al., 2017]). Aucun de ces types de contraintes n'a été traité dans la version dynamique.

3.2.2 Synthèse

Plusieurs méthodes peuvent être élaborées pour résoudre un problème de tournées de véhicules dynamiques. Les méthodes qui ont montré une bonne qualité de solution repose sur la ré-optimisation basée sur des événements. En effet, la plupart des recherches concernant les problèmes de tournées de véhicules dynamiques utilisent une approche périodique et ne tiennent pas compte des demandes à caractère urgent.

D'autre part, la plupart des recherches concernant le PDPTW ou le DARP dynamique utilisent une approche réactive et ne tiennent pas compte de l'étroitesse des périodes inter-arrivées. Ils supposent que le temps entre les arrivées est suffisamment long. Cependant, cette hypothèse est probablement incorrecte, en particulier dans le cas de problèmes à grande échelle.

Finalement, aucun travail dans la littérature n'a traité les contraintes liées au "besoins en main-d'œvre", à l'attribution de pauses et la minimisation des heures supplémentaires des chauffeurs dans la version dynamique.

Ce qui rend le DMTP particulièrement difficile à aborder est la combinaison de certaines caractéristiques qui prises séparément portent moins de challenges.

En effet, gérer des équipages aux caractéristiques différentes avec une flotte de véhicule hétérogène devient particulièrement délicat lorsque les ambulanciers ont des horaires de travail différents, des pauses à prendre en compte et les équipages peuvent être recomposés à volonté durant la journée.

Comme dans le cas statique, il découle de ces propriétés une synchronisation entre les tournées respectives de chaque chauffeur et véhicule, ce qui a tendance à augmenter naturellement la complexité algorithmique des procédures combinatoires classiques d'optimisation.

3.3 Description générale du problème

Dans cette section nous décrivons les entités impliquées et le déroulement d'une régulation en insistant sur la diversité des décisions à prendre et les difficultés rencontrées par les régulateurs dans la prise de décisions.

Cette section est présentée dans le cadre des pratiques françaises mais donne un aperçu des caractéristiques majeures de transport sanitaire dont la plupart ont déjà été évoquées dans le chapitre traitant de la planification pour le lendemain.

3.3.1 Régulation du transport sanitaire

Cette section décrit la gestion de transport des usagers en temps réel. Sur le terrain, nous avons visité plusieurs grandes entreprises de transport sanitaire ayant plus de 1000 demandes/jour. Nous avons pu recueillir ainsi des informations sur le transport de leurs usagers et prestations. Notre objectif était de définir le problème le plus générique possible et qui répond aux demandes de l'ensemble des clients.

3.3.1.1 Requête de transport

Une requête de transport sanitaire consiste à transporter des usagers depuis/vers des pôles de santé pour des raisons de soins ou de diagnostic ayant des difficultés pour assurer leurs déplacements d'une manière autonome.

Le service de régulation reçoit des demandes de transport sanitaire parvenant de deux types de clients :

— Les établissements hospitaliers liés parfois par de conventions avec la compagnie d'ambulance.

— Les usagers particuliers généralement munis d'une prescription médicale.

Trois types de transport sont à distinguer dans une journée de régulation :

- **Transports connus la veille :** Les transports sollicités avant la journée de prise en charge.
- **Transports de la journée :** Les demandes de transport qui arrivent au cours de la journée.
- **Transports urgents :** Les transports urgents qui nécessitent une prise en charge dans les minutes qui viennent. Ce service qui vient en complément du SMUR et en particulier la nuit peut être contractualisé avec une forme d'astreinte.

Un quatrième type de demandes pourrait être défini en considérant des demandes supposées ou probables. C'est l'exemple des transports allers et retours. Le retour n'est pas toujours réservé en même temps que l'aller mais le régulateur peut l'anticiper et parfois définir l'heure de départ du retour. Par exemple les durées de dialyse sont très régulières. Parfois des parcours un peu plus complexes peuvent même être anticipés, comme la consultation, le passage au laboratoire d'analyses et le retour au domicile.

Le pourcentage de chacun de ces types de requêtes permet d'étudier plusieurs contextes réels. Les urgences de type SMUR sont marginales dans le contexte étudié et de nombreux travaux leurs sont dédiées. Le ratio entre les requêtes connues la veille et celles qui apparaissent durant la journée définissent la dynamicité du problème. Comme nous le verrons dans la partie expérimentale, ce paramètre influe sur la performance des stratégies de planification. En général, le niveau de dynamicité est stable pour un secteur d'opération donné.

3.3.1.2 Ambulancier

Le rôle de l'ambulancier est d'assurer le transport des usagers entre le lieu du rendez-vous et la destination. Dès l'arrivée au lieu de prise en charge, l'ambulancier installe l'usager, vérifie les documents et le transporte vers le lieu de dépose. À destination, l'ambulancier fait signer à l'usager des documents pour attester l'accomplissement de la mission.

Le régulateur envoie à l'ambulancier au fur et à mesure dans la journée les indications sur les prochaines missions : heure de prise en charge, destination, pauses, changement de coéquipier, changement de véhicule, ...

Dans les grandes compagnies, un ambulancier peut être affecté à une ou plusieurs agences. Une agence de départ avec laquelle l'ambulancier démarre son activité et une autre de débauche qui marque la fin de son service. De plus, il peut changer de véhicule d'une agence à l'autre. Généralement, c'est très rare que cela se produise. Ainsi dans ce travail, nous nous limitons à une seule agence par ambulancier. Cette extension sera intégrée dans le chapitre suivant.

Les journées de travail des ambulanciers sont décidées sur la base d'un planning bimensuel préparé par le régulateur. Ce planning comporte les heures de début et de fin de

travail journalières théoriques. La veille, et grâce au planning de la veille pour le lendemain, le régulateur remet à jour les heures de travail. Ces horaires peuvent encore être modifiés durant la journée.

Le régulateur gère aussi les pauses des ambulanciers. Ces dernières sont de trois types :

1. **La pause déjeuner** est fixée pour chaque ambulancier qui commence sa journée avant 11 :00 et finit après 11 :15. Cette pause dure une heure et doit être prise à l'agence. Sinon, elle peut être raccourcie d'une demi-heure et prise à l'extérieur, ce qui engendre un coût supplémentaire à l'employeur.
2. **La pause dîner** est fixée pour chaque ambulancier qui commence sa journée avant et finit après 18 :00 avec les mêmes modalités que la pause déjeuner.
3. **La pause de repos** est fixée à chaque ambulancier qui travaille 6H en continu (sans pause). Cette pause dure 20 minutes et est généralement permise sur la route pour de longs trajets.

3.3.1.3 Véhicule

Il existe trois types de véhicules dans le transport sanitaire français, à savoir :

— L'ambulance qui nécessite 2 ambulanciers, un chauffeur et un assistant, avec des diplômes spécifiques (un diplôme d'ambulancier d'état pour le chauffeur et un diplôme d'auxiliaire pour l'assistant). Ce type de véhicule ne permet pas le partage de l'ambulance entre plusieurs usagers pendant le transport.

— Le véhicule sanitaire léger (VSL) qui nécessite un seul ambulancier avec le diplôme d'auxiliaire. Ce véhicule peut transporter plusieurs usagers à la fois.

— Le taxi qui nécessite une licence de conducteur et une qualification de taxi-ambulance. Ce véhicule peut transporter plusieurs usagers à la fois.

Comme décrit dans la partie introductive, le mode de remboursement diffère selon le type de véhicule utilisé. Les règles de calcul sont relativement complexes et différentes d'un département à l'autre avec des variations en fonction du kilométrage entre origine et destination, l'heure de la journée et si le transport est partagé ou pas. Dans notre modélisation, un bénéfice constant propre à chaque véhicule et requête est considéré.

En général, chaque véhicule est attribué à un dépôt (agence) où il commence et finit la journée. Mais certains chauffeurs ne respectent pas cette règle, notamment ceux qui ramènent le véhicule chez eux le soir.

3.3.1.4 Régulateur

Le régulateur est chargé de recevoir les appels des clients, de satisfaire au mieux leurs besoins (horaires, types de véhicules, contraintes spéciales,...) et d'organiser le travail des ambulanciers.

A chaque appel, le régulateur négocie une plage horaire de prise en charge et de dépose, les types de transport possibles et traite certaines autres demandes particulières de l'usager (ex : chaise bariatrique). En général, le régulateur doit fournir une réponse d'acceptation ou refus immédiatement à son interlocuteur. Ce fonctionnement implique que le détail de la nouvelle demande (lieux exacts , matrices de distances, contraintes horaires, type de transport, ...) n'a pas forcément été intégré dans le système informatique pour supporter la décision.

Même dans les régulations de taille modeste, deux régulateurs travaillent dans une journée avec une période de chevauchement entre des postes respectifs de 06 :00 à 13 :00 et 10 :00 à 18 :00. Le premier assure la gestion des transports jusqu'à l'arrivée du deuxième régulateur qui prend le relais. Le premier régulateur finit son service par des tâches administratives (facturation, etc.). En fin de journée, lorsque l'activité est plus calme, le deuxième régulateur prépare le plan pour le lendemain sur la base des demandes déjà enregistrées. Bien sûr ce fonctionnement observé chez l'un de nos clients, varie en fonction de la taille des régulations et du nombre d'opérateurs.

Pendant toute la journée, le régulateur dispose d'un plan prévu dont la profondeur dépend de ses préférences ou de celles de l'entreprise. Ce planning représente les prochaines destinations prévues (prise en charge , dépose, pause, retour agence, changement de véhicule) de chaque ambulancier.

À une certaine fréquence, qui elle aussi dépend du régulateur, les nouvelles demandes non encore traitées sont insérées dans le plan prévu avec l'objectif de réduire au maximum les coûts de transport, les retards et les heures supplémentaires des chauffeurs.

Le régulateur transmet les prochaines destinations aux ambulanciers quand il estime que cela est nécessaire. Dès que l'ambulancier reçoit ses prochaines destinations, la décision d'affectation de ces missions au équipages et véhicules n'est remise en cause qu'en cas de force majeure.

3.3.2 Différentes approches de régulations

Durant nos visites chez certains clients de Lomaco, nous avons observé deux types d'approches de régulation manuelles.

3.3.2.1 Méthode des petits pas :

Le régulateur attend qu'une équipe finisse une mission ou prenne son service pour lui attribuer la prochaine. En cas d'insertion d'une nouvelle mission, il opte pour une équipe disponible.

Cette approche présente un avantage : le régulateur réduit en effet l'incertitude au moment de prendre sa décision. Il ne perd pas de temps à remettre en cause des choses déjà planifiées. L'inconvénient majeur, par contre, est que manuellement, il est difficile d'intégrer une vision globale de la planification à chacune de ces décisions ; les modifications

d'équipages sont généralement occultées. D'autant plus si plusieurs véhicules achèvent leurs missions dans un délai limité. Le niveau de stress du régulateur est élevé. Et ce processus serait inefficace en cas de panne du système ou indisponibilité du régulateur. Avec cette approche, il n'y a pas la possibilité d'informer à l'avance les ambulanciers de leur plan de route. Ce qui ne leur permet d'avoir une visibilité quant à leurs pauses, lieux du déjeuner, heure de débauche, ...

Pour prendre sa décision, le régulateur filtre les missions pour lesquelles il faut rapidement affecter un équipage et affecte celle qui lui paraît être la plus critique. La difficulté est d'intégrer que d'autres équipages seront bientôt libres et peuvent être mieux placés pour satisfaire certaines requêtes. Dans le cas d'insertion de nouvelles missions ce sont les véhicules disponibles qui sont filtrés par proximité.

Une des clefs pour améliorer la performance de cette approche est d'être capable de résoudre un problème de planification très rapidement avec une vision globale à chaque arrivée d'équipage.

3.3.2.2 Méthode d'anticipation :

Le régulateur construit les plans de route des chauffeurs sur une période assez longue pour couvrir plusieurs courses. L'insertion de nouvelles missions doit se faire dans des plans de routes existants pour chaque équipage.

Cette approche permet alors de sécuriser la régulation sur une période et de tirer profit d'une vision globale des données. Le plan de route peut être fourni à l'avance aux équipages selon le choix du régulateur.

Le principal inconvénient de cette approche est que de nombreux événements incertains peuvent se produire durant la réalisation du plan prévisionnel. Le régulateur peut réactualiser son plan souvent, ce qui est assez lourd. Sinon, il peut avoir une stratégie conservative qui va produire des solutions moins efficaces au final sans éviter certaines replanifications en cas de perturbations majeures.

Face à ce problème plus complexe, les régulateurs observés construisent les routes sur la période à planifier équipage par équipage.

Une des clefs pour améliorer la performance de cette approche est d'être capable de résoudre un problème de planification avec une vision globale qui soit robuste face aux perturbations.

3.3.2.3 Commentaires :

Outre le rythme et la portée des planifications, le régulateur a le choix de sa stratégie de communication aux chauffeurs et usagers. Sauf cas exceptionnel, le respect des fenêtres de temps garantit la qualité de service à l'usager sans être obligé de lui communiquer des modifications horaires. Pour les chauffeurs, une communication anticipée (incluant des annulations et modifications) ou de dernière minute ont un impact sur la

méthode de régulation choisie. La méthode des "petits pas" ne pouvant pas permettre de communication anticipée.

Pour les deux approches les régulateurs ont des besoins en outils d'aide à la décision pour être efficaces dans les temps impartis.

3.4 Définition du Problème de Transport Médicalisé Dynamique (DMTP)

Dans cette section, nous décrivons le problème de planification du transport sanitaire que nous traitons. Plusieurs caractéristiques communes au problème statique sont répétées ici pour permettre une lecture indépendante de ce chapitre.

Dans le DMTP, six types de décisions doivent être prises :

1. Accepter ou de refuser chaque nouvelle demande durant la même journée. cette décision doit se prendre immédiatement à l'arrivée de la demande. On suppose ici que les informations relatives à la demande sont saisies dans le système par le régulateur.
2. Affecter chaque demande acceptée à un équipage et un véhicule.
3. Assigner les temps de pause aux ambulanciers.
4. Déterminer l'heure de fin de service de chaque ambulancier.
5. Les compositions et recompositions d'équipages.
6. Les horaires de passage. Les horaires sont ceux au plus tôt en fonction des autres décisions.

Généralement la première décision est prise sans saisir le descriptif complet de la requête. Plusieurs approches peuvent prendre en considération cette pratique.

1. Installer une commande vocale liée au téléphone qui remplit directement les champs des requêtes.
2. Générer automatiquement les données qui manquent approximativement à partir d'informations restreintes. Par exemple la distance entre deux adresses est approchée par la distance moyenne entre les codes postaux.
3. Développer une stratégie d'admission dédiée, travaillant uniquement sur un jeu de données minimal.

3.4.1 Réseau de transport

Soit un graphe orienté $G = (V, A)$, où V représente l'ensemble de nœuds et $A = \{(i, j)|i, j \in V\}$ est l'ensemble des arcs. Les nœuds $P = \{1, \ldots, |P|\}$ sont les ramassages, les nœuds $D = \{1, \ldots, |D|\}$ les déposes et les nœuds $O = \{1, \ldots, |O|\}$ sont les

dépôts. Les missions de transport sont représentées par l'identifiant de leur nœud de ramassage.

Les nœuds de G n'identifient pas des lieux géographiques distincts. D'ailleurs le temps de transport t_{ij} entre deux nœuds i et j peut être différent de t_{uv} avec u au même lieu que i et j que v. Cela permet d'attribuer des temps différents au même trajet effectué à des périodes différentes.

Un coût c_{ij} relatif à la distance est également associé à chaque arc.

Contrairement à de nombreux modèles de la littérature, les coûts et les temps de trajet peuvent violer l'inégalité triangulaire, puisque ces valeurs ne sont pas proportionnelles sur les réseaux routiers réels. De plus, dans la pratique, les habitudes des conducteurs ne suivent pas l'optimisation du même critère sur chaque arc et donc l'inégalité triangulaire peut être violée pour n'importe quel poids.

L'hypothèse d'inégalité triangulaire permet de diminuer la complexité de plusieurs procédures algorithmiques ou de garantir la faisabilité ou l'optimalité de leur résultat. Les algorithmes que nous proposons fonctionnent sans cette hypothèse.

Le temps trajet t_{ij} entre deux nœuds i et j représente le temps du trajet prévu qui peut être de celui réel.

3.4.2 Chauffeurs et dépôts

Un chauffeur e est affecté à un seul dépôt $o_e \in O$ et il peut conduire un ensemble limité de véhicules K_e qui sont tous basés à o_e. Par conséquent, la composition / recomposition de l'équipage ne peut être effectuée qu'au dépôt et par les conducteurs et les véhicules affectés à ce dépôt. L'ensemble E de tous les chauffeurs est une partition des ensembles $\{E_o : o \in O\}$ des chauffeurs affectés à chaque dépôt.

Une fenêtre de temps de travail $[ew_e, lw_e]$ est définie pour chaque ambulancier e. L'ambulancier ne peut pas démarrer sa journée avant ew_e. Finir plus tard que lw_e est autorisé mais pénalisé par une fonction en escalier dans la fonction objectif 3.2. Chaque t minute où l'ambulancier dépasse son heure de fin de service prévue (lw_e) est pénalisée par un coût d'heures supplémentaires CHS qui suit une fonction en escalier sur-linéaire décrite par l'équation (3.1) et la figure 3.2.

$$CHS(x) = \beta^{n_x}.c$$

$$n_x = \begin{cases} argmin\{n \in \mathbb{N} : lw_e + nt < x \leq lw_e + (n+1)t\} & \text{si } x > lw_e \\ 0 & \text{sinon} \end{cases} \tag{3.1}$$

Le pas de temps t où le coût change est généralement d'une heure car chaque heure supplémentaire entamée est payée. Les paramètres β et c sont fixés à 4 et 8 respectivement.

Deux types de pause sont modélisés. Le premier type correspond aux pauses déjeuner ou dîner. Une fenêtre de temps sur le début de la pause est imposée $[eb_{be}, lb_{be}]$ pour

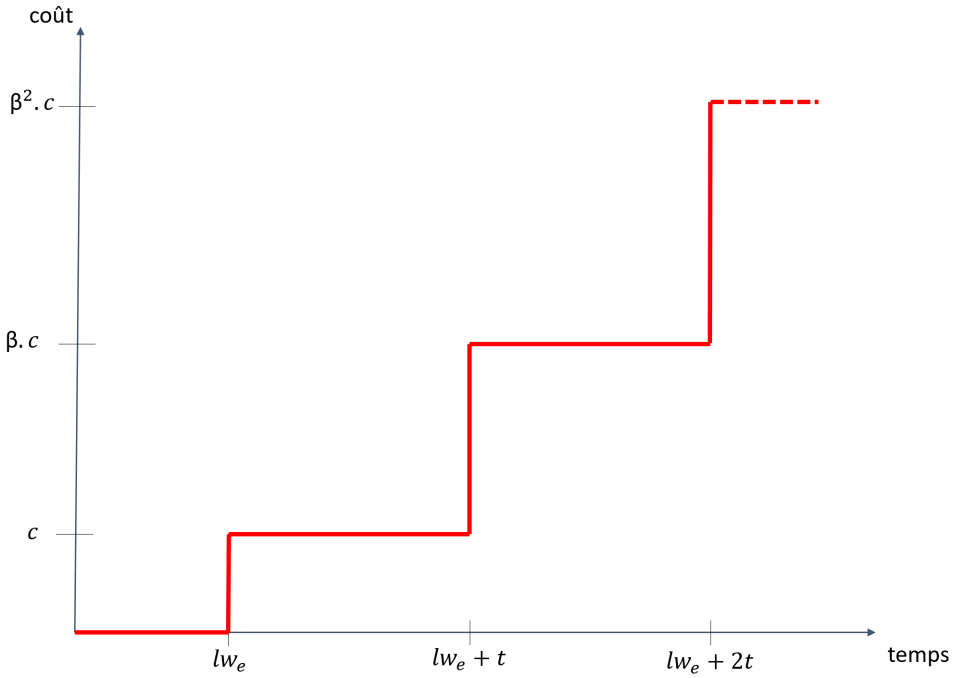

FIGURE 3.2 – Coût associé aux heures supplémentaires

chaque pause b du chauffeur e. Cette interruption dure δ_{be} minutes et ne s'applique que si le chauffeur commence à travailler avant l'heure et_{be} et termine après lt_{be}. Sont également définis, un temps limite de travail Δ_{be} après lequel il faut déclencher la pause, un indicateur pb_{be} si des passagers sont admis durant la pause et un indicateur loc_{be} si la pause doit être faite au dépôt ou non.

Tous les indicateurs n'ont pas d'importance pour tous les types de pauses. Les deux types de pauses intégrées dans les expérimentations sont la pause repas et celle de vingt minutes après six heures de travail. Nous référons le lecteur au tableau descriptif des valeurs des attributs du chapitre précédent.

Nous rappelons qu'aucun usager ne peut rester dans le véhicule pendant une pause au dépôt. Mais à l'extérieur du dépôt, les usagers peuvent partager la pause déjeuner avec les chauffeurs ; pour de longs trajets par exemple.

Les attributs binaires, nommés qualifications, sont attribués à chaque chauffeur. L'ensemble Q de qualifications permet de définir des contraintes fines sur les indemnités de conduite et les compétences requises pour répondre à certaines demandes.

3.4.3 Véhicule

Les véhicules $k \in K = U_{e \in E} K_e$ sont utilisables en fonction de leurs besoins en équipage A_k (une liste de qualifications), du mode de transport (assis ou couché) m_k, du nombre maximal de passagers (sans chauffeurs) κ_k et du coût par kilomètre αk. Ce coût dépend de la consommation de carburant et des péages autoroutiers.

3.4.4 Demande

Chaque requête $i \in P$ est définie par un service d'origine p_i et une destination d_i un temps de service s_{p_i} (resp. s_{d_i}), une fenêtre de temps de service de démarrage $[es_{p_i}, ls_{p_i}]$ (resp. $[es_{d_i}, ls_{d_i}]$), le nombre de places occupées q_i (certains accompagnons peuvent être transportés), acceptation de partage $\rho_i \in \{0, 1\}$ et un ensemble d'exigences concernant l'équipage et le véhicule. Un ensemble de véhicules appropriés B_i est défini selon le mode de transport prescrit ω_i et une liste Ω_i d'exigences relatives aux dispositifs tels que les équipements médicaux des usagers. Outre les contraintes liées aux véhicules, l'usager peut exiger une liste de caractéristiques binaires sur les conducteurs décrits par les ensembles M_e. Deux configurations sont autorisées : les contraintes douces et dures. Les listes correspondantes sont notées respectivement Φ et Ψ. La force et les langues parlées sont les compétences requises les plus courantes. $\Omega_i = \{\Phi_{i,1}, ..., \Phi_{n_{1,i},1}, \Psi_{i,1}, ..., \Psi_{n_{2,i},1}\}$.

L'arrivée en avance à chaque service j c'est-à-dire avant l'heure es_j n'est pas autorisée. En revanche, l'arrivée en retard au service (après ls_j) est autorisée mais soumise à une pénalité.

Notez que les heures de départ des fenêtres horaires sont souvent calculées en pratique à partir de l'heure au plus tard de dépose ls_{d_i}, un temps de trajet maximal et t_{p_i,d_i} pour une demande sortante i (sortant signifie que le point de dépose d_i représente le point de rendez-vous). Les demandes entrantes (retour d'un point de rendez-vous vers la maison et depuis l'hôpital) sont construites symétriquement à partir de l'heure de sortie au plus tôt es_i au lieu de l'heure de dépose au plus tôt. Le temps de trajet maximal permet d'absorber les retards sur le trajet et la durée du service, et permet le covoiturage.

Les demandes hors celles connues la veille sont reçues de manière dynamique dans la journée, soit ha_i représentent l'instant de réception de la demande i au système (si la requête i est connue la veille $ha_i = 0$). D'autres données qui concernent les requêtes et qui peuvent apparaitre de façon dynamique durant la journée comme une éventuelle annulation ou modification d'horaires de passage. Soient, an_i correspond à l'instant de la réception d'une annulation de la demande i et mod_i représente l'instant de réception d'une demande de modification d'horaires avec $[es'_{p_i}, ls'_{p_i}]$ et $[es'_{d_i}, ls'_{d_i}]$ représentent les nouvelles fenêtres de temps de prise en charge et de dépose.

3.4.5 Objectifs et contraintes

Le problème vise à concevoir de manière efficace les itinéraires de prise en charge et de dépose des usagers sous les contraintes suivantes : (i) Contraintes de visite : chaque lieu de ramassage et de dépose d'une demande n'est visité qu'une seule fois ; (ii) Contraintes de dépôt : l'itinéraire de chaque véhicule et ambulancier commence et se termine aux dépôts correspondants ; (iii) Contraintes de capacité : les capacités des véhicules ne peuvent pas être dépassées ; (iv) Contraintes de parité : les lieux de ramassage et de dépose de chaque demande doivent être visités par le même véhicule et équipage ; (v) Contraintes

de priorité : chaque usager doit être pris en charge avant d'être déposé ; (vi) Contraintes de ressources : l'affectation " équipage / véhicule " à l'usager doit respecter le type du véhicule, le nombre d'ambulanciers et les diplômes adéquats et l'ensemble des exigences dures mentionnés au niveau de la demande ; (vii) Contraintes de pauses : Des pauses doivent être attribuées aux ambulanciers durant leur journée de travail ; (viii) Contraintes de changement du véhicule : Tout changement de véhicule effectué par les ambulanciers doit être effectué au dépôt ; (ix) Contraintes d'amplitude : tous les ambulanciers ne doivent pas dépasser l'amplitude de travail maximal ; et (x) Contraintes de simultanéité : Ne pas transporter deux usagers au même temps dans le véhicule pour ceux qui le préfèrent.

L'efficacité de l'itinéraire est mesurée par un compromis entre plusieurs objectifs contradictoires :

1. objectif de satisfaction : minimiser le nombre de demandes non servies ;
2. objectif de retard : minimiser le carré du retard $(hr_i - ls_i)$ sur les arrivées aux services (ramassage comme dépose) ;
3. objectif des heures supplémentaires : minimiser l'écart entre la fin de service réelle et la fin de service prévue des ambulanciers ;
4. maximiser le bénéfice associé aux demande couvertes ;
5. minimiser les coûts kilométriques parcours par les véhicules et
6. minimiser le nombre de contraintes (exigences) souples des demandes acceptées (les instances testées ne contiennent pas d'exigences).

$$\min Z = w_1 U + w_2 R + w_3 CHS - w_4 B + w_5 km + w_6 C; \tag{3.2}$$

Avec U correspond au nombre de requêtes non servies, R représente la somme des retards en minutes, km correspond à la distance totale parcourue par les véhicules, B représente la somme des bénéfices des demandes servies, C correspond au nombre des exigences souples non respectées, CHS définit le coût associé aux heures supplémentaires effectuées par ambulanciers et les w_i représentent les poids à donner pour valoriser chaque objectif déjà cité. La fonction objectif Z est représentée comme une somme pondérée des objectifs.

Les poids utilisés sont toujours les mêmes dans les expérimentations : $w_1 = 5000$; $w_2 = 2$; $w_3 = 1$; $w_4 = 1$; $w_5 = 1$ et $w_6 = 1$.

La fonction objectif Z diffère de la fonction objectif du problème statique de la veille car les retards subis doivent être intégrés (objectifs 2 et 3). Dans la partie statique des fenêtres de temps dures permettent d'interdire les retards tout simplement.

Dans les entreprises de transport sanitaire visitées, le pourcentage de demandes de transport réservées plus d'un jour à l'avance va de 30% à 70%. Le reste des demandes provient progressivement durant la journée.

Dans un tel environnement, un itinéraire planifié pour des ambulanciers et véhicules est une séquence de services qui ont été planifiées mais qui n'ont pas encore été traitées. Les événements dynamiques qui peuvent apparaître dans la journée comprennent : (i) Les événements liés au usager : arrivée de nouvelles demandes de transport ou retard dans la disponibilité, (ii) Des annulations de demandes.

3.4.6 Hypothèses et simplifications par rapport au problème réel

Une série d'hypothèses sont faites pour simplifier l'évaluation des solutions ou faire face au manque d'informations disponibles en temps réel.

1. Les pauses de dîner et de déjeuner sont attribuées toujours au dépôt avec une durée d'une heure : Comme mentionné précédemment, les pauses peuvent être attribuées n'importe où et avec une durée de moins d'une heure mais avec une pénalité, la manière de calculer cette pénalité diffère d'une entreprise à une autre. Afin de ne pas rendre le problème complexe encore plus et de rajouter plusieurs objectifs dans tous les sens, nous avons limité cette décision toujours au dépôt et avec une durée d'une heure.
2. Tout changement de véhicule se fait uniquement au dépôt du chauffeur : un chauffeur peut normalement être attribué à plusieurs dépôts à la fois en plus de son dépôt principal, par conséquent, l'ambulancier peut changer de véhicule dans n'importe quel dépôts. La décision de changement de véhicule hors le dépôt principal de l'équipage arrive en réalité très rarement et dans des cas extrêmes. En effet, et en tenant compte du nombre exponentiel des décisions qui peut générer cette propriété, nous avons décidé de ne pas tenir compte de cet aspect.
3. Un chauffeur ou un véhicule démarre et finit sa journée au dépôt.
4. Les heures de début de service réelles des chauffeurs sont fixées par leur heure de début prévue. Les heures de début de service sont généralement décidées la veille, en utilisant le planning de la veille pour le lendemain, le fait de changer l'horaire de travail d'un chauffeur le jour même est généralement à éviter.
5. La trajectoire d'un véhicule ne peut pas être déviée entre deux points. Il peut cependant l'être pendant une mission s'il s'arrête pour du covoiturage. Cette hypothèse sert à palier l'imprécision actuelle sur la localisation des véhicules entre deux points de service.
6. Un équipage inactif (sans missions prévues à venir) attend quinze minutes avant de rentrer au dépôt. Sur le trajet retour, la trajectoire d'un véhicule ne peut pas être déviée.
7. Un équipage quitte le service courant à l'heure au plus tard possible pour arriver à l'heure au plus tôt possible à la prochaine destination.

3.5 Problèmes intermédiaires considérés

Afin d'éviter de faire les analyses de résultats avec plusieurs interactions de contraintes complexes, nous avons décidé de travailler sur un problème plus simple et d'intégrer indépendamment les contraintes essentielles. Ces contraintes sont :

— Les ambulanciers sont contraints par des horaires de début et de fin de service prévus.

— Pauses (de déjeuner, de dîner et de repos)

— Plusieurs types de véhicules à considérer

3.5.1 Problème simplifié P_1 – DARP

Le premier problème "Simplifié" auquel nous nous intéressons est caractérisé par :

— Les ambulanciers sont tout le temps disponibles $ew_e = 0$, $lw_e = \infty$.

— Aucune pause à donner aux ambulanciers

— Toutes les demandes sont transportées par un seul type de véhicule (VSL).

Dans ce problème, chaque ambulancier dispose de son propre véhicule ; $|K_e| = 1$ à sa base. Les ambulanciers étant toujours disponibles et les véhicules homogènes il n'y a pas d'intérêt à planifier des changements de véhicules. Ce problème est proche du DARP dynamique étudié dans la littérature.

3.5.2 Problème P_2 – DARP avec temps de travail

Le deuxième problème "P_2", est le problème simplifié P_1 en lui rajoutant la contrainte des heures d'embauche et débauche des ambulanciers (spécificités 4 et 5). Chaque ambulancier e dispose d'une heure de début et fin prévue $(ew_e; lw_e)$. Le dépassement de l'heure de fin prévue est pénalisé dans la fonction objectif voir Figure 3.2. Le repoussement de la date de fin de travail des chauffeurs est autorisé autant de fois que possible et se fait par tranche de t minutes ($t = 1$ heure dans nos expérimentations). Une fois que l'ambulancier est libéré, il ne pourra plus être affecté à aucune nouvelle demande.

3.5.3 Problème P_3 – DARP avec pauses

Dans ce problème P_3, nous considérons le problème simplifié P_1 en lui rajoutant la contrainte de la prise obligatoire des pauses par les ambulanciers (spécificité 6).

Trois pauses sont considérées- : déjeuner, dîner et les six heures maximales de travail continu(voir Tableau 3.1).

TABLE 3.1 – paramètres des pauses

Notation	Pause déjeuner	Pause diner	Pause travail
δ	1 heure	1 heures	20 minutes
$[eb, lb]$	$[11:00, 14:00]$	$[18:00, 20:00]$	$]-\infty, +\infty[$
$]et, lt[$	$]11:00, 11:15[$	$]18:00, 18:15[$	$]-\infty, +\infty[$
Δ	$+\infty$	$+\infty$	6 heures
pb	True/False	True/False	False
loc	Dépôt/Sur la route	Dépôt/Sur la route	Sur la route

3.5.4 Problème P_4 – DARP avec plusieurs types de véhicules

Dans P_4, on considère le problème simplifié P_1 mais on suppose que nous avons plusieurs types de véhicules (spécificités 1, 2, 3, 7). Désormais, une demande de transport peut être prise en charge par un ou plusieurs types de véhicules et des types de véhicules nécessitant un équipage composé de deux ambulanciers.

Contrairement aux problèmes précédents, dans celui-là, les ambulanciers peuvent changer plusieurs fois de véhicule et d'équipage durant la journée. Par conséquent, chaque demande de transport doit être affectée à un équipage (composé d'un ou deux ambulanciers) et un véhicule. De plus le nombre de véhicules de chaque type est limité à chaque base.

3.6 Stratégies de planification

Dans cette section, nous décrivons les stratégies d'optimisation développées pour résoudre nos cinq problèmes, quatre problèmes simplifiés de P_1 à P_4 et le DMTP complet.

Deux stratégies représentent des comportements de régulateurs observés et une autre proposée autour d'algorithme d'optimisation. Elles se distinguent par le niveau d'optimisation des décisions mais aussi par la hiérarchisation des décisions et des contraintes en commençant plutôt par les activités (requêtes de transport) ou par les ressources (équipages et véhicules).

Les problèmes P_1 à P_3 ont servi à étalonner les trois stratégies. De cette comparaison, S_1 s'avère largement dominante et est donc la seule stratégie appliquée à P_4 et DMTP.

Pour l'ensemble des stratégies nous définissons dans la section 3.6.1 un cadre formel de description qui permet de concevoir un large panel de stratégies en jouant sur quelques paramètres.

3.6.1 Fonctionnement des stratégies et pré-traitements communs

Les trois stratégies suivent le même schéma avec à chaque appel à la stratégie, un plan prévisionnel qui est calculé sur un horizon futur à partir du précédent.

Pour les deux stratégies représentant les régulations manuelles, les requêtes pouvant être groupées en simultané, le sont de manière anticipée par rapport à la planification suivant la procédure de la section 3.6.1.3.

En entrée la stratégie prend un plan prévisionnel, les nouveaux événements depuis la dernière réalisation ainsi que la réalisation des variables aléatoires (l'heure d'arrivée réelle d'un équipage sur un service, l'heure de début de service réelle d'un ambulancier, ...).

3.6.1.1 Plan prévu

Un plan prévu Pl est un ensemble de trips T ; séquences de services avec un même équipage et un même véhicule.

Un exemple est donné dans la figure 3.3. Soient $Prec_x$ l'ensemble de trips qui précèdent le trip x et $Suiv_x$ l'ensemble de trips qui succèdent x. , $Prec_I = \emptyset$, $Suiv_I = \{II\}$; $Prec_{II} = \{I\}$, $Suiv_{II} = \{III, VI, \}$; $Prec_{III} = \{II\}$, $Suiv_{III} = \emptyset$; $Prec_{IV} = \{III, V\}$, $Suiv_{IV} = \{VII\}$; $Prec_V = \{IV\}$, $Suiv_V = \{VI\}$; $Prec_{VI} = \{III, V\}$, $Suiv_{VI} = \{VII\}$; $Prec_{VII} = \{VI\}$, $Suiv_{VII} = \emptyset$.

FIGURE 3.3 – Exemple plan prévu

Les décisions prises sont les affectations des équipages aux véhicules et aux requêtes et les heures de départ au plus tôt de chaque localisation. On note $E(i)$ l'équipage affecté à la requête i et hp_j l'heure prévue pour chaque service j. On peut en déduire les variables $prec(i)$ qui donnent le nœud précédent i pour son équipage. La date de fin de service

prévue de chauffeurs e est notée hfp_e. On considère que les heures prises de services prévues la veille ne sont pas remises en cause et peuvent juste générer du retard si elles ne sont pas respectées.

Les données révélées au fur et à mesure sont hr_j, l'heure réelle du service j, et $I(e)$, la localisation de l'équipage e courante s'il est en déplacement ou en route vers sa prochaine destination.

L'arrivée d'un véhicule à destination ou l'annulation d'une demande déclenche une mise à jour du plan prévu. Il s'agit de prévoir les heures de passage pour les requêtes servies à partir des dernières heures de passage connues pour chaque équipage. Les horaires de passage sont calculés au plus tôt. Cette procédure de programmation dynamique est très rapide et basée sur la formule de récurrence (3.3), où $prec(i, e)$ est le prédécesseur de i pour le chauffeur e.

$$hp_i = \begin{cases} hr_i & \text{si } i \text{ déjà visité} \\ \max\{es_i; \max_{e \in E(i)}\{hp_{prec(i,e)} + s_{prec(i,e)} + t_{prec(i,e),i}\}\} & \text{sinon} \end{cases} \tag{3.3}$$

Afin d'accélérer la procédure on peut limiter la procédure aux services dont au moins un des prédécesseurs ou eux-mêmes ont subi une modification.

Dans la suite du document, les notations T_x et T_{cur} représentent respectivement la date de la dernière planification et la date courante.

3.6.1.2 Schéma général d'une stratégie

Un cadre commun est défini pour la conception et la description de stratégies. Dans ce cadre, les valeurs données aux paramètres permettent de définir des stratégies très différentes, comme très anticipatrices et rigides ou réactives et flexibles.

Une stratégie de planification désigne :

- Une liste d'événements déclencheurs $EvDec$ pour une ré-optimisation du plan prévu avec en particulier deux modes. Le premier, suit une fréquence F de ré-optimisation prédéfinie. Le deuxième dépend de la réalisation du plan prévisionnel et des variables aléatoires (ex : arrivée d'une nouvelle requête, heure de planification d'une requête atteinte, arrivée d'un véhicule à destination, seuil de retard dépassé, ...)
- Pour chaque événement déclencheur à l'instant T_0 :
  - Un ensemble $\mathcal{D}_{fix}$ de décisions prises au préalable et qui ne peuvent pas être modifiées.
  - Un ensemble $\mathcal{D}_{modif}$ de décisions prises au préalable qui peuvent être modifiées avec une pénalisation.
  - Un ensemble $\mathcal{D}_{libre}$ de décisions prises au préalable ou non encore considérées qui peuvent être complètement remises en cause.

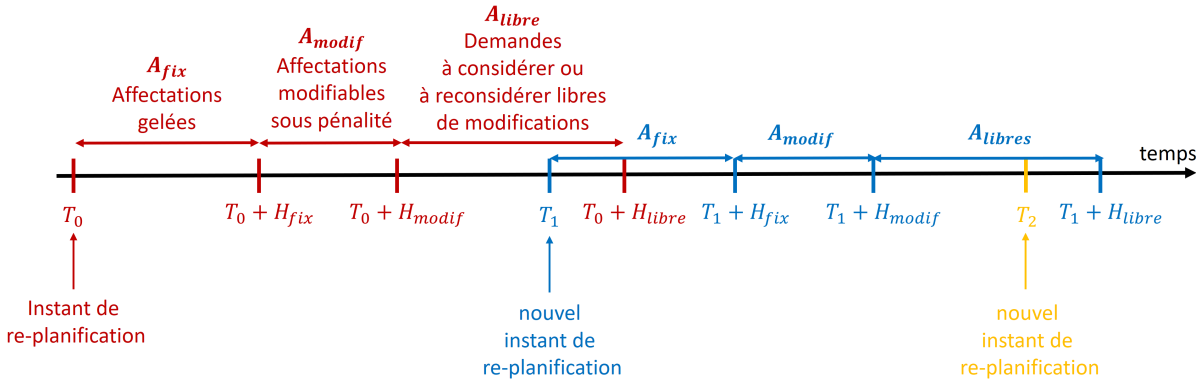

FIGURE 3.4 – Description des stratégies

— Un algorithme de prise de décisions pour les trois ensembles.

Ces ensembles sont construits essentiellement à partir de considérations temporelles. Les ensembles de décisions au $x$ième appel sont respectivement déterminés par les plages horaires : $[T_x, T_x + H_{fix}]$, $[T_x, T_x + H_{modif}]$ et $[T_x, T_x + H_{libre}]$. à partir de ces intervalles de temps sont construits les ensembles de requêtes A_{fix}, A_{modif} et A_{libre} avec les définitions des équations (3.4)-(3.6) et la figure 3.4.

Pour chaque requête i un attribut $hplan_i$ est calculé pour signifier à partir de quand un véhicule devrait être affecté à cette requête. Cet attribut a pour objectif d'anticiper la planification de requêtes a priori 'loin de tout' et de retarder la planification de requêtes a priori proches des véhicules.

$$A_{fix} := \{i \in |P| : hplan_i \leq T_x + H_{fix}\} \tag{3.4}$$

$$A_{modif} := \{i \in |P| : T_x + H_{fix} < hplan_i \leq T_x + H_{modif}\} \tag{3.5}$$

$$A_{libre} := \{i \in |P| : T_x + H_{modif} < hplan_i \leq T_x + H_{libre}\} \tag{3.6}$$

La contrainte de ne pas dérouter un équipage avant sa prochaine destination, implique que les requêtes pour lesquelles un équipage/véhicule est en route vers le lieu de ramassage, ne pourront pas être réaffectées à un autre équipage/véhicule.

Les décisions conservées d'une planification à l'autre sont limitées aux affectations d'équipages et non aux horaires de passage.

Ainsi à l'instant de replanification T_x, les décisions gelées $\mathcal{D}_{fix}$ correspondent aux requêtes déjà affectées à un équipage ($E(i) \neq \emptyset$) ou pour lesquelles le véhicule est en route à T_x. Les décisions modifiables sont celles prises à l'itération précédentes dans la bonne tranche horaire et qui ne sont pas gelées. Enfin, les requêtes libres sont celles qui sont dans l'horizon de planification ($hplan_i \leq T_x + H_{libre}$) et qui ne sont pas dans A_{fix} ou A_{modif} ; cela inclut les requêtes non couvertes à l'itération précédente ou qui ont été révélées depuis.

$$\mathcal{D}_{fix} := \{E(i) : E(i) \neq \emptyset \wedge i \in A_{fix}\} \cup \{E(i) : E(i) \neq \emptyset \wedge hp_i - t_{prec(i),i} < T_x\} \tag{3.7}$$

$$\mathcal{D}_{modif} := \{E(i) : E(i) \neq \emptyset \wedge i \in A_{modif}\} \setminus \mathcal{D}_{fix} \tag{3.8}$$

$$\mathcal{D}_{libre} := (\{E(i), hp_i : i \in A_{libre}\} \cup \{E(i), hp_i : E(i) = \emptyset \wedge i \in A_{fix} \cup A_{modif}\} \cup \{i \in P : ha_i < T_{x-1}\}) \setminus (\mathcal{D}_{fix} \cup \mathcal{D}_{modif}) \tag{3.9}$$

Les stratégies déployées correspondent à différentes valeurs de $EvDec$, H_{fix}, H_{modif}, H_{libre} et $hplan_i$. Pour $hplan_i$ plusieurs calculs peuvent être adoptés, statiques (ex : un délai avant l'heure de ramassage au plus tôt) ou dynamiques (ex : le délai pour le véhicules le plus proche dans le plan prévisionnel). Dans tous nos tests $hplan_i = es_i$ l'heure de prise en charge au plus tôt, d'où une pénalisation des requêtes lointaines.

Dans les tableaux 3.2 et 3.3, sont donnés les paramétrages de chaque stratégie avec plus de détail.

3.6.1.3 Regroupement des demandes en simultané

Afin de simplifier l'algorithme de prise de décision de certaines stratégies manuelles S_2 et S_3, on utilise un pré-traitement qui regroupe arbitrairement des requêtes en simultané.

L'algorithme 7 indique comment se fait se regroupement des requêtes à chaque appel à la planification, en se basant sur le coefficient de proximité des requêtes $R(i,j)$ définie par l'équation (3.10). Les requêtes sont regroupées dans l'ordre croissant de leur coefficient de proximité, avec $Dur_{a,b,c,d}$ représente la durée minimale pour réaliser le chemin a, b, c, d.

$$R(i,j) := \frac{\min\{Dur_{p_i,p_j,d_i,d_j}; Dur_{p_i,p_j,d_j,d_i}\}}{Dur_{p_i,d_i,p_j,d_j}} \tag{3.10}$$

Algorithm 7 Algorithme de regroupement des requêtes

Require: $\alpha \in \mathbb{R}$ (paramètre), $\mathcal{P}_{cur}$ =Ensemble de demandes connues à l'instant T_{cur}
  $\mathcal{L} = \{(i,j) \in \mathcal{P}^2_{cur} : R(i,j) < \alpha\}$
  while $\mathcal{L} \neq \emptyset$ **do**
    $(u,v) = argmin_{(i,j)\in\mathcal{L}}\{R(i,j)\}$
    Créer la séquence (u,v) en simultané qui minimise $R(u,v)$
    $\mathcal{L} = \mathcal{L} \setminus \{(i,j) : \{i,j\} \cap \{u,v\} \neq \emptyset\}$
  end while

L'algorithme donné fonctionne pour des paires de sommets mais pourrait être facilement étendu à des triplets ou plus dans le calcul des coefficients de proximité. Sur les instances testées les triplets et plus sont rares.

Dans nos expérimentations $\alpha = 1$. C'est-à-dire que le simultané est recommandé s'il fait gagner du temps par rapport à une succession des services.

3.6.2 Stratégie basée sur les activités S_1

Le principe général de la stratégie S_1 est une planification sur horizon glissant s'appuyant sur un algorithme d'optimisation.
Le tableau 3.2 décrit le paramétrage de la stratégie S_1.

Paramètre	Valeurs
$hplan_i$	es_i
$EvDec$	$F \in \{15min., 30min., 60min.\} \wedge \{i \in P : T_x < ha_i \wedge hplan_i \leq T_{x+F}\} \neq \emptyset$
H_{fix}	$\{10min., 30min., 60min.\}$
H_{modif}	H_{fix}
H_{libre}	$\{60min., 120min., 240min.\}$

TABLE 3.2 – Paramètres de la stratégie $S1$

Les événements déclencheurs de cette stratégie sont les suivants :

1. La fréquence F de ré-optimisation.
2. Une demande urgente ($hplan_i \leq T_{x+F}$) est apparue récemment ($T_x < ha_i$).

Les annulations des demandes n'apparaissent pas généralement à la dernière minute avant leur prise en charge. De plus, cela ne pose pas de vrai problème au niveau de la planification. Il engendre plutôt une sous-optimalité de la solution. Ainsi, les annulations ne génèrent pas de déclenchement de la planification.
Par la suite l'algorithme d'application de S_1 est décrit pour P_1, puis son adaptation aux autres variantes de problèmes sont données dans les sections suivantes.

3.6.2.1 S_1 appliquée à P_1

Les étapes sont- : mise à jour du plan prévisionnel et préparation des données d'entrée ; prise de décisions ; communication des décisions. Les deux premières étapes relèvent du noyau d'optimisation et sont présentées ici.

Préparation des données et du plan initial Pi La première étape de la méthode est d'annuler les décisions du plan prévu qui sont dans $\mathcal{D}_{libre}$ et $\mathcal{D}_{modif}$ pour avoir un plan initial $\mathcal{P}\rangle$, puis de tâcher de fixer des décisions pour toutes les décisions de $\mathcal{D}_{libre} \cup \mathcal{D}_{modif}$. D'un autre point de vue, il s'agit de planifier en figeant tout ce qui est passé ainsi que les affectations d'équipages aux requêtes de $\mathcal{D}_{fix}$.

Dans le problème P_1, un seul type de véhicule est considéré. En effet, chaque ressource ambulancier ou véhicule n'effectue qu'une seule tournée. Une fois le chauffeur embarqué, l'amalgame entre équipage, véhicule et ambulancier pourra être fait sans ambiguïté.

Les horaires prévus de $\mathcal{D}_{fix}$ et $\mathcal{D}_{modif}$ sont mis à jour par la même procédure qui actualise les temps de passage réels de la formule de récurrence (3.3), à partir du dernier horaire de passage réel connu pour chaque ambulancier.

Il faut noter que les horaires prévus pour des requêtes de $\mathcal{D}_{fix}$ peuvent subir des retards importants qui auront peu de chances d'être réduits, puisque l'affectation aux chauffeurs est figée pour ces requêtes. La seule flexibilité est de changer l'ordre de la séquence de visites pour l'ambulancier. Cette option qui semble peu pertinente dans le cas de contraintes horaires serrées n'est pas envisagée dans notre algorithme d'optimisation.

La liste $\mathcal{L}$ des requêtes à insérer dans le plan initial $\mathcal{P}i$ est construite à partir des ensembles $\mathcal{D}_{modif}$ et $\mathcal{D}_{libre}$.

Construction du nouveau plan prévu Pl^n L'algorithme 8 consiste à ordonner la liste $\mathcal{L}$ des requêtes à insérer, insérer ces requêtes une par une dans l'ordre au chauffeur/véhicule et à la position (de prise en charge et dépose) qui minimise le coût marginal de cette insertion en utilisant la formule (3.2). La première fois, la liste de requêtes est insérée en suivant les dates de début au plus tôt, puis de façon totalement aléatoire. La procédure est appliquée Nit fois, ce paramètre est fixé dans notre implémentation à la racine carrée du nombre de requêtes à insérer.

3.6.2.2 S_1 appliquée à P_2

Dans cette partie, nous considérons le problème simplifié P_1 en lui rajoutant la contrainte de débauche des ambulanciers.

Cette nouvelle contrainte n'impacte pas la replanification effectuée pour P_1, si ce n'est dans l'évaluation du coût d'insertion qui donne à présent des valeurs non nulles pour les fins de journées retardées des ambulanciers.

La règle de fin de journée pour les ambulanciers est qu'ils finissent leur journée si à l'instant courant T_{cur} ils n'ont aucune tâche planifiée pour la suite, sinon leur date de fin de journée prévue est repoussée par tranches d'une heure (valeur de t dans l'équation (3.1)) jusqu'à dépasser l'heure de fin prévue du service qui lui est affecté.

3.6.2.3 S_1 appliquée à P_3

Dans cette partie, nous considérons le problème simplifié en lui rajoutant la contrainte de la prise obligatoire des pauses par les ambulanciers.

Les pauses intégrées ici sont le déjeuner, le dîner et les vingt minutes toutes les six heures. On considère que ces pauses sont prises au dépôt uniquement. Elles sont modélisées comme des requêtes spéciales qui ne peuvent être faites que par un ambulancier donné.

Algorithm 8 Algorithme d'insertion de S_1

Require: Pl^{n-1}, $\mathcal{L}$
Ensure: Nouveau plan prévu Pl^n
  $Nit := \sqrt{|\mathcal{L}|}$
  Préparation des données : $\mathcal{P}i \underset{(3.3)}{\leftarrow} Pl^{n-1}$
  for all $it = 1, \ldots, Nit$ **do**
    $Pl^n(it) \leftarrow \mathcal{P}i$
    if $it = 1$ **then** $\mathcal{L}_{it} \underset{es_i\, croissant}{\leftarrow} \mathcal{L}$
    else $\mathcal{L}_{it} \underset{Ordre\, aleatoire}{\leftarrow} \mathcal{L}$
    end if
    while $r \leftarrow pop(\mathcal{L}_{it}) \neq \varnothing$ **do**
      for all $e \in E$, e qualifié pour r **do**
        for all pos position d'insertion de r dans les trips de e **do**
          Calculer $\Delta Z(e, pos)$
        end for
      end for
      $(e^*, pos^*) := argmin_{e,pos}\{\Delta Z(e, pos)\}$
      Insérer r dans le trip e^* dans $Pl^n(it)$ à la position pos^*
    end while
  end for
  $Pl^n := argmin_{it}\{Z(Pl^n(it))\}$

Les chauffeurs sont disponibles toute la journée et donc pause déjeuner et dîner sont à planifier.

Pour les pauses de vingt minutes après les six heures, l'insertion n'est faite qu'en cas de violation de la contrainte. Elle est insérée au dernier moment, avant de dépasser les six heures de travail continu. En pratique, cette pause ne s'applique pas souvent.

3.6.2.4 S_1 appliquée à P_4

Dans P_4, le problème P_1 est enrichi de différents types de véhicules. Les ambulanciers sont amenés à changer de véhicules pour satisfaire les besoins des usagers. De plus, des véhicules nécessitent plusieurs ambulanciers et des usagers peuvent être servis par différents types de véhicules.

Comme mentionné auparavant dans 3.6.1.1, le plan prévu est représenté comme un ensemble de trips, chacun est constitué de points de passage (ramassage, dépose, dépôt, pause). Chaque trip est affecté à un équipage et un véhicule. Une ressource (chauffeur ou véhicule) peut être attribuée à plusieurs trips. Ces derniers sont toutes des tournées (qui commencent et qui finissent au dépôt) à l'exception du premier trip qui commence par le dernier endroit déjà visité.

Dans ce contexte, l'évaluation du coût marginal d'insertion d'une requête dans un trip est plus compliqué à effectuer car il faut propager les répercussions pour tous les trips

des ambulanciers impliqués.

Pour simplifier cette procédure, l'équipage et le véhicule d'un trip ne sont pas remis en cause pour une insertion au milieu de ce même trip.

Dans l'algorithme 8, au lieu de considérer les chauffeurs individuellement ce sont les équipages et les véhicules de chaque trip qui sont à prendre en compte. De plus, pour des ambulanciers et véhicules disponibles au même dépôt, les équipages possibles sont également constitués et évalués pour le calcul d'un coût marginal $\Delta Z(e, pos)$ (voir Algorithme 9).

3.6.2.5 S_1 appliquée à DMTP

DMTP combine les propriétés de P_1 à P_4.

Puisque les chauffeurs ne sont pas toujours disponibles, l'algorithme servant à résoudre P_3 est peu modifié à la différence que les requêtes pauses déjeuner et dîner sont initialement insérées et jamais retirées des trips du plan prévu, car on n'avance jamais la date de débauche prévue initialement. En revanche, lorsque la date de débauche est repoussée les pauses nécessaires sont ajoutées dans le plan de travail de l'ambulancier. Ce qui implique une replanification d'une partie des activités des ambulanciers concernés.

3.6.3 Stratégie basée sur les ressources puis les activités S_2

La stratégie S_2 reflète un régulateur qui n'anticipe jamais. Une fois que l'équipage/véhicule dépose le dernier patient (finit sa dernière mission), ils sont chargés d'une nouvelle mission. Si aucune mission n'est affectée à un ambulancier libre, ce dernier est reconsidéré (événement déclencheur) toutes les $Wait = 15$ minutes pour une nouvelle affectation.

Les paramètres de S_2 s'expriment dans le tableau 3.3.

Paramètre	Valeurs
$hplan_i$	es_i
$EvDec$	$\{j \in P : T_{cur} = hr_j \vee (\frac{T_{cur} - hr_j}{Wait} \in \mathbb{N} \wedge \{i \in A_{libre} : E(i) = E(j)\} = \emptyset)\} \neq \emptyset$
H_{fix}	$\{10min., 30min., 60min.\}$
H_{modif}	H_{fix}
H_{libre}	$\{60min., 120min., 240min.\}$

TABLE 3.3 – Paramètres de la stratégie $S2$

3.6.3.1 S_2 appliquée à P_1

La procédure de mise à jour des horaires de passage de la section 3.6.2.1 est appliquée de la même manière au cours de la journée comme à l'appel de la stratégie de replanification.

L'idée première de la replanification consiste à trouver la prochaine mission de l'ambulancier qui vient de se libérer. Mais un problème plus général est considéré, celui d'affecter la prochaine mission à tous les ambulanciers libres. Pour cela un problème d'affectation plus général encore est considéré, en incluant l'ensemble des ambulanciers pour trouver leurs prochaines missions.

Au final ne seront retenues que les décisions pour les ambulanciers disponibles à T_{cur} la date courante.

Ce problème se modélise par le programme linéaire binaire (3.11)-(3.14), où $X_{i,e}$ décide de l'affectation de l'ambulancier e à la mission i et $\Delta_{i,e}$ le coût marginal d'insérer i à la fin de journée de l'ambulancier e. Rappelons ici, que la fonction objectif minimise le nombre de requêtes non couvertes. Le coût marginal d'insertion est donc souvent négatif et incite à fixer des variables de décision à 1.

$$\min \sum_{i \in P} \sum_{e \in E} X_{i,e} \Delta_{i,e} \tag{3.11}$$

$$\sum_{i \in P} X_{i,e} \leq 1 \ \forall e \in E \tag{3.12}$$

$$\sum_{e \in E} X_{i,e} \leq 1 \ \forall i \in P \tag{3.13}$$

$$X_{i,e} \in \{0,1\} \ \forall i \in P \ \forall e \in E \tag{3.14}$$

Afin de garantir de servir des requêtes en simultané, une mission peut ici représenter une séquence de services. Les missions en simultané sont précalculées par la procédure de la section 3.6.1.3.

3.6.3.2 S_2 appliquée à P_2

La même approche pour S_1 (voir section 3.6.2.2) est appliquée pour S_2 avec pour seule modification là aussi la valeur des coûts marginaux d'insertion des missions dans les trips des ambulanciers.

3.6.3.3 S_2 appliquée à P_3

Puisque les insertions de missions se font en fin de journée, la nécessité d'insérer ou pas une pause est traitée directement au moment du calcul du coût marginal d'insertion.

De plus, à la fin de chaque re-planification, une pause est affectée aux chauffeurs libres qui n'ont aucune demande de prévue et qui sont aptes à prendre cette pause.

3.6.4 Stratégie basée sur les activités puis les ressources S_3

La stratégie S_3 modélise un régulateur qui anticipe, et qui insère une par une les demandes à la fin de la journée des ambulanciers. Cette stratégie peut être vue comme une

approche pour traiter le problème P_1, dans laquelle les équipages sont traités un par un séparément.

Ainsi, le paramétrage de la stratégie S_3 est celui du tableau 3.2.

3.6.4.1 S_3 appliquée à P_1

Ici encore les mises à jour du plan prévu et la préparation en annulant les décisions de $\mathcal{D}_{modif}$ et $\mathcal{D}_{libre}$ sont identiques aux stratégies précédentes.

L'algorithme général est de trier les demandes par l'heure au début au plus tôt de l'heure de prise en charge. Ensuite, pour chaque demande dans l'ordre, insérer la demande à la fin de la tournée du chauffeur e qui minimise le coût marginal d'insertion. Pour l'ambulancier e ainsi déterminé, son plan de travail est rempli par la fin en prenant les demandes suivantes dans l'ordre établi tant qu'aucun retard n'est subi ; c'est à dire l'heure de passage prévue de chaque service hp_i ne doit pas dépasser son heure de passage au plus tard ls_i (sans réaliser de retards). Cette procédure est décrite par l'algorithme 10.

3.6.4.2 S_3 appliquée à P_2

Le traitement est identique à celui de S_1 décrit en section 3.6.4.2.

3.6.4.3 S_3 appliquée à P_3

Comme pour la stratégie S_2, les requêtes sont insérées à la fin de la journée en cours de planification d'un ambulancier. Donc la question des pauses est traitée directement au moment d'évaluer le coût marginal d'une insertion.

Comme dans S_2, à la fin de chaque replanification, la pause est affectée aux chauffeurs à la fin de la journée qui n'ont pas encore pris de pause et qui n'ont pas de pause affectée.

A l'inverse de S_1, les pauses peuvent être retirées du plan prévisionnel si elles sont dans $\mathcal{D}_{modif} \cup \mathcal{D}_{libre}$. Elles seront éventuellement réinsérées à la prochaine replanification.

3.7 Expérimentation numérique

Dans cette section nous présentons la plateforme de tests développée puis les résultats d'un plan d'expérience complet qui permet de montrer l'intérêt de la stratégie S_1, d'en estimer les meilleurs paramétrages et enfin de mesurer l'apport du plan de la veille pour le lendemain. Le tout dans différents contextes de taille et de dynamicité du système.

3.7.1 Plateforme de tests

La plateforme de tests comprend un module de simulation à événements discrets pour évaluer les stratégies en moyenne et dans différents contextes, un générateur de données

sous incertitude à partir de données historiques et l'outil de planification statique du chapitre précédent qui permet de calculer un plan de départ pour chaque jour et de calculer une borne à confronter aux stratégies dynamiques.

Le processus de simulation est décrit par la figure 3.5. Au début de la journée, le premier plan prévu est obtenu en utilisant l'algorithme "ALNS" (décrit dans le chapitre précédent). ALNS est exécuté sur un scénario d'une journée complète généré sur la base de demandes connues (données de la veille) et d'autres demandes aléatoires décrites dans la section 3.7.2. Les nouvelles arrivées, le changement d'horaires ou l'annulation d'une demande sont reçus en continu durant la journée. La liste des demandes est mise à jour après chaque nouvelle information reçue. Les événements déclencheurs d'une nouvelle re-planification sont : (i) toutes les F minutes (stratégie S_1 et S_3) ; (ii) à l'arrivée d'un équipage sur le point de dépose du dernier client affecté ; (iii) à l'arrivée d'une nouvelle demande urgente, d'un changement d'horaire d'une requête urgente ou d'une annulation d'une requête urgente déjà insérée dans le plan prévu.

Dans le cas d'arrivée de nouvelles informations relatives à une demande, cette dernière devient urgente si :

— Il est prévu d'envoyer l'équipage vers le service de sa prise en charge avant l'heure du prochain événement déclencheur connu. Dans le cas où la demande est déjà insérée dans le plan prévu.

— L'heure de sortie au plus tard, depuis le plus proche dépôt sans arriver en retard au point de prise en charge de la demande, est avant le prochain événement déclencheur. Dans le cas où la demande n'est pas insérée dans le plan prévu.

Pour chaque demande de re-planification pour la mise à jour du plan prévu, les ensembles de requêtes (A_{fix}, A_{modif} et A_{libre}) sont mises en place à partir du plan prévu et l'ensemble des demandes connues. Les équipages/véhicules quittent réellement le service à l'heure de sortie prévue (Sortie du service), et ils arriveront à destination à un moment qui peut être différent de ce qui était prévu initialement car le temps de parcours réel est différent de ce qui est prévu.

La durée réelle d'un trajet est connue lors de sa réalisation, si elle se réalise avant la durée prévue. Sinon, une fois la date prévue d'arrivée dépassée, on utilise comme nouvelle durée prévue la durée réelle. C'est un biais introduit dans la simulation qui évalue les stratégies de régulation pour en simplifier la prédiction des durées de transport restantes. L'implémentation commerciale s'appuie sur une prévision de la date d'arrivée avec les outils de surveillance du trafic temps réel, cette prévision peut être espérée très proche de la date réelle d'arrivée.

Une fois l'heure d'arrivée prévue d'un équipage sur une mission (emplacement de prise en charge ou dépose d'une demande) est atteinte alors que l'équipage n'est toujours pas arrivé à destination, l'heure d'arrivée réelle devient connue et les horaires prévus du plan prévu sont remis à jour.

à l'arrivée réelle d'un équipage sur le lieu de la mission, les horaires prévus du plan prévu ainsi que la liste des demandes connues non encore servies sont remis à jour.

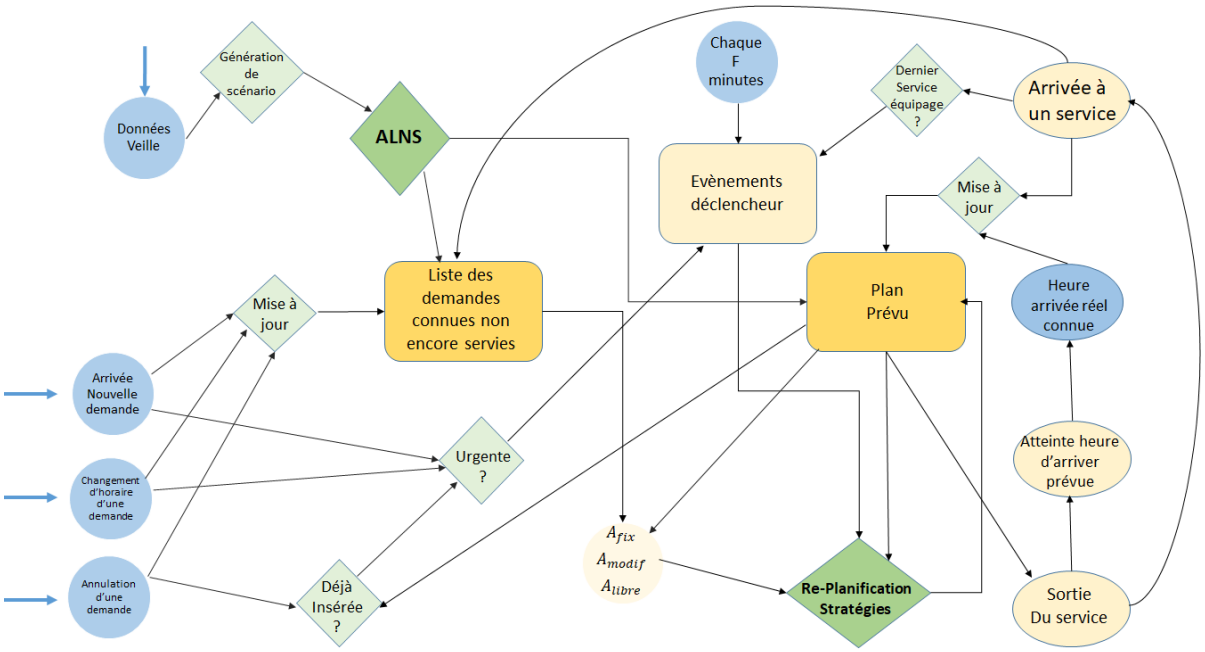

FIGURE 3.5 – Description de l'outil de simulation

L'algorithme 11, montre la procédure générale utilisée dans la simulation. La première étape consiste à extraire l'ensemble des demandes connues la veille (V) avec la fonction $REST$. L'extraction est réalisée à partir d'une donnée historique I et un pourcentage pf.

La deuxième étape consiste à générer un scénario, (i.e les aléas d'une journée complète) avec la fonction $GENSC(V, I)$, en gardant l'ensemble V comme ensemble de demandes connues la veille. L'étape suivante revient à appliquer l'algorithme de la veille pour le lendemain "ALNS" sur le scénario généré (G). Les décisions admises dans le planning de la veille (PV) sont limitées à celles qui concernent les demandes connues la veille V et dont l'heure de passage de leurs prises en charge prévues sont avant HV. Ces affectations ne seront jamais remises à jour par la suite et seront toujours dans $\mathcal{D}_{fix}$.

Ensuite, pour chaque réplication j, un nouveau scénario G_j est généré en gardant les mêmes demandes connues la veille. La stratégie ζ est appliquée sur le scénario généré G_j en prenant PV comme le plan prévu initial. Les critère d'optimisation de la fonction objectif (présentés dans la fonction 3.2) sont stockés dans KPI_j.

Finalement, après les $|J|$ réplications un intervalle de confiance de 95% de chaque KPI est retourné.

Dans la sous-section suivante, nous allons présenter les deux fonctions utilisées pour la génération des données :

— $V = REST(I)$ pour l'extraction des demandes connues la veille à partir d'une journée historique (I) ;

— $G = GENSC(V, I)$ pour la génération d'un nouveau scénario, à partir des demandes de la veille et de la journée historique I .

3.7.2 Génération de données

Les données historiques, ont été récupérées depuis la base de données des clients sur des journées réellement réalisées.

Les informations dont on dispose dans les données historiques d'une journée de régulation (I) concernent :

— les demandes traitées durant la journée (point de prise en charge, point de dépôt, fenêtres de temps à la prise en charge et dépose, temps de service à la prise en charge et dépose, types de véhicules, nombre de personnes à transporter,..) ;

— les véhicules disponibles durant la journée : type et dépôt de localisation ;

— les ambulanciers disponibles durant la journée : dépôt de rattachement, heure de début et heure de fin de travail prévues et diplômes en possession.

Le tableau 3.4 regroupe l'ensemble des informations disposées qui concernent les données utilisées dans nos expérimentations (nom, nombre de demandes, nombre de chauffeurs, nombre de véhicules et nombre de dépôts).

Nom	nb demandes	nb chauffeurs	nb véhicules	nb bases
100-1	128	20	16	10
200-1	189	27	22	6
300-1	312	42	34	25
1000-1	1628	267	200	61
1000-2	1178	196	165	61

TABLE 3.4 – informations sur les instances utilisées

Les informations majeures manquantes dans ces données historiques sont l'ensemble des demandes connues la veille ainsi que l'heure d'apparition des demandes dynamiques.

Les demandes générées dans nos expérimentations ne tiennent pas compte des annulations des horaires des demandes. Ces dernières sont très rares durant une journée de régulation.

La première étape de la génération de données consiste à sélectionner parmi l'ensemble de demandes de la journée les demandes connues la veille.

3.7.2.1 Demande de la veille ($REST(I)$)

Soient les deux paramètres :

— pf le pourcentage souhaité des demandes connues la veille.

— $In_1 = [0, \Delta t[, In_2 = [\Delta t, 2\Delta t[, ..., In_n = [(n-1)\Delta t, n\Delta t[$ les intervalles de temps d'une journée avec n le nombre des intervalles et $\Delta t = \frac{1440}{n}$. Dans nos expérimentations $n = 7$.

— $P(In)$ la probabilité qu'une demande r dont l'heure de début au plus tôt de prise en charge es_{p_r} appartient à l'intervalle In soit inconnue la veille. Dans nos expérimentations $P(In_1) = 0,001$, $P(In_2) = 0,004$, $P(In_2) = 0,005$, $P(In_4) = 0,04$, $P(In_5) = 0,2$, $P(In_6) = 0,3$ et $P(In_7) = 0,45$.

La procédure est la suivante : prendre une demande r aléatoirement d'une façon uniforme parmi l'ensemble de demandes non encore sélectionnées, la demande r est connue la veille avec la probabilité $1 - P(In)$, avec $es_{p_r} \in In$. Répéter la procédure jusqu'à atteindre le pourcentage pf.

La deuxième étape de la génération de données est de générer des nouvelles demandes dynamiquement durant la journée.

3.7.2.2 Génération de nouvelles demandes

L'idée de la génération des nouvelles demandes est de piocher dans l'historique en respectant les distributions statistiques par tranches horaires. Les attributs des requêtes sont conservés intacts car on suppose qu'ils sont corrélés à leur tranche horaire. Cependant leur heure de début est modifiée et suit une distribution de Poisson propre à chaque tranche horaire. En général ce sont les heures d'appel qui suivent une loi de Poisson mais n'ayant pas cette donnée dans l'historique, on utilise l'heure de début à la place. Cela revient à supposer une relation simple entre heure d'appel et heure de début de service.

Ainsi, la première étape de la génération de nouvelles demandes est de calculer la moyenne entre deux prises en charge au plus tôt des demandes non sélectionnées par la procédure précédente.

Les valeurs des paramètres utilisés dans les définitions suivantes se trouvent dans le tableau 3.5.

Soient :

— N_{In} le nombre de demandes non connues la veille dont leur heure de début au plus tôt de prise en charge appartient à l'intervalle In ;

— d_{In} la taille de l'intervalle In ;

— $\lambda_{In} = \frac{d_{In}}{N_{In}}$ la moyenne entre deux prises en charge au plus tôt dans l'intervalle In.

La deuxième étape est de générer les heures de prise en charge au plus tôt des nouvelles arrivées sous l'hypothèse qu'elles suivent un processus de Poisson avec l'équation suivante

$$tp_{i+1} = tp_i + EXP(\lambda_{In_z}) \text{ tant que } tp_{i+1} < z\Delta t,$$

où tp_{i+1} représente l'heure de prise en charge de la nouvelle demande $i + 1$, en tenant compte de l'heure de prise en charge au plus tôt de la dernière demande générée i.

Au début, z est initialisé à 1, $tp_i = 0$. Une fois $tp_{i+1} > z\Delta t$, z est incrémenté de 1 et $tp_i = (z-1)\Delta t$.

Une fois que l'heure au plus tôt de prise en charge de la nouvelle demande est générée tp_{i+1}, la génération des autres informations de la demande constitue la prochaine étape.

Une requête r est choisie aléatoirement parmi l'ensemble des demandes initiales dont l'heure de début au plus tôt appartient à un intervalle $[tp_{i+1} - \Delta, tp_{i+1} + \Delta]$, avec Δ un paramètre. L'ensemble des informations de la requête i sont attribuées à la nouvelle demande générée, sauf pour les fenêtres de temps de prise en charge et de dépose qui sont décalées par rapport à tp_{i+1}.

Les durées réelles de transport $tr_{i,j}$ entre chaque paire de sommets (i,j) sont générées à partir du temps de trajet prévu $tp_{i,j}$.

$$tr_{i,j} = K_{i,j}.t_{i,j}$$

Avec $K_{i,j}$ un nombre aléatoire tiré de manière uniforme entre $[\Delta_1, \Delta_2]$.

L'heure d'apparition de la nouvelle demande est ensuite générée $hap_i = tp_{i+1} - max_{d \in Depot} tr_{d,P_r} - \omega$, avec $\omega =$ est un nombre aléatoire tiré de manière uniforme entre $[\Delta_3, \Delta_4]$

Paramètre	Δ	Δ_1	Δ_2	Δ_3	Δ_4	iter
	120	0,95	1,2	5	360	$\sqrt{\lvert R \rvert}$); $R =$ L'ensemble de requêtes à insérer

TABLE 3.5 – Valeurs des paramètres

3.7.3 Résultats numériques

Les résultats numériques commencent par déterminer les bons paramétrages des stratégies en fonction du contexte d'exécution. Ensuite la performance des meilleures stratégies est comparée à des bornes basées sur la connaissance d'une information parfaite. Enfin l'impact de plan calculé la veille avec une proportion de demandes connues est évalué pour différents contextes d'utilisation.

3.7.3.1 Stratégies et paramétrages

Nous avons généré pour chaque instance historique, 3 différents pourcentages de requêtes connues la veille : $Pf = 30\%; Pf = 50\%$ ou $Pf = 70\%$. Pour chaque valeur de Pf, nous avons généré 30 journées différentes (réplications) grâce à la génération de données. Les stratégies sont exécutées sur 22 paramètres différents avec toutes les combinaisons de :

- $H_{fix} = 10, 30, 60$
- $F = 15, 30, 60$
- $H_{libre} = 60, 120, 240$

telles que $H_{libre} \leq H_{fix} + F$. L'ensemble des expérimentations présentées dans cette partie, sont exécutées avec $HV = 0$ (i.e le plan prévu est initialisé comme étant vide).

Le tableau 3.6 montre l'efficacité de la stratégie S_1 par rapport aux stratégies S_2 et S_3 sur les 3 problèmes(P_1, P_2 et P_3).

Chaque ligne du tableau est calculée sur la base de 1980 lancements des stratégies(30 réplications $\times$ 3 pourcentages Pf $\times$ 22 paramètres) .

La première colonne représente le nom de l'instance historique, la deuxième colonne montre sur quel problème les stratégies sont exécutées. Les colonnes str1, str2 et str3 montrent la stratégie exécutée. La sous-colonne $\%B_t$ représente le pourcentage où la stratégie en question est meilleure par rapport aux deux autres. La colonne AgG montre le gap moyen entre la stratégie en question j et la stratégie 1, $AgG_j = \frac{1}{1980}\sum_{pf,rep,pm} Gap_j$tel que $Gap^j_{pf,rep,pm} = \frac{Z_{pf,rep,pm}(str_j) - Z_{pf,rep,pm}(str_1)}{Z_{pf,rep,pm}(str_1)}$ avec $Z_{pf,rep,pm}(str_j)$ représente la valeur de la fonction objectif en appliquant la stratégie j (str_j) avec le paramétrage pm sur le jeu de données généré avec le pourcentage de dynamicité pf pour la réplication rep. La sous colonne $Gmax$ (resp. $Gmin$) représente le gap maximum (resp. minimum) entre la stratégie en question j et la stratégie 1, $Gmax_j = \max_{pf,rep,pm} Gap^j_{pf,rep,pm}$ (resp.$Gmin_j = \min_{pf,rep,pm} Gap^j_{pf,rep,pm}$). La colonne P_{GMax} (resp. P_{GMin}) montre le paramétrage où $Gmax$ (resp. $Gmin$) est atteint, le paramétrage de la colonne $Gmax$ ou $Gmin$ est présenté avec $((H_{fix}, F, H_{libre}))$.

Les résultats affichés dans le tableau 3.6 montrent que la stratégie 1 domine à $93,13\%$, en moyenne, les deux autres stratégies sur tous les problèmes et paramètres. Au meilleur des cas, la solution de la stratégie 1 est meilleure de 1200%, et au pire cas la solution de la stratégie 1 est plus mauvaise de 40% que la stratégie 2 et 26% pour la stratégie 3. En revanche, en moyenne, la stratégie 1 est toujours meilleure que les deux autres stratégies. Généralement la stratégie 1 est performante lorsque $H_{libre} = 240$ et beaucoup moins performante lorsque $H_{libre} = 60$.

L'efficacité de la stratégie 1 sur les problèmes P_1, P_2 et P_3, nous permet de déduire que la stratégie 1 sera aussi performante sur les problèmes P_4 et P_5.

Les tableaux 3.7, 3.8 et 3.9 montrent l'influence des paramètres H_{fix}, F et H_{libre} sur la qualité des solutions. Les résultats sont calculés en utilisant la stratégie 1 sur le problème 5. Chaque ligne des tableaux est calculée sur la base de 5 instances $\times$ 30 réplications $\times$ 22 paramètres.

Le tableau 3.7 montre l'influence du paramètre H_{libre} et le degré de dynamisme pf sur la qualité de la solution, chaque colonne $Z(H_{libre} = hl_1) < Z(H_{libre} = hl_2)$ affiche le pourcentage de fois où la valeur de la fonction objectif avec le paramètre $H_{libre} = hl_1$ est meilleur que la valeur de la fonction objectif avec le paramètre $H_{libre} = hl_2$.

Les résultats montrent que plus le paramètre H_{libre} est grand plus la solution est meilleure. La valeur de la solution s'améliore encore plus si le pourcentage des requêtes connues la veille est plus grand.

Le tableau 3.8 montre l'influence du paramètre H_{fix} et le degré de dynamisme pf sur

la qualité de la solution lorsque le paramètre H_{libre} est fixé, chaque colonne $Z(H_{fix} = hf_1) < Z(H_{hfix} = hf_2)$ affiche le pourcentage de fois où la valeur de la fonction objectif avec le paramètre $H_{fix} = hf_1$ est meilleure que la valeur de la fonction objectif avec le paramètre $H_{fix} = hf_2$ en fixant le paramètre H_{libre}.

Les résultats montrent que plus le paramètre H_{fix} est petit plus la solution est meilleure, la valeur de la solution s'améliore encore plus si le pourcentage des requêtes connues la veille est plus grand.

Le tableau 3.9 montre l'influence du paramètre F sur la qualité de la solution sur chaque paramètre de dynamisme pf, chaque colonne $Z(F = f_1) < Z(F = f_2)$ affiche le pourcentage de fois où la valeur de la fonction objectif avec le paramètre $H_{libre} = hl_1$ est meilleure que la valeur de la fonction objectif en fixant les paramètres H_{libre} et H_{fix}.

Les résultats montrent que plus le paramètre F est petit plus la solution est meilleure pour tout paramètre H_{fix} et H_{libre} fixés.

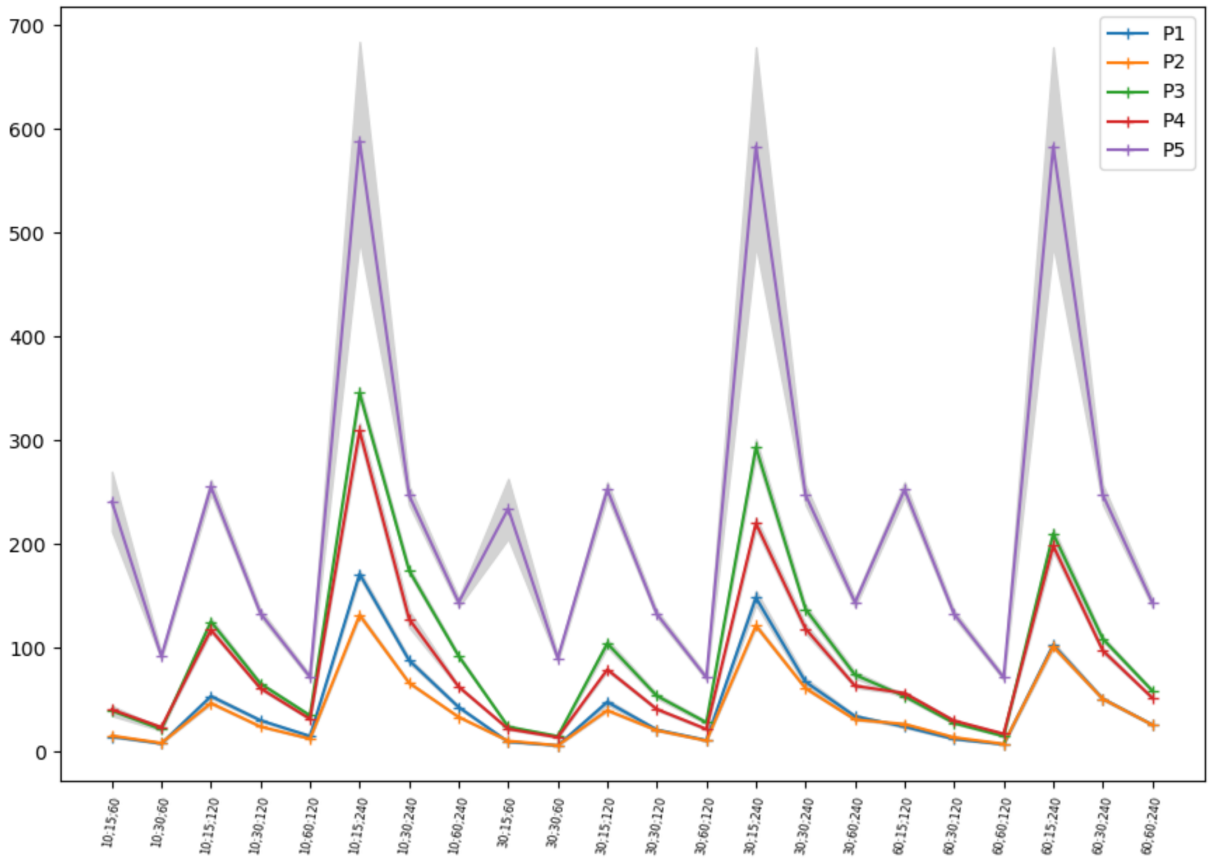

FIGURE 3.6 – Temps de calculs

La figure 3.6 affiche l'évolution du temps de calcul par rapport aux 22 différents paramétrages sur chaque problème P_j. Le graphe affiche le temps de calcul moyen des 30 réplications de la stratégie 1 en utilisant l'instance historique $1000 - 1$. La partie grisée du graphe montre l'intervalle pour 95% des 30 réplications.

On voit que la résolution du problème P_5 prend plus de temps de calcul que P_3,P_4,P_1 et P_2. Une résolution du problème général P_5 sur une journée entière de 1650 requêtes (pire cas) avec le paramétrage (10 ;15 ;240), peut atteindre 11 minutes de calcul. La figure montre aussi que plus le paramètre H_{libre} augmente et H_{fix}, F diminue, le temps de calcul devient plus important.

Dans les sous sections suivantes, nous nous concentrons sur la résolution du problème général P_5 en utilisant la stratégie 1 avec 4 paramétrages retenus, à savoir :

— Aspect dynamique $(10; 15; 60)$

— Meilleur paramétrage $(10; 15; 240)$

— Utilisation réelle $(30; 30; 120)$

— Paramétrage moyen $(60; 15; 240)$

3.7.3.2 Bornes avec information parfaite

Dans cette sous-section, nous comparons la solution de la stratégie 1 avec 2 bornes en cas d'information parfaite, *i.e*, tous les incertitudes de la journée sont connues dès le début de la journée. La première borne constitue l'utilisation du plan de la veille pour le lendemain en connaissant toutes les données de la journée. La deuxième borne est l'utilisation de la stratégie 1 dans le cas de l'information parfaite.

Afin de pouvoir comparer, nous devons changer la fonction objectif (2.1) du plan de la veille pour le lendemain par la fonction objectif (3.2) utilisée en temps réel. En effet, le plan de la veille pour le lendemain ne tient pas compte des retards à l'arrivée d'un ambulancier sur un service. Pour remédier à ce problème et pour pouvoir comparer entre toutes les solutions, le jeu de données de la planification en statique est changé par rapport à la solution de la stratégie 1. Le changement s'effectue sur l'heure au plus tard de passage à chaque service. Pour chaque requête r, et pour chaque service ser_r d'origine p_r ou de destination d_r. hr_{ser_r} représente l'heure de passage réel au service ser_r dans la solution de la stratégie 1, la fenêtre de temps $[es_{ser_r}, ls_{ser_r}]$ est mise à jour avec $[es_{ser_r}, ls'_{ser_r}]$ tel que, $ls'_{ser_r} = \max\{ls_{ser_r}; hr_{ser_r}\}$.

Dans ses parties d'expérimentation, le paramétrage de l'algorithme de la planification de la veille (ALNS) étudié dans le chapitre précédent est le même que celui utilisé dans 2.7 avec les mêmes poids utilisés dans 2.6.

Les tableaux 3.10, 3.11 et 3.12 montrent respectivement la solution de la stratégie 1, la borne en utilisant le planning de la veille pour le lendemain avec information parfaite et la borne en utilisant la stratégie 1 avec information parfaite sans utilisation du plan prévu de la veille (i.e $HV = 0$). Chaque ligne des tableaux est calculée sur la base d'une réplication avec le degré de dynamisme pf et le paramétrage pm, tel que :

— $pm = 1$ représente le paramétrage à l'aspect dynamique ($H_{fix} = 10; F = 15; H_{libre} = 60$)

— $pm = 2$ représente le meilleur paramétrage ($H_{fix} = 10; F = 15; H_{libre} = 240$)

— $pm = 3$ représente le paramétrage à utilisation réelle ($H_{fix} = 30; F = 30; H_{libre} = 120$)

— $pm = 4$ représente le paramétrage moyen ($H_{fix} = 60; F = 15; H_{libre} = 240$)

Chaque colonne des tableaux représente le jeu de données historiques utilisé. La sous-colonne *V.Ob* affiche la valeur de la fonction objectif et *U.R* représente le nombre de requêtes non servies.

Afin de bien comparer les trois solutions, les tableaux 3.14 et 3.13 représentent respectivement l'écart entre la solution de la stratégie 1 et la solution de la veille pour le lendemain avec information parfaite, et l'écart entre la solution de la stratégie 1 et la solution de la stratégie 1 avec information parfaite tels que :

$$Gap(v_a, v_b) = \frac{v_a - v_b}{v_b}$$

.

Le tableau 3.13 montre que le fait de connaitre à l'avance toutes les informations améliore toujours la valeur de la fonction objectif entre 9% et 62%, et de 0% à 100% de requêtes non servies en moins. Le tableau montre aussi que plus le pourcentage de requêtes connu la veille plus l'écart entre les deux solutions diminue.

Le tableau 3.14 montre que le fait de connaître à l'avance toutes les informations et d'utiliser un algorithme d'optimisation plus performant, améliore encore plus la valeur de la fonction objectif entre 33% et 98%, et de 0% à 100% de requêtes non servies en moins. L'écart entre la solution obtenue avec le planning de la veille (VPL) et la stratégie 1 avec information parfaite laisse penser que la stratégie 1 peut être améliorée. En effet, la stratégie 1 est un algorithme glouton comparé à l'algorithme ALNS.

3.7.3.3 Intégration du plan de la veille pour le lendemain dans la planification temps réel

Dans cette partie nous étudions l'impact du plan de la veille pour le lendemain sur les décisions prises par la stratégie dynamique (stratégie 1).

Dans ces parties d'expérimentation, le paramétrage de l'algorithme de la planification de la veille (ALNS) étudié dans le chapitre précédent est le même que celui utilisé dans 2.7 avec les mêmes poids utilisés dans 2.6.

Le premier test que nous avons effectué est d'étudier l'impact de l'heure limite HV dans les décisions de temps réel. Pour ce faire, nous avons comparé trois valeurs de HV différentes :

— $HV = 0$ (Sans utilisation du planning de la veille pour le lendemain) ;

— $HV = 12h$, fixer à l'avance toutes les décisions des requêtes connues la veille dont l'heure de dépose prévue est avant midi ;

— $HV = 18h$, fixer à l'avance toutes les décisions des requêtes connues la veille dont l'heure de dépose prévue est avant $18h$.

Dans cette expérimentation, la génération du scénario du plan de la veille $GENSC$ est limitée aux requêtes connues la veille sans générer de nouvelles demandes.

Comparaison entre HV=0h, 12h et 18h Les tableaux 3.15 et 3.16 montrent l'impact de l'utilisation du planning de la veille pour le lendemain avec des limitations ($HV = 0h; 12h$ et $18h$) dans la planification en temps réel. Chaque ligne du tableau est calculée sur la base de $J = 30$ réplications. Les trois premières colonnes représentent respectivement : Instance historique I, pourcentage de demandes connues la veille pf et le paramétrage pm. Les trois colonnes suivantes représentent respectivement : solution avec $Hv = 0h$, $Hv = 12h$ et $Hv = 18h$. La sous colonne $V.Ob$ représente la moyenne de l'intervalle de confiance de la fonction objectif et $U.R$ fait référence à la moyenne de l'intervalle de confiance du nombre de requêtes non servies. Les deux dernières colonnes $Gap_{a,b}$ affichent le gap moyen entre la valeur de la solution objectif de $HV = a$ et $HV = b$, tel que $Gap_{a,b}^{I,pm} = \frac{1}{4}\sum_{pf} \frac{V.Ob_{HV=a}^{I,pm,pf} - V.Ob_{HV=b}^{I,pm,pf}}{V.Ob_{HV=b}^{I,pm,pf}}$.

Les résultats montrent que l'utilisation du planning de la veille pour le lendemain n'est pas bénéfique lorsque le pourcentage de requêtes connues la veille pf égal à 30% avec une baisse de 2% au pire cas et de 1% au meilleur cas. Par contre, si le pourcentage de nombre de requêtes connues la veille dépasse les 50%, l'amélioration de la valeur de la fonction objectif est en moyenne assurée. L'amélioration peut atteindre jusqu'à 8% de la valeur de la fonction objectif.

L'utilisation du planning de la veille pour le lendemain en fixant les décisions jusqu'à $HV = 18h$ n'est pas toujours bénéfique par rapport à $HV = 12h$. La différence devient plus importante lorsque le pourcentage du nombre de requêtes connues la veille diminue et que les deux valeurs objectifs ce rapprochent dans le cas contraire.

Dans tous les prochains tests, la valeur HV sera toujours fixée à $12h$.

La prochaine expérimentation traite la différence entre l'utilisation du planning de la veille pour un scénario d'une journée complète et seulement sur les requêtes connues la veille.

Comparaison entre l'utilisation du planning de la veille avec un scénario complet et seulement avec les données connues la veille Chaque ligne des tableaux 3.17 et 3.18 est calculée sur la base de ($J = 30$) réplications. Les colonnes I, pf et pm représentent respectivement l'instance, le pourcentage de demandes connues la veille et le paramétrage de la stratégie 1. Les colonnes $VPL(Scenario)$ et $VPL(Veille)$ affichent la moyenne de l'intervalle de confiance des 30 réplications, dans le cas de la génération d'un scénario complet pour $VPL(Scenario)$ et $VPL(Veille)$ dans le cas de la génération d'un scénario avec seulement les requêtes de la veille. Les sous colonnes $V.Ob$ et

$U.R$ affichent la moyenne de l'intervalle de confiance respectivement de la fonction objectif et du nombre de requêtes non servies. La dernière colonne $Gaps_{VPL(Veille),VPL(Scenario)}$ représente le gap de la valeur objectif ($Gap_{V.Ob}$) , du nombre de requêtes non servies et la moyenne de gap de la valeur objectif pour chaque paramétrage pf entre la solution $VPL(Veille)$ et $VPL(Scenario)$.

Les résultats montrent que la solution obtenue en utilisant un scénario complet est toujours (pour tous les paramètres pf , pm et I) meilleure que la solution obtenue en n'utilisant que les requêtes de la veille. L'amélioration est de 3% au pire des cas et de 14% au meilleur cas.

Le dernier test sert à étudier l'impact du planning de la veille si le nombre de chauffeurs utilisés est minimisé. Pour se faire, nous avons ajouté dans la fonction objectif du planning de la veille pour le lendemain (voir section 2.3.5), un autre critère à minimiser qui est le nombre de chauffeurs utilisés. Dans ce test, le poids de ce nouveau critère est fixé à 50 dans les expérimentations.

Minimisation du nombre de chauffeurs utilisés dans le planning de la veille Chaque ligne des tableaux 3.19 et 3.20 est calculée sur la base de ($J = 30$) réplications. Les colonnes I, pf et pm représentent respectivement l'instance, le pourcentage de demandes connues la veille et le paramétrage de la stratégie 1. Les colonnes $AvecMinimisation$ et $SansMinimisation$ affichent la moyenne de l'intervalle de confiance des 30 réplications, dans le cas de minimisation du nombre de chauffeurs utilisé ($AvecMinimisation$) et sans minimisation du nombre des chauffeurs $SansMinimisation$ dans le planning de la veille. Les sous colonnes $V.Ob$ et $U.R$ affichent la moyenne de l'intervalle de confiance respectivement de la fonction objectif et du nombre de requêtes non servies. La dernière colonne $Gaps_{SansMin,AvecMin}$ représente le gap de la valeur objectif ($Gap_{V.Ob}$), du nombre de requêtes non servies($Gap_{U.R}$) et la moyenne de gap de la valeur objectif pour chaque paramétrage pf ($AgGap_{V.Ob}$) entre les deux solutions $VPL(Veille)$ et $VPL(Scenario)$.

Les résultats montrent que le fait de minimiser le nombre de chauffeurs utilisés dans le planning de la veille améliore généralement la valeur objectif de la solution quelque soit I, pf et pm, sauf pour l'instance $300-1$ où deux cas avec ($pf = 0,5$ $pm = 1$ et$pf = 0,5$ $pm = 3$). L'amélioration peut atteindre jusqu'à 9% au meilleur cas. Généralement, la solution s'améliore plus en moyenne lorsque le pourcentage de requêtes connues la veille augmente.

3.8 Conclusion

Ce chapitre traite un problème de planification en temps réel d'un grand réseau d'ambulance, le problème a été décomposé à travers plusieurs problèmes plus simples :

— P_1 DARP dynamique (Dynamic Dial A Ride Problem) ;

— P_2 DARP dynamique + contrainte temps de travail des ambulanciers ;

— P_3 DARP dynamique + Attribution des pauses aux ambulanciers ;

— P_4 DARP dynamique + Plusieurs types de véhicules ;

— P_5 Problème complet de planification en temps réel (DMTP).

Les problèmes ont été traités avec une approche périodique. Pour chaque période de re-planification (l'instant T_0) deux paramètres (H_{fix} et H_{libre}) entrent en jeu, tels que toutes les affectations des requêtes déjà attribuées aux planifications précédentes dont les heures de début au plus tôt appartiennent à l'intervalle $[T_0, T_0 + H_{fix}]$ restent figées et la nouvelle re-planification tiendra compte de toutes les requêtes dont l'heure de début au plus tôt appartient à $[T_0, T_0 + H_{libre}]$. Trois stratégies d'insertions ont été proposées pour résoudre le problème, à savoir :

— $str1$: algorithme glouton d'insertion, avec F comme fréquence de re-planification.

— $str2$ (Méthode de petit pas) : algorithme d'insertion avec un PLNE. L'insertion ne se fait qu'à la fin des tournées, la re-planification ce fait à chaque fois un chauffeur termine une course.

— $str3$ (Méthode d'anticipation) : algorithme glouton d'insertion, l'insertion ne se fait qu'à la fin des tournées avec F comme fréquence de re-planification.

Les trois stratégies ont été comparées sur les trois premiers problèmes, avec plusieurs paramétrages de H_{fix} , H_{libre} et F. les résultats ont montré que $str1$ est la stratégie efficace, dans plus de 93% des cas en moyenne. Ce qui implique que la stratégie 1 serait normalement la plus efficace pour le problème général P_5.

La deuxième série de tests se concentre exclusivement sur le problème général P_5. Le but de ces tests est de comparer la stratégie $str1$ avec les bornes à information parfaite. La solution de la stratégie $str1$ dans un environnement dynamique à été comparée avec deux bornes, à savoir :

— Solution en utilisant le planning de la veille pour le lendemain dans le contexte où toutes les informations sont révélées au début de la journée.

— Solution en utilisant $str1$ dans le contexte où toutes les informations sont révélées au début de la journée.

Les résultats ont montré que le fait de savoir toutes les informations à l'avance, améliore la qualité de la solution entre 33% à 98%. Le grand écart entre les deux bornes à informations parfaites montre que la stratégie $str1$ est beaucoup moins efficace que l'algorithme ALNS et ce qui est normale en comparaison avec le temps de calcul de chaque algorithme.

La dernière série de tests traite le problème complet de la thèse (partie proactive + partie réactive). Autrement dit, nous étudions l'impact du planning de la veille pour le lendemain dans les décisions prises en temps réel. Les résultats montrent que l'utilisation du planning de la veille pour le lendemain a un impact positif sur les décisions du jour

même. Une amélioration est de $6,9\%$ en moyenne est possible si l'algorithme de la veille est utilisé sur un scénario complet et encore une amélioration de 3% en moyenne si l'algorithme de la veille minimise le nombre de chauffeurs utilisés.

Algorithm 9 Algorithme d'insertion de S_1 pour P_4

Require: Pl^{n-1}, $\mathcal{L}$
Ensure: Nouveau plan prévu Pl^n
$Nit := \sqrt{|\mathcal{L}|}$
Préparation des données : $\mathcal{P}i \underset{(3.3)}{\leftarrow} Pl^{n-1}$
for all $it = 1, \ldots, Nit$ **do**
$Pl^n(it) \leftarrow \mathcal{P}i$
if $it = 1$ **then** $\mathcal{L}_{it} \underset{es_i\ croissant}{\leftarrow} \mathcal{L}$
else $\mathcal{L}_{it} \underset{Ordre\ aleatoire}{\leftarrow} \mathcal{L}$
end if
while $r \leftarrow pop(\mathcal{L}_{it}) \neq \varnothing$ **do**
for all $t \in \mathcal{T}_{Pl^n(it)}$: T est l'ensemble des trips de $Pl^n(it)$ **do**
if l'équipage (E_t) et le véhicule K_t affectés à t peuvent faire r **then**
for all *pos* position d'insertion de r dans le trips de T **do**
Calculer $\Delta Z(t, pos)$
end for
end if
end for
$(t^*, pos^*) := argmin_{e,pos}\{\Delta Z(e, pos)\}$
for all E_r équipage possible pour faire r **do**
Créer une tournée $trip_r$ avec seulement la demande r.
for all pos_{trip_e} position d'insertion du trip t dans $Pl^n(it)$ du chauffeur e ; $\forall e \in E_r$ **do**
for all Vehicule $k \in K_r$ **do**
Calculer $\Delta Z(E_r, k, pos_{trip}, trip_r)$
end for
end for
end for
$(E_r^*, k^*, pos_{trip}^*, trip_r^*) := argmin_{E_r,k,pos_{trip},trip_r}\{\Delta Z(E_r, k, pos_{trip}, trip_r)\}$
if $\Delta Z(E_r^*, k^*, pos_{trip}^*, trip_r^*) \geq \Delta Z(t^*, pos^*)$ **then**
Insérer r dans le trip e^* dans $Pl^n(it)$ à la position pos^*
else Insérer le trip t dans le trip $trip_r^*$ dans $Pl^n(it)$ à l'équipage E_r^* aux positions pos_{trip}^* et au véhicule k^*
end if
end while
end for
$Pl^n := argmin_{it}\{Z(Pl^n(it))\}$

Algorithm 10 Algorithme d'insertion de S_3

Require: Pl^{n-1}, $\mathcal{L}$
Ensure: Nouveau plan prévu Pl^n
$Nit := \sqrt{|\mathcal{L}|}$
Préparation des données : $\mathcal{P}i \underset{(3.3)}{\leftarrow} Pl^{n-1}$
$\mathcal{L}' \underset{es_i\ croissant}{\leftarrow} \mathcal{L}$
while $r \leftarrow pop(\mathcal{L}') \neq \varnothing$ **do**
  for all $e \in E$, e qualifié pour r **do**
    Calculer $\Delta Z(e)$, en insérant r à la fin de journée de e.
  end for
  $(e^*) := argmin_e\{\Delta Z(e)\}$
  Insérer r dans le dernier trip de e^* dans Pl^n à la dérnière position
  for all $r' \in \mathcal{L}'$ **do**
    if $hp_i \leq ls_i\ \forall i \in r'$ si r est insèrer à la fin de journée de e **then**
      Insérer r' dans le dernier trip de e^* dans Pl^n à la dernière position
      $\mathcal{L}' = \mathcal{L}' \setminus \{r'\}$
    elseBreak
    end if
  end for
end while

Algorithm 11 Algorithme pour évaluer une stratégie

Require: Instance I prise à partir de l'historique, un nombre de réplications J (= nombre de jours simulés), un pourcentage de demande connues la veille (pf) et HV qui représente l'heure limite de considération des décisions de la veille pour le lendemain.
$V = REST(I)$ // *Créer instance de la veille*
$G = GENSC(V, I)$ // *Compléter V avec un scénario G*
$PV = ALNS(G, V, HV)$ // *Calcul d'un plan statique PV sur (G, V)*
for all itération $j \in J$ **do**
  $G_j = GENSC(V, I)$ // *Génération d'un nouveau scénario G_j*
  $KPI_j = \zeta(G_j, PV)$ // *Dérouler stratégie ζ sur G_j avec PV comme plan de départ jusqu'à l'heure HV*
end for
Retourner l'intervalle de confiance à 95% de tous les KPI.

Inst	Pr	str_1	str_2					str_3				
		$\%B_t$	$\%B_t$	AgG	Gmax	P_{GMax}	Gmin	P_{GMin}	$\%B_t$	AgG	Gmax	P_{GMax}	Gmin	P_{GMin}
1000-1	P_1	98	0	3,49	4,80	30;15;240	0,77	30;30;60	1,52	0,37	0,69	30;60;120	-0,01	60;30;240
	P_2	100	0	2,02	3,08	10;30;240	1,06	10;30;60	0	1,08	1,76	10;30;60	0,63	30;15;60
	P_3	100	0	5,89	12,55	30;15;240	1,69	30;30;60	0	1,23	3,97	60;60;120	0,29	10;15;120
1000-4	P_1	94	4,55	0,57	1,30	10;15;240	-0,15	30;30;60	1,52	0,35	0,67	10;60;240	0	60;15;240
	P_2	100	0	1,48	1,81	60;15;240	0,93	30;30;60	0	1	1,41	10;30;60	0,64	60;30;240
	P_3	100	0	4,86	8,09	30;15;60	2,96	60;60;120	0	0,70	1,28	60;60;120	0,19	60;30;240
100-1	P_1	82	18,18	0,48	0,84	30;15;120	-0,41	30;30;60	0	0,52	0,99	10;15;60	0,23	30;30;60
	P_2	100	0	2,51	5,30	30;15;240	0,57	30;30;60	0	3,28	6,43	30;15;240	1,50	30;30;60
	P_3	97	0	2,66	5,25	10;15;240	0,45	30;30;60	3,03	1,06	3,16	10;60;240	-0,03	30;60;120
200-1	P_1	100	0	0,27	0,46	10;30;240	0,05	30;30;60	0	0,22	0,48	10;30;60	0,14	60;30;240
	P_2	100	0	0,46	0,62	10;60;240	0,25	60;60;120	0	0,22	0,44	10;30;60	0,12	60;15;120
	P_3	98	0	1,37	1,97	30;15;240	0,11	60;60;120	1,52	0,69	1,12	10;60;240	-0,12	60;60;120
300-2	P_1	73	27,27	0,52	1,83	10;15;240	-0,40	30;30;60	0	0,45	1,25	30;60;240	0,13	30;30;60
	P_2	100	0	2,25	3,22	30;15;120	1,31	60;60;120	0	2,07	3,70	10;60;240	0,94	30;30;60
	P_3	55	12,12	0,37	1,26	10;60;240	-0,16	60;60;120	33,33	0,05	0,34	60;15;240	-0,26	60;60;240

TABLE 3.6 – Comparaisons des stratégies sur chaque problème

Pf	$Z(H_{libre} = 240) < Z(H_{libre} = 120)$	$Z(H_{libre} = 120) < Z(H_{libre} = 60)$	$Z(H_{libre} = 240) < Z(H_{libre} = 60)$
30%	65%	80%	100%
50%	70%	84%	100%
70%	90%	98%	100%

TABLE 3.7 – Influence du paramètre H_{libre} sur la qualité de la solution

Pf	$Z(H_{fix} = 10) < Z(H_{fix} = 30)$	$Z(H_{fix} = 30) < Z(H_{fix} = 60)$	$Z(H_{fix} = 10) < Z(H_{fix} = 60)$
30%	55%	66%	100%
50%	65%	77%	100%
70%	70%	88%	100%

TABLE 3.8 – Influence du paramètre H_{Fix} sur la qualité de la solution avec H_{libre} fixé

Pf	$Z(F = 15) < Z(F = 30)$	$Z(F = 30) < Z(F = 60)$	$Z(F = 15) < Z(F = 60)$
30%	100%	100%	100%
50%	100%	100%	100%
70%	100%	100%	100%

TABLE 3.9 – Influence du paramètre F sur la qualité de la solution avec H_{libre} et H_{Fix} fixés

pf		1000 − 1		1000 − 2		100 − 1		200 − 1		300 − 1	
	Pm	V.Ob	U.R	V.Ob	U.R	V.Ob	U.R	V.Ob	U.R	V.Ob	U.R
0,3	1	503713,2	54,0	460223,0	48,0	52292,2	6,0	38495,2	2,0	35613,8	4,0
	2	463644,0	50,0	358782,9	42,0	41633,8	5,0	29770,0	2,0	29162,5	3,0
	3	500541,4	53,0	418270,9	45,0	42828,8	5,0	29212,7	2,0	27786,8	3,0
	4	496345,3	54,0	432955,8	47,0	48048,4	5,0	30012,1	2,0	32907,1	3,0
0,5	1	460660,5	53,0	449489,2	46,0	49641,1	6,0	28935,6	1,0	34222,1	3,0
	2	439615,8	47,0	382274,1	42,0	38512,5	4,0	26767,1	1,0	27674,9	3,0
	3	464429,1	53,0	383079,7	45,0	54072,3	6,0	22226,6	1,0	27761,0	3,0
	4	470309,4	53,0	428428,0	46,0	46863,1	6,0	34640,5	2,0	33279,2	3,0
0,7	1	433041,8	46,0	444085,0	47,0	48949,1	6,0	25977,6	1,0	33293,3	3,0
	2	376088,1	41,0	378920,7	40,0	35015,3	4,0	23313,4	1,0	24579,3	2,0
	3	398244,1	45,0	381409,9	43,0	42897,7	5,0	25768,7	1,0	26518,7	2,0
	4	414782,5	45,0	427645,3	45,0	43935,8	5,0	25735,8	1,0	25992,2	2,0

TABLE 3.10 – Résultats avec une seule réplication de la stratégie 1 sur 4 paramètres

pf		1000 − 1		1000 − 2		100 − 1		200 − 1		300 − 1	
	Pm	V.Ob	U.R	V.Ob	U.R	V. Obj	U.R	V.Ob	U.R	V.Ob	U.R
0,3	1	359479,6	33,0	309675,2	29,0	36547,5	4,0	24652,3	1,0	20543,0	2,0
	2	347635,6	32,0	246209,8	25,0	27302,6	3,0	19006,5	1,0	19335,4	2,0
	3	345814,9	32,0	266142,7	26,0	32895,0	4,0	18270,4	1,0	19922,3	2,0
	4	343481,1	35,0	287714,2	28,0	33314,3	4,0	19015,9	1,0	20761,3	2,0
0,5	1	317623,9	36,0	301357,6	27,0	25545,2	3,0	21234,7	1,0	23365,9	2,0
	2	277267,7	28,0	244157,1	26,0	21157,9	2,0	15296,8	0,0	19356,0	2,0
	3	290220,2	33,0	228385,1	26,0	27383,6	3,0	17244,6	1,0	19112,3	2,0
	4	324943,7	34,0	272187,9	28,0	31565,0	4,0	22208,9	1,0	21650,8	2,0
0,7	1	293382,6	27,0	266160,8	27,0	28709,5	3,0	14710,6	0,0	22199,9	2,0
	2	243420,3	25,0	228183,9	23,0	20218,4	2,0	17384,4	1,0	15829,2	1,0
	3	262560,1	28,0	236489,3	26,0	21715,8	2,0	19042,5	1,0	16785,2	1,0
	4	260233,1	28,0	276036,5	27,0	25640,9	3,0	18713,3	1,0	15834,5	1,0

TABLE 3.11 – Résultats avec une seule réplication de l'algorithme ALNS sur 4 paramètres

pf		1000 − 1		1000 − 2		100 − 1		200 − 1		300 − 1	
	Pm	V.Ob	U.R	V.Ob	U.R	V.Ob	U.R	V.Ob	U.R	V.Ob	U.R
0,3	1	425519,4	42,0	391987,7	36,0	46480,7	5,0	25583,5	1,0	25237,2	3,0
	2	378434,0	39,0	308750,3	32,0	36035,5	4,0	20028,2	1,0	20308,1	2,0
	3	435150,4	40,0	338219,6	32,0	37594,2	4,0	19139,3	1,0	20713,8	2,0
	4	424475,2	40,0	348549,6	34,0	42746,1	4,0	19536,2	1,0	21255,2	2,0
0,5	1	398571,9	41,0	395684,5	36,0	48098,7	5,0	21661,5	1,0	24976,6	2,0
	2	379872,0	39,0	342567,7	35,0	32910,8	3,0	19190,0	1,0	20440,0	2,0
	3	399780,0	42,0	322930,9	35,0	47480,1	5,0	18088,7	1,0	20111,2	2,0
	4	408940,7	43,0	369610,1	37,0	41534,7	5,0	22688,7	1,0	22182,6	2,0
0,7	1	394830,2	39,0	394282,5	39,0	43645,6	5,0	18736,1	1,0	23029,7	2,0
	2	344364,2	36,0	339817,8	35,0	30014,4	3,0	18172,1	1,0	16812,7	1,0
	3	356485,2	37,0	332286,5	35,0	37772,7	4,0	20268,3	1,0	16824,0	1,0
	4	361976,1	36,0	373467,5	36,0	39117,5	4,0	18991,0	1,0	16053,7	1,0

TABLE 3.12 – Résultats avec une seule réplication de la stratégie 1 sur 4 paramètres avec informations parfaites

pf		1000 − 1		1000 − 2		100 − 1		200 − 1		300 − 1	
	Pm	V.Ob	U.R	V.Ob	U.R	V.Ob	U.R	V.Ob	U.R	V.Ob	U.R
0,3	1	0,18	0,29	0,17	0,33	0,13	0,20	0,50	1,00	0,41	0,33
	2	0,23	0,28	0,16	0,31	0,16	0,25	0,49	1,00	0,44	0,50
	3	0,15	0,33	0,24	0,41	0,14	0,25	0,53	1,00	0,34	0,50
	4	0,17	0,35	0,24	0,38	0,12	0,25	0,54	1,00	0,55	0,50
0,5	1	0,16	0,29	0,14	0,28	0,03	0,20	0,34	0,00	0,37	0,50
	2	0,16	0,21	0,12	0,20	0,17	0,33	0,39	0,00	0,35	0,50
	3	0,16	0,26	0,19	0,29	0,14	0,20	0,23	0,00	0,38	0,50
	4	0,15	0,23	0,16	0,24	0,13	0,20	0,53	1,00	0,50	0,50
0,7	1	0,10	0,18	0,13	0,21	0,12	0,20	0,39	0,00	0,45	0,50
	2	0,09	0,14	0,12	0,14	0,17	0,33	0,28	0,00	0,46	1,00
	3	0,12	0,22	0,15	0,23	0,14	0,25	0,27	0,00	0,58	1,00
	4	0,15	0,25	0,15	0,25	0,12	0,25	0,36	0,00	0,62	1,00

TABLE 3.13 – Gap entre solution stratégie 1 et stratégie 1 avec information parfaite

pf		1000 − 1		1000 − 2		100 − 1		200 − 1		300 − 1	
	Pm	V.Ob	U.R	V.Ob	U.R	V.Ob	U.R	V.Ob	U.R	V.Ob	U.R
0,3	1	0,40	0,64	0,49	0,66	0,43	0,50	0,56	1,00	0,73	1,00
	2	0,33	0,56	0,46	0,68	0,52	0,67	0,57	1,00	0,51	0,50
	3	0,45	0,66	0,57	0,73	0,30	0,25	0,60	1,00	0,39	0,50
	4	0,45	0,54	0,50	0,68	0,44	0,25	0,58	1,00	0,59	0,50
0,5	1	0,45	0,47	0,49	0,70	0,94	1,00	0,36	0,00	0,46	0,50
	2	0,59	0,68	0,57	0,62	0,82	1,00	0,75	-	0,43	0,50
	3	0,60	0,61	0,68	0,73	0,97	1,00	0,29	0,00	0,45	0,50
	4	0,45	0,56	0,57	0,64	0,48	0,50	0,56	1,00	0,54	0,50
0,7	1	0,48	0,70	0,67	0,74	0,70	1,00	0,77	-	0,50	0,50
	2	0,55	0,64	0,66	0,74	0,73	1,00	0,34	0,00	0,55	1,00
	3	0,52	0,61	0,61	0,65	0,98	1,50	0,35	0,00	0,58	1,00
	4	0,59	0,61	0,55	0,67	0,71	0,67	0,38	0,00	0,64	1,00

TABLE 3.14 – Gap entre la solution de la stratégie 1 et VPL

			$HV = 0h$		$HV = 12h$		$HV = 18h$		$Gap_{0,12}$	$Gap_{12,18}$
I	pf	pm	V.Ob	U.R	V.Ob	U.R	V.Ob	U.R		
1000-1	0,3	1	495510,50	53,97	505730,82	56,07	561212,56	67,47	-0,02	-0,09
		2	459894,52	50,09	468184,27	51,79	516459,85	61,69		
		3	481489,44	52,45	489769,13	54,15	535550,98	63,55		
		4	491153,66	53,50	496995,00	54,70	543725,70	64,30		
	0,5	1	480776,94	52,36	468353,68	49,81	502213,14	56,76	0,02	-0,06
		2	446616,36	48,64	434902,16	46,24	465163,84	52,44		
		3	480776,94	52,36	468353,68	49,81	502213,14	56,76		
		4	480776,94	52,36	488328,33	53,91	505623,44	57,46		
	0,7	1	434178,56	46,44	408821,04	41,24	421889,92	43,92	0,07	-0,03
		2	382608,65	41,88	357648,59	36,78	367926,26	38,88		
		3	386678,30	44,22	362187,89	39,22	373747,36	41,58		
		4	423988,15	46,13	395681,06	40,33	410517,88	43,37		
1000-4	0,3	1	435166,20	47,57	445714,65	49,77	489826,39	58,97	-0,01	-0,09
		2	390712,73	42,71	394564,76	43,51	434048,07	51,71		
		3	413338,78	45,18	416221,55	45,78	467150,58	56,38		
		4	424437,18	46,40	431156,31	47,80	473390,85	56,60		
	0,5	1	446153,45	44,53	435398,98	42,28	463838,58	48,23	0,03	-0,06
		2	387029,53	43,31	376896,47	41,21	387029,53	43,31		
		3	377697,39	44,41	366809,60	42,16	398989,07	48,81		
		4	428652,74	45,92	417858,38	43,67	450241,44	50,42		
	0,7	1	450961,19	49,62	422155,88	43,61	438862,96	47,10	0,07	-0,04
		2	382608,65	41,88	356561,48	36,48	371032,13	39,48		
		3	386678,30	44,22	360112,56	38,72	371608,28	41,10		
		4	423988,15	46,13	396143,00	40,33	413042,12	43,85		

TABLE 3.15 – Comparaison entre HV=720 et HV=1200 sur grandes instances

			$HV = 0h$		$HV = 12h$		$HV = 18h$		$Gap_{0,12}$	$Gap_{12,18}$
I	pf	pm	V.Ob	U.R	V.Ob	U.R	V.Ob	U.R		
100-1	0,3	1	49651,18	6,02	50132,04	6,12	56864,16	7,51	-0,02	-0,11
		2	44412,37	5,64	45380,07	5,83	51186,29	7,03		
		3	43060,41	4,96	44507,41	5,25	49330,77	6,25		
		4	45947,00	5,11	46908,25	5,30	51714,51	6,30		
	0,5	1	51153,18	5,98	49717,96	5,67	54023,61	6,57	-0,01	-0,04
		2	44222,22	5,01	45184,18	5,21	46627,12	5,51		
		3	50266,82	5,64	51699,61	5,94	52654,80	6,14		
		4	46654,94	5,82	47619,98	6,02	49550,07	6,42		
	0,7	1	47336,91	5,64	43968,72	4,94	45027,29	5,16	0,08	-0,04
		2	39293,77	4,83	36381,94	4,23	38129,04	4,59		
		3	43044,68	5,20	39658,91	4,50	41303,43	4,84		
		4	42447,77	4,84	39072,53	4,14	40711,93	4,48		
200-1	0,3	1	36291,56	2,43	37245,64	2,63	39153,79	3,03	-0,02	-0,06
		2	26385,69	1,64	26868,20	1,74	28315,75	2,04		
		3	24919,43	1,69	25403,45	1,79	27339,51	2,19		
		4	29244,19	2,00	29725,81	2,10	31652,29	2,50		
	0,5	1	28930,88	1,73	29651,52	1,88	28930,88	1,73	0,01	-0,03
		2	26045,27	1,36	25322,56	1,21	27008,88	1,56		
		3	22940,24	1,35	23182,24	1,40	23908,24	1,55		
		4	27983,64	1,52	27261,02	1,37	28947,12	1,72		
	0,7	1	31131,82	1,90	29687,66	1,60	30554,15	1,78	0,05	-0,03
		2	23785,11	1,32	22818,21	1,12	23785,11	1,32		
		3	24720,85	1,21	23756,15	1,01	24142,03	1,09		
		4	26310,48	1,64	24862,77	1,34	25731,39	1,52		
300-1	0,3	1	36325,74	4,03	37294,33	4,23	40684,41	4,92	-0,02	-0,10
		2	31236,18	3,60	31723,84	3,70	35625,16	4,49		
		3	30355,52	3,41	30842,79	3,51	34740,95	4,31		
		4	34270,66	3,67	34755,58	3,77	38634,98	4,57		
	0,5	1	34423,92	3,06	33701,28	2,91	35869,19	3,36	0,00	-0,05
		2	28104,11	3,11	28836,01	3,26	30055,85	3,51		
		3	27804,02	3,02	28534,54	3,17	29265,07	3,32		
		4	30403,19	3,02	29674,93	2,87	31859,69	3,32		
	0,7	1	35901,22	3,36	33490,73	2,86	34744,18	3,12	0,07	-0,03
		2	27420,78	2,85	25469,26	2,44	26249,87	2,61		
		3	27213,48	2,64	25266,58	2,23	26045,34	2,40		
		4	29420,76	2,64	27482,61	2,24	28257,87	2,40		

TABLE 3.16 – Comparaison entre HV=720 et HV=1200 sur moyennes instances

			$VPL(Scenario)$		$VPL(Veille)$		$Gaps_{VPL(Veille),VPL(Scenario)}$		
I	pf	pm	V.Ob	U.R	V.Ob	U.R	$Gap_{V.Ob}$	$Gap_{U.R}$	$AgGap_{V.Ob}$
1000-1	0,3	1	486267,79	52,07	505730,82	56,07	0,04	0,08	0,04
		2	451118,41	48,29	468184,27	51,79	0,04	0,07	
		3	471749,51	50,44	489769,13	54,15	0,04	0,07	
		4	480934,71	51,40	496995,00	54,70	0,03	0,06	
	0,5	1	445716,17	45,16	468353,68	49,81	0,05	0,10	0,07
		2	403671,03	39,84	434902,16	46,24	0,08	0,16	
		3	441336,82	44,26	468353,68	49,81	0,06	0,13	
		4	442797,32	44,56	488328,33	53,91	0,10	0,21	
	0,7	1	384460,06	36,24	408821,04	41,24	0,06	0,14	0,09
		2	326845,33	30,48	357648,59	36,78	0,09	0,21	
		3	329895,08	32,62	362187,89	39,22	0,10	0,20	
		4	362525,98	33,53	395681,06	40,33	0,09	0,20	
1000-4	0,3	1	426537,99	45,76	445714,65	49,77	0,04	0,09	0,04
		2	381565,22	40,81	394564,76	43,51	0,03	0,07	
		3	404218,33	43,28	416221,55	45,78	0,03	0,06	
		4	415803,32	44,59	431156,31	47,80	0,04	0,07	
	0,5	1	406989,01	36,33	435398,98	42,28	0,07	0,16	0,10
		2	351813,52	36,01	376896,47	41,21	0,07	0,14	
		3	321626,39	32,81	366809,60	42,16	0,14	0,28	
		4	378819,74	35,51	417858,38	43,67	0,10	0,23	
	0,7	1	394852,04	37,91	422155,88	43,61	0,07	0,15	0,07
		2	334871,75	31,98	356561,48	36,48	0,06	0,14	
		3	336967,96	33,92	360112,56	38,72	0,07	0,14	
		4	372647,68	35,43	396143,00	40,33	0,06	0,14	

TABLE 3.17 – Résultats avec une seule réplication de la stratégie 1 pour 4 paramètres avec information parfaite sur les grandes instances

			$VPL(Scenario)$		$VPL(Veille)$		$Gaps_{VPL(Veille),VPL(Scenario)}$		
I	pf	pm	V.Ob	U.R	V.Ob	U.R	$Gap_{V.Ob}$	$Gap_{U.R}$	$AgGap_{V.Ob}$
100-1	0,3	1	47739,38	5,62	50132,04	6,12	0,05	0,09	0,05
		2	43448,45	5,44	45380,07	5,83	0,04	0,07	
		3	42580,82	4,86	44507,41	5,25	0,05	0,08	
		4	44994,13	4,91	46908,25	5,30	0,04	0,08	
	0,5	1	47345,85	5,18	49717,96	5,67	0,05	0,10	0,13
		2	38978,17	3,91	45184,18	5,21	0,16	0,33	
		3	44554,77	4,44	51699,61	5,94	0,16	0,34	
		4	40901,77	4,62	47619,98	6,02	0,16	0,30	
	0,7	1	41118,39	4,34	43968,72	4,94	0,07	0,14	0,08
		2	33000,62	3,53	36381,94	4,23	0,10	0,20	
		3	36799,03	3,90	39658,91	4,50	0,08	0,15	
		4	36711,68	3,64	39072,53	4,14	0,06	0,14	
200-1	0,3	1	36291,56	2,43	37245,64	2,63	0,03	0,08	0,03
		2	25912,25	1,54	26868,20	1,74	0,04	0,13	
		3	24436,83	1,59	25403,45	1,79	0,04	0,13	
		4	28773,26	1,90	29725,81	2,10	0,03	0,11	
	0,5	1	27976,90	1,53	29651,52	1,88	0,06	0,23	0,05
		2	24185,46	0,96	25322,56	1,21	0,05	0,26	
		3	21517,48	1,05	23182,24	1,40	0,08	0,33	
		4	26565,65	1,22	27261,02	1,37	0,03	0,12	
	0,7	1	27764,09	1,20	29687,66	1,60	0,07	0,33	0,05
		2	21913,84	0,92	22818,21	1,12	0,04	0,22	
		3	22841,37	0,81	23756,15	1,01	0,04	0,25	
		4	23457,84	1,04	24862,77	1,34	0,06	0,29	
300-1	0,3	1	35845,84	3,93	37294,33	4,23	0,04	0,08	0,04
		2	30266,92	3,40	31723,84	3,70	0,05	0,09	
		3	29384,38	3,20	30842,79	3,51	0,05	0,09	
		4	33307,76	3,47	34755,58	3,77	0,04	0,09	
	0,5	1	32518,12	2,66	33701,28	2,91	0,04	0,09	0,10
		2	25203,94	2,51	28836,01	3,26	0,14	0,30	
		3	26355,82	2,72	28534,54	3,17	0,08	0,17	
		4	26534,75	2,21	29674,93	2,87	0,12	0,29	
	0,7	1	31592,37	2,46	33490,73	2,86	0,06	0,16	0,09
		2	22567,32	1,84	25469,26	2,44	0,13	0,33	
		3	23333,54	1,83	25266,58	2,23	0,08	0,22	
		4	25105,25	1,74	27482,61	2,24	0,09	0,29	

TABLE 3.18 – Résultats avec une seule réplication de la stratégie 1 sur 4 paramètres avec information parfaite sur les instances moyennes

			AvecMinimisation		SansMinimisation		$Gaps_{SansMin,AvecMin}$		
I	pf	pm	V.Ob	U.R	V.Ob	U.R	$Gap_{V.Ob}$	$Gap_{U.R}$	$AgGap_{V.Ob}$
1000-1	0,3	1	479450,53	50,67	486267,79	52,07	0,01	0,03	0,01
		2	444290,74	46,89	451118,41	48,29	0,02	0,03	
		3	468339,78	49,74	471749,51	50,44	0,01	0,01	
		4	475577,32	50,30	480934,71	51,40	0,01	0,02	
	0,5	1	438396,82	43,66	445716,17	45,16	0,02	0,03	0,02
		2	395861,69	38,24	403671,03	39,84	0,02	0,04	
		3	429142,19	41,76	441336,82	44,26	0,03	0,06	
		4	433526,03	42,66	442797,32	44,56	0,02	0,04	
	0,7	1	366407,09	32,54	384460,06	36,24	0,05	0,11	0,05
		2	309697,63	26,98	326845,33	30,48	0,06	0,13	
		3	311749,31	28,92	329895,08	32,62	0,06	0,13	
		4	344451,14	29,83	362525,98	33,53	0,05	0,12	
1000-4	0,3	1	426056,90	45,66	426537,99	45,76	0,00	0,00	0,01
		2	374342,85	39,31	381565,22	40,81	0,02	0,04	
		3	402289,05	42,88	404218,33	43,28	0,00	0,01	
		4	408120,58	42,99	415803,32	44,59	0,02	0,04	
	0,5	1	397419,35	34,33	406989,01	36,33	0,02	0,06	0,02
		2	343614,21	34,31	351813,52	36,01	0,02	0,05	
		3	313365,17	31,11	321626,39	32,81	0,03	0,05	
		4	371589,46	34,01	378819,74	35,51	0,02	0,04	
	0,7	1	373707,03	33,51	394852,04	37,91	0,06	0,13	0,06
		2	316551,18	28,18	334871,75	31,98	0,06	0,13	
		3	320048,18	30,42	336967,96	33,92	0,05	0,12	
		4	352968,31	31,33	372647,68	35,43	0,06	0,13	

TABLE 3.19 – Comparaison entre la minimisation et sans minimisation du nombre de chauffeurs utilisés dans VPL

			AvecMinimisation		SansMinimisation		$Gaps_{SansMin,AvecMin}$		
I	pf	pm	V.Ob	U.R	V.Ob	U.R	$Gap_{V.Ob}$	$Gap_{U.R}$	$AgGap_{V.Ob}$
100-1	0,3	1	47247,29	5,52	47739,38	5,62	0,01	0,02	0,02
		2	42960,92	5,34	43448,45	5,44	0,01	0,02	
		3	41131,14	4,56	42580,82	4,86	0,04	0,07	
		4	44024,68	4,71	44994,13	4,91	0,02	0,04	
	0,5	1	45892,84	4,88	47345,85	5,18	0,03	0,06	0,03
		2	37972,80	3,71	38978,17	3,91	0,03	0,05	
		3	43107,17	4,14	44554,77	4,44	0,03	0,07	
		4	39903,07	4,42	40901,77	4,62	0,03	0,05	
	0,7	1	37719,37	3,64	41118,39	4,34	0,09	0,19	0,07
		2	31047,11	3,13	33000,62	3,53	0,06	0,13	
		3	34343,01	3,40	36799,03	3,90	0,07	0,15	
		4	34736,41	3,24	36711,68	3,64	0,06	0,12	
200-1	0,3	1	35814,52	2,33	36291,56	2,43	0,01	0,04	0,02
		2	25420,71	1,43	25912,25	1,54	0,02	0,07	
		3	23951,43	1,49	24436,83	1,59	0,02	0,07	
		4	28281,01	1,80	28773,26	1,90	0,02	0,06	
	0,5	1	27489,73	1,42	27976,90	1,53	0,02	0,07	0,03
		2	23636,94	0,86	24185,46	0,96	0,02	0,12	
		3	21004,55	0,94	21517,48	1,05	0,02	0,11	
		4	26057,00	1,12	26565,65	1,22	0,02	0,09	
	0,7	1	26800,55	1,00	27764,09	1,20	0,04	0,2	0,04
		2	21368,51	0,82	21913,84	0,92	0,03	0,12	
		3	22309,71	0,71	22841,37	0,81	0,02	0,14	
		4	22451,10	0,84	23457,84	1,04	0,04	0,24	
300-1	0,3	1	35357,18	3,83	36325,74	3,41	0,03	-0,11	0,04
		2	29773,27	3,29	31236,18	3,67	0,05	0,11	
		3	28893,80	3,11	30355,52	2,16	0,05	-0,30	
		4	32815,99	3,37	34270,66	2,61	0,04	-0,23	
	0,5	1	31533,94	2,46	30123,50	2,22	-0,04	-0,10	-0,01
		2	24201,43	2,31	25681,38	2,11	0,06	-0,08	
		3	25856,15	2,62	23929,78	2,46	-0,07	-0,06	
		4	25549,49	2,01	26068,58	2,04	0,02	0,01	
	0,7	1	30118,33	2,16	31596,01	1,83	0,05	-0,15	0,07
		2	21567,96	1,64	23533,58	1,94	0,09	0,18	
		3	21860,85	1,53	23328,87	2,00	0,07	0,30	
		4	24576,83	1,64	26065,59	1,33	0,06	-0,19	

TABLE 3.20 – Comparaison entre la minimisation et sans minimisation de chauffeurs utilisés dans VPL

Chapitre 4

Implémentation industrielle

Résumé

Dans ce chapitre, nous présentons l'application Lomaco-Online (LOL), ses fonctionnalités et ses interfaces. En plus de l'ensemble des algorithmes développés précédemment, d'autres algorithmes d'aide à la décision sont implémentés dans LOL afin d'aider le régulateur à prendre ses propres décisions, parmi ces aides :

— Aide à la régulation manuelle : procédure gloutonne pour aider le régulateur à construire le planning de la veille plus rapidement ;

— Aide aux simultanés : Permet de lister en temps réel pour chaque demande un ensemble de demandes les plus avantageuses pour être regroupées en transport partagé ;

— Aide à l'affectation : permet de lister en temps réel les équipages les plus mieux adaptés à répondre à une demande sélectionnée.

Une série de tests est effectuée chez les clients afin de bien calibrer et personnaliser les algorithmes selon leurs profils et intérêts. Durant ces tests, plusieurs clients ont demandé des adaptations particulières qu'ils estiment nécessaires pour le bon fonctionnement des algorithmes, parmi ces demandes :

— Gestion des points d'embauche et de débauche des chauffeurs : Les chauffeurs peuvent démarrer et finir leur journée dans un point hors dépôt.

— Minimisation du nombre de ressources utilisées dans le planning de la veille pour le lendemain.

— Donner la priorité aux véhicules fortement kilométrés.

Nous montrons dans ce chapitre aussi, les adaptations apportées aux algorithmes pour la prise en compte de ces demandes.

Sommaire

4.1 Introduction 122
4.2 Lomaco Online (LOL) 122
4.2.1 Fonctionnalités 123
4.2.2 Interface Régulation / Planning 124
4.3 Intégration des demandes des clients 127
4.3.1 Gestion des différents points d'embauche et de débauche 127
4.3.2 Critère tarifaire des choix de type de véhicule 128
4.3.3 Minimiser le nombre de ressources utilisées dans le planning de la veille pour le lendemain 129
4.3.4 Donner la priorité aux véhicules fortement kilométrés 129
4.4 Conclusion 130

4.1 Introduction

Ce chapitre est consacré à l'implémentation des algorithmes développés durant la thèse dans une application industrielle (Lomaco-Online). Une série de tests a été effectuée par les clients afin d'évaluer leurs préférences et de bien calibrer les paramètres des algorithmes. Plusieurs nouveaux objectifs et contraintes qu'ils estiment nécessaires pour le bon fonctionnement de leur régulation ont été remontés. Ces nouveaux objectifs et contraintes nécessitent parfois des modifications importantes sur les algorithmes développés.

Dans la section 4.2 nous présentons les fonctionnalités de l'application développée. Pour des raisons de confidentialité, certaines fonctionnalités ne sont pas présentées ou détaillées. Puis dans la section 4.3 nous listons l'ensemble des requêtes formulées par les clients et comment les intégrer dans les algorithmes présentés dans les chapitres précédents.

4.2 Lomaco Online (LOL)

L'application Lomaco Online est un logiciel métier pouvant couvrir les besoins de plusieurs métiers :

— Pompes funèbres ;

— Fournisseurs en matériel médical ;

— Gestion comptable ;

— Transports sanitaires.

Aucune installation n'est requise pour utiliser l'application, c'est une plateforme *full*-Web utilisable sur un simple navigateur, il suffit de saisir une URL dans le navigateur et l'accès est immédiat une fois l'identification effectuée. Il est accessible partout, tout le temps depuis n'importe quel ordinateur, tablette ou téléphone du moment qu'on dispose d'une connexion internet.

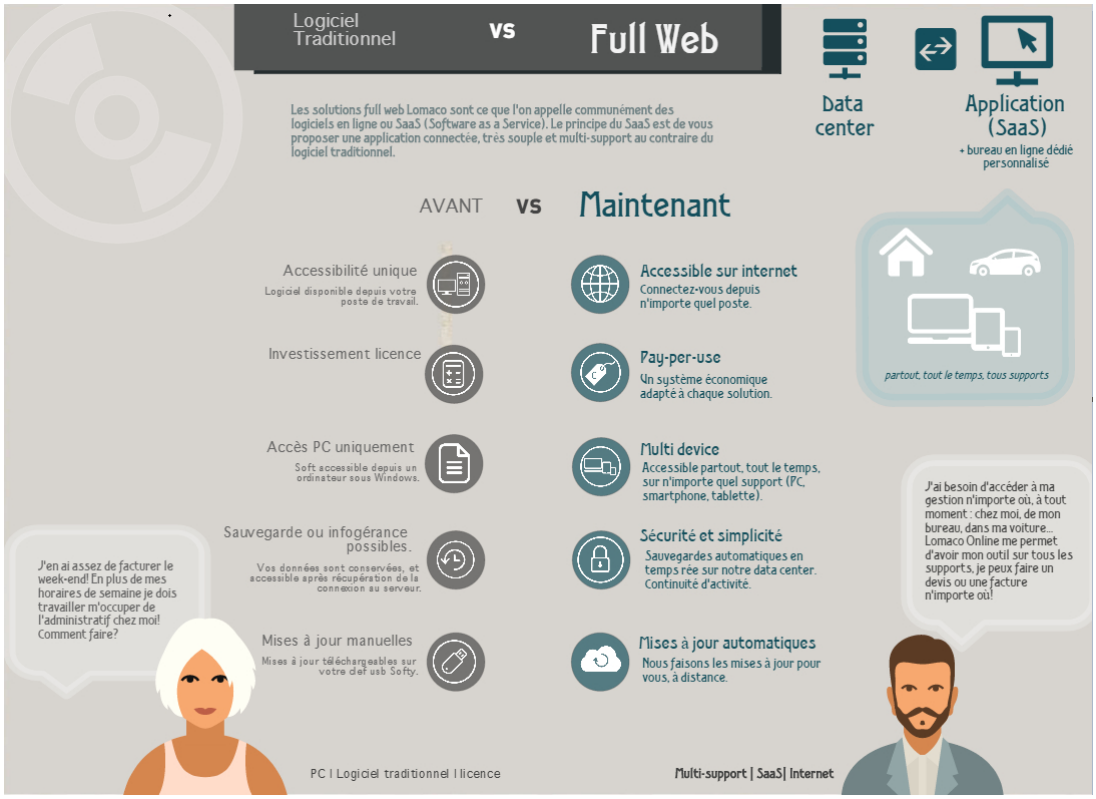

FIGURE 4.1 – Application LOL

Le logiciel est sécurisé et simple à utiliser. La sauvegarde se fait en temps réel et la continuité d'activité est garantie en cas d'incident. Les mises à jour sont faites à distance et sans perturber les activités des utilisateurs. Les solutions full web Lomaco sont ce que l'on appelle communément des logiciels en ligne ou SaaS (Software as a Service). Le principe du SaaS est de proposer une application connectée, très souple et multi-support (Voir Figure 4.1)

4.2.1 Fonctionnalités

Lomaco Online permet de :

1. Facturer les courses des transports sanitaires (ambulances, VSL, taxis). Le service de facturation intégré, permet de mieux contrôler les factures, dématérialiser les documents et d'être remboursé plus rapidement.
2. Réaliser une facturation libre et contractuelle et préparer aussi des devis.
3. Gérer des heures de travail. Cet outil d'organisation est dédié à la gestion du temps de travail du personnel, en accord avec la convention des ambulanciers et le droit du travail. Il permet de planifier et gérer les horaires en lien avec le planning et la régulation de la flotte de véhicules.
4. Localiser n'importe quel véhicule sur différentes cartographies et de visualiser les statuts des véhicules, les détails des transports en cours, l'équipage, etc...

5. Remonter des positions GPS et envoyer leurs missions aux ambulanciers via smartphone.
6. Avoir une interface Régulation / Planning.

Dans ce chapitre, nous allons nous intéresser particulièrement à la fonctionnalité 6 : interface Régulation / Planning.

4.2.2 Interface Régulation / Planning

L'interface permet de sélectionner une journée de travail et de visualiser l'ensemble des demandes, leurs affectations aux équipages et véhicules, les horaires de prise en charge et de dépose prévus, le lieu de prise en charge et de dépose, la raison du transport et le kilométrage nécessaire pour effectuer la mission. L'interface permet aussi de réaliser des filtres détaillés, qui permettent de visualiser l'enclenchement des demandes d'un véhicule, chauffeur ou équipage (voir figure 4.2).

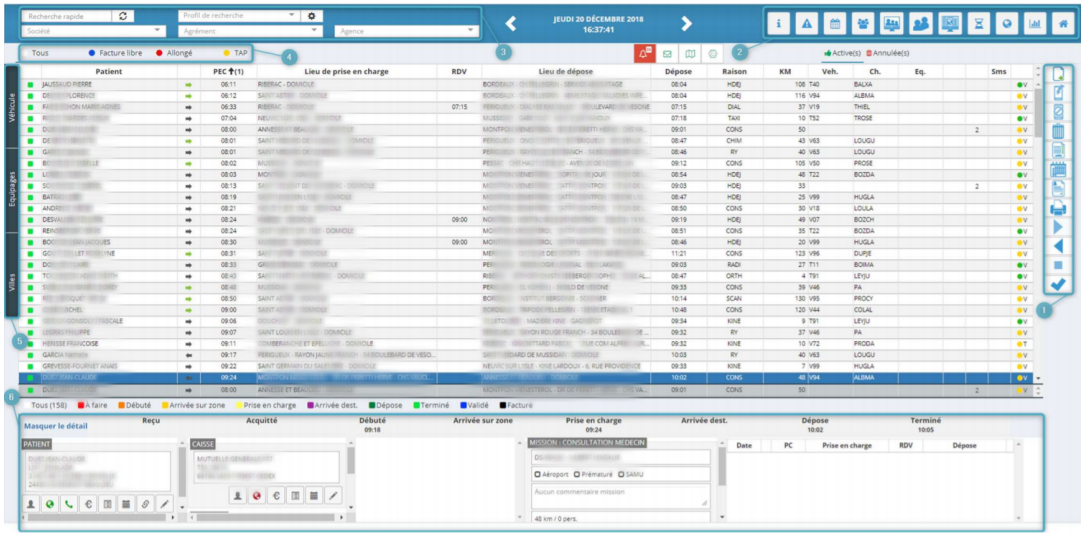

FIGURE 4.2 – Capture LOL Interface régulation

Une barre de commande est affichée aussi sur l'interface qui regroupe plusieurs fonctionnalités, comme saisir les données d'un nouveau patient, envoyer les missions sur mobile aux chauffeurs, etc... (voir figure 4.3)

4.2.2.1 Saisir un nouveau RDV

Comme montré dans 2.3.4, les rendez-vous diffèrent selon :

— Le type de véhicule TAP (VSL ou Taxi) ou ambulance pour l'allongé.

— L'origine de l'appel : Établissement de santé, urgence, médecin, patient, ...

— La raison du transport : consultation par exemple.

— Un trajet aller vers un établissement de santé ou retour (retour au domicile).

— La date et l'heure du rendez-vous. L'outil calcule directement, grâce au service cartographique, les fenêtres de temps de prise en charge et de dépose par rapport à l'heure du rendez-vous.

— L'adresse de prise en charge et de dépose.

— D'autres informations qui peuvent êtres rajoutées dans la rubrique suppléments comme le nombre de personnes en accompagnement, le refus de transport partagé ... (voir figure 4.4)

4.2.2.2 Affecter manuellement un transport

Lorsqu'une requête est affichée sur l'écran, on peut affecter un véhicule et un équipage selon la disponibilité. La course peut être envoyée directement au chauffeur sur son mobile en cliquant sur l'icône du mobile et la mission sera marquée avec un [T], i.e., Transmise comme dans la figure 4.5.

Après avoir validé la course (par le chauffeur), son statut change en fonction de l'évolution de la mission (Débutée, En charge, Dépose, Signature du patient, Terminée).

4.2.2.3 Exigences et préférences

Dans la modélisation mathématique du problème, les exigences et préférences des requêtes sont exprimées par rapport à des propriétés des ambulanciers et véhicules.

Comme illustré dans la figure 4.6, une liste des exigences types est créée dans l'application ; chacune est définie pour chaque acteur (usager, ambulancier, véhicule) selon trois modes : bloquant (rouge), non-bloquant (jaune) ou sans objet (vide).

Ces entités "exigences types" peuvent ensuite être affectées à des requêtes, des ambulanciers ou des véhicules. Ainsi l'exigence bariatrique telle que définie dans la figure 4.6, indique que le véhicule ne peut transporter que des personnes avec cette exigence. Inversement une requête avec l'exigence bariatrique ne peut être servie que par un véhicule bariatrique. En revanche, n'importe quel chauffeur peut servir la requête.

Un véhicule équipé par des fauteuils roulants peut transporter des usagers sans cette exigence, tandis que les usagers en fauteuil roulant imposent l'utilisation d'un véhicule adapté.

Dans cette représentation, il est aisé de se ramener à la modélisation des chapitres précédents.

4.2.2.4 Outils d'aide à la décision

Outre les algorithmes développés dans cette thèse, d'autres outils sont également intégrés à LOL pour aider les planificateurs.

L'algorithme de la veille pour le lendemain présenté dans le chapitre 2, est actuellement intégré dans LOL. Par contre, l'algorithme de planification en temps réel est en cours d'intégration.

Une interface est proposée pour la planification automatique. Pour le bon fonctionnement de l'algorithme, l'utilisateur doit paramétrer les poids des objectifs, l'amplitude des fenêtres de temps, les temps de services, les pauses et le temps de calcul souhaité (voir figure 4.7).
L'utilisateur peut désigner la plage horaire de son planning, par exemple établir uniquement un planning pour les premières requêtes de la journée (voir figure 4.8).
Une fois le calcul lancé, les affectations se font automatiquement et s'affichent directement dans l'interface régulation (voir figure 4.9). En plus des algorithmes de planification automatique présentés précédemment dans les chapitres 2 et 3, d'autres outils d'aide à la décision pour les planifications manuelles ont été développés dans LOL.

4.2.2.5 Aides manuelles

Ils existent trois types d'aides manuelles développées.

— Aide à la régulation ;

— Aide aux simultanés ;

— Aide à la planification.

Aide à la régulation manuelle L'aide à la régulation est un outil utilisé spécifiquement pour la planification de la veille pour le lendemain. L'aide à la régulation manuelle, permet au régulateur d'organiser le planning de la veille pour le lendemain plus rapidement en remplissant la journée de travail de chaque ressource successivement. Pour chaque ressource sélectionnée (véhicule/équipage) e, l'algorithme propose au fur et mesure, une liste des requêtes triées par un score Ser qui représentent les susceptibles prochaines destinations. L'algorithme laisse le choix au régulateur de la prochaine destination pour la ressource ; Ser représente la somme pondérée des trois critères : kilométrage à vide Km_{vide}, retard $retard$ et temps d'attente $TempsAttente$.

Aide aux simultanés L'aide aux simultanés permet au régulateur d'organiser les trajets partagés et de visualiser les courses partagées réalisables à travers la cartographie qu'elle soit pour le planning de la veille pour le lendemain ou en temps réel. En sélectionnant une demande, l'algorithme calcule un score de simultanéité entre la requête en question et l'ensemble des demandes non encore affectées. L'algorithme peut être paramétré pour utiliser des filtres sur une zone géographique ou un créneau de prise en charge et de dépose limité. Le score de simultanéité Sim_{ri} est calculé entre la requête sélectionnée r et toutes les autres demandes i. Le score Sim_{ri} représente une somme pondérée de trois critères. $S_{cr} = P_1 Kmvide + P_2 retard + P_3 TempsAttente$. L'algorithme propose l'ensemble des possibilités triées par le score Sim_{ri} et laisse le choix au régulateur pour prendre la décision qui lui convient le plus. Une aide visuelle en utilisant

la cartographie est proposée au régulateur avec plusieurs détails et statistiques comme dans la figure 4.10.

Aide à l'affectation L'aide à l'affectation permet de calculer un score d'affectation S_{cr} d'une ressource (véhicule+chauffeur) c à une nouvelle demande r. Le score S_{cr} représente une somme pondérée de plusieurs critères ; $S_{cr} = P_1 Kmvide + P_2 retard + P_3 TempsAttente$; avec Km_{vide} qui représente le nombre de kilomètres durant lesquels le véhicule ne transporte aucun patient, *retard* qui représente le nombre de minutes de retards générées et $TempsAttente$ les temps en minutes où la ressource doit attendre avant d'entamer la nouvelle mission. La solution de l'algorithme retourne l'ensemble de combinaisons d'affectations possibles triées par S_{cr}, et qui laissent la possibilité au régulateur de choisir l'affectation qui lui convient le plus.

4.3 Intégration des demandes des clients

L'algorithme ALNS a été testé chez quelques clients avec +3000 demandes traitées par jour.

Durant les essais chez les clients de Lomaco-OnLine, plusieurs rectifications des algorithmes ont été demandées. Parmi ces demandes :

- Gestion des différents points d'embauche et de débauche.
- Critère tarifaire des choix du type de véhicule.
- Minimisation du nombre de ressources pour le planning de la veille pour le lendemain.
- Donner la priorité aux véhicules fortement kilométrés s'il s'agit des courtes distances.

Malgré le souci d'avoir un algorithme robuste à différentes utilisations, certaines modifications ont nécessité des développements significatifs.

4.3.1 Gestion des différents points d'embauche et de débauche

Chez plusieurs clients, un ambulancier peut démarrer ou finir sa journée de travail directement depuis chez lui, sans passer à l'agence.

Comme expliqué dans la section 2.4.2, pour la planification de la veille pour le lendemain ou dans la section 3.6.2.1 pour le temps réel, une requête peut être insérée dans une tournée existante ou en créant une nouvelle tournée. Chaque tournée qui commence et s'achève dans un dépôt unique, la synchronisation géographique des tournées se fait assez simplement. Par contre, si l'ambulancier démarre ou finit sa journée dans des endroits hors dépôt, la synchronisation n'est plus évidente.

En effet, nous pouvons faire correspondre la première ou la dernière tournée d'un ambulancier au premier et au dernier service qui convient. Par contre, le fait d'insérer ou

supprimer la première (resp. la dernière) tournée nécessite une mise à jour d'autres tournées notamment le service de départ (resp. service final) de la première (resp. la dernière) tournée. Ces modifications peuvent affecter la faisabilité des autres tournées hors celles insérées ou supprimées.

Pour gérer ce problème, nous avons rajouté un type d'ambulancier spécial pour ceux qui démarrent ou/et qui finissent leur journée hors dépôt. Ces chauffeurs ne peuvent pas effectuer plusieurs tournées dans la journée, ce qui veut dire que les ambulanciers en question ne changent pas de véhicule ou d'équipage durant la journée. Le changement dans les algorithmes est d'interdire d'insérer une nouvelle tournée pour un équipage qui a au moins un chauffeur de ce type.

Cette solution a été bien testée et approuvée par les clients, vu le faible nombre d'ambulanciers qui ont ce privilège de démarrer et finir leur journée directement chez eux.

4.3.2 Critère tarifaire des choix de type de véhicule

Le critère tarifaire d'une demande de transport sanitaire dépend de plusieurs critères comme montré dans le tableau1.3. Le bénéfice direct utilisé dans la fonction objectif 2.1 n'est qu'une valeur approximative du tarif de facturation et qui ne dépend pas du type de véhicule affecté. Les clients nous ont fait la remarque que les solutions obtenues ne tiennent pas compte de la relation entre le type de véhicule affecté pour un transport assis (VSL ou TAXI) et le kilométrage parcouru entre le point de prise en charge et de dépose. Par exemple, si le kilométrage entre le point de prise en charge et de dépose fait moins de (14 km), la réalisation en VSL est beaucoup plus avantageuse qu'en TAXI (les valeurs changent d'un département à un autre).

Afin de considérer cette nouvelle propriété, nous avons rajouté un nouveau critère à minimiser dans la fonction objectif 2.1 :

$$\sum_{i \in P} \sum_{k \in K} \sum_{j \in J} \Theta_{7,j,k}\ \alpha_{i,j}\ km_i\ Z_{i,k}$$

Avec P est l'ensemble de demandes ; K = ensemble de véhicules ; $Z_{p,k} \in \{0,1\}$ est 1 si la demande p est affectée au véhicule k ;et J représente l'ensemble des intervalles de facturations possibles. Pour chaque intervalle j, ($j = [e_{1,j}, e_{2,j}[$), on associe :

- km_i est le kilométrage entre la prise en charge et dépose de la demande i.
- $\alpha_{i,j} \in \{0,1\}$ avec $\alpha_{i,j} = 1$ si le kilométrage entre la prise en charge et dépose de la demande i km_i appartient à l'intervalle j.
- $\Theta_{7,j,k}$ est la facturation kilométrique d'une demande par le véhicule k dont le kilométrage appartient à l'intervalle j.

4.3.3 Minimiser le nombre de ressources utilisées dans le planning de la veille pour le lendemain

Plusieurs clients estiment qu'un bon planning de la veille pour le lendemain est celui qui minimise le nombre de chauffeurs utilisés pour laisser la place à des éventuelles nouvelles demandes qui peuvent apparaître dans la journée ce qui d'ailleurs a été démontré dans la partie expérimentale du chapitre 3.

Ce critère qui revient à minimiser le nombre de véhicules dans la littérature classique des problèmes de tournés de véhicules, a été facilement intégré dans notre algorithme.

Nous avons ajouté dans la fonction objectif (voir 2.3.5) un nouveau critère $\Theta_8 N$ qui représente le coût d'utilisation d'un chauffeur et N qui représente le nombre de chauffeurs utilisés.

4.3.4 Donner la priorité aux véhicules fortement kilométrés

Il s'agit d'utiliser en premier lieu les véhicules ayant un nombre de kilométrage conséquent et d'affecter les véhicules neufs à des longs trajets.

Ce problème est traité dans un post-traitement. Un programme linéaire en nombres entiers est exécuté sur la solution finale du ALNS. Le but de ce post-traitement est de ré-organiser les affectations des tournées aux véhicules. Soient : $\mathbb{T}$ correspond à l'ensemble de tournées fournies par la solution de ALNS. Pour chaque tournée $t \in \mathbb{T}$, nous avons d_t et f_t qui représentent respectivement l'heure de début et de fin de la tournée ; $\theta_{t,k} \in \{0,1\}$, $\theta_{t,k} = 1$ si le véhicule k peut faire la tournée t (i.e. v détient toutes les capacités de réaliser la tournée et le dépôt de k est le point de début et de fin de la tournée) ; ckm_k correspond au nombre de kilomètres au compteur du véhicule k ; $b_t = 1$ si la tournée t détient un trajet long et 0 sinon, et $x_{kt} \in \{0,1\}$, est une variable de décision qui prend la valeur 1 si le véhicule k est affecté à la tournée t.

$$\min \sum_{k} \sum_{t} (x_{kt}\ b_t\ ckm_v - x_{kt}\ (1 - b_t)\ ckm_k) \tag{4.1}$$

$$\sum_{v} x_{kt} = 1 \quad \forall t \in T \tag{4.2}$$

$$x_{vt} \leq \theta_{t,k} \forall t \in T \quad \forall k \in K \tag{4.3}$$

$$x_{kt} + x_{kt'} \leq 1 \quad \forall k \in K, \forall t \in T, \forall t' \in T : (d_t < d_{t'} < f_t) \tag{4.4}$$

La fonction objectif 4.1 minimise l'utilisation des véhicules les moins kilométrés sur des tournées avec des longs trajets et maximise l'utilisation des véhicules les plus kilométrés sur des tournées avec des courts trajets. La contrainte 4.2 assure l'affectation de chaque tournée à un véhicule, la contrainte 4.3 assure l'affectation des véhicules autorisés sur chaque tournée, et finalement, la contrainte 4.4 interdit l'affectation de deux tournées qui se chevauchent au même véhicule.

4.4 Conclusion

Nous avons présenté dans ce chapitre l'intégration industrielle des algorithmes développés au cours de la thèse. Lomaco-online est une application full web désormais utilisée chez les plus grandes entreprises de transport sanitaire français. Les algorithmes s'intègrent parmi les nombreuses fonctionnalités de LOL : facturation, gestion des heures de travail des ambulanciers, la géolocalisation des flottes de véhicules, une interface régulation et planning qui propose aux régulateurs plusieurs aides visuelles et décisionnels pour une meilleure prise de décision.

Plusieurs tests ont été effectués chez les clients, afin de calibrer le poids des fonctions objectifs pour une meilleure prise de décision. Les solutions trouvées par l'algorithme sont analysées par des régulateurs expérimentés pour voir si l'enchainement des demandes, la construction des équipages et l'affectation sont de bonne qualité ou pas. Durant ces tests, d'autres commentaires ont été formulés par les clients qui nécessitent des modifications sur les algorithmes développés. Parmi ces remontées, quatre ont été traitées : la minimisation du nombre de chauffeurs utilisés, la gestion des différents points d'embauche et de débauche des chauffeurs, le choix du type de véhicule dépend principalement du critère tarifaire et l'utilisation des anciens véhicules en premier pour les trajets courts. Après plusieurs jours de tests, les régulateurs étaient bien satisfaits des résultats fournis par l'algorithme. Ces derniers ont conclu que l'algorithme réalise des enchainements satisfaisants. De plus, ils ont conclu que l'exécution de ces enchainements est difficilement réalisable par l'être humain.

Les perspectives pour les mois à venir sont :

— Finir l'intégration de l'algorithme de la planification en temps réel ;

— Tester et calibrer l'algorithme par rapport aux demandes des clients ;

— Intégrer des éventuelles demandes des clients ;

— Intégrer dans les algorithmes (planification de la veille et en temps réel) des temps de trajets qui se mettent à jour en fonction de l'infotrafic.

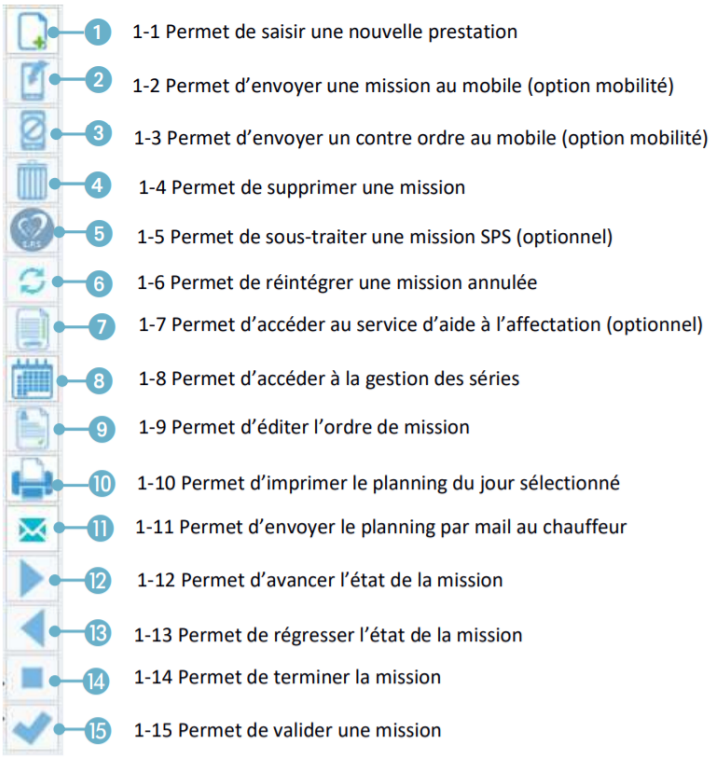

FIGURE 4.3 – Barre de commande de l'interface régulation

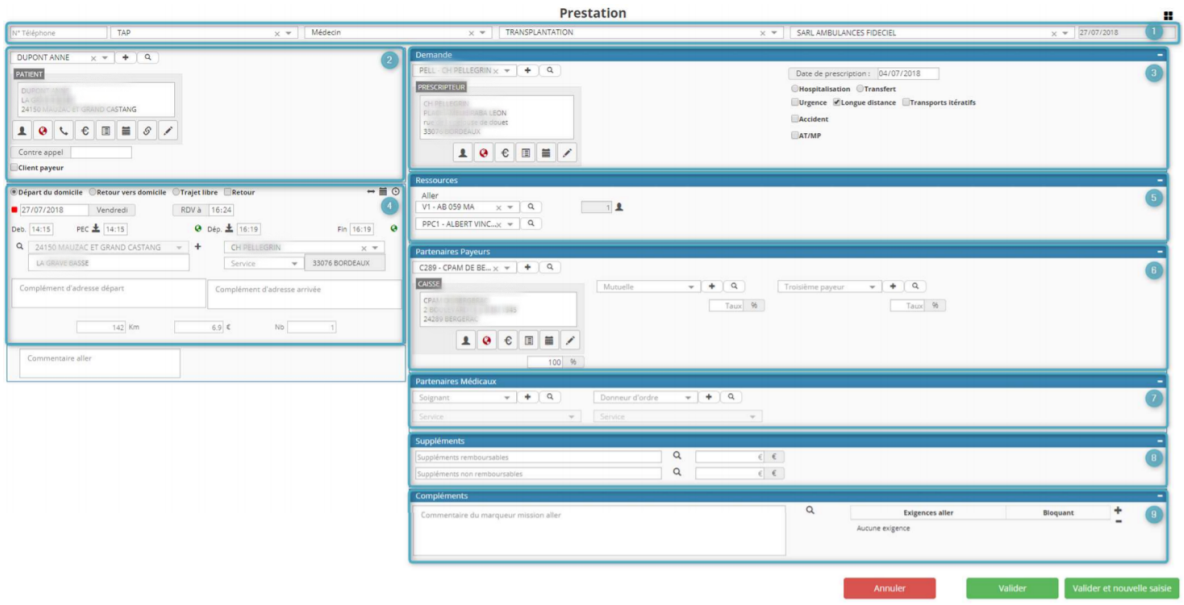

FIGURE 4.4 – Saisie des données d'un patient

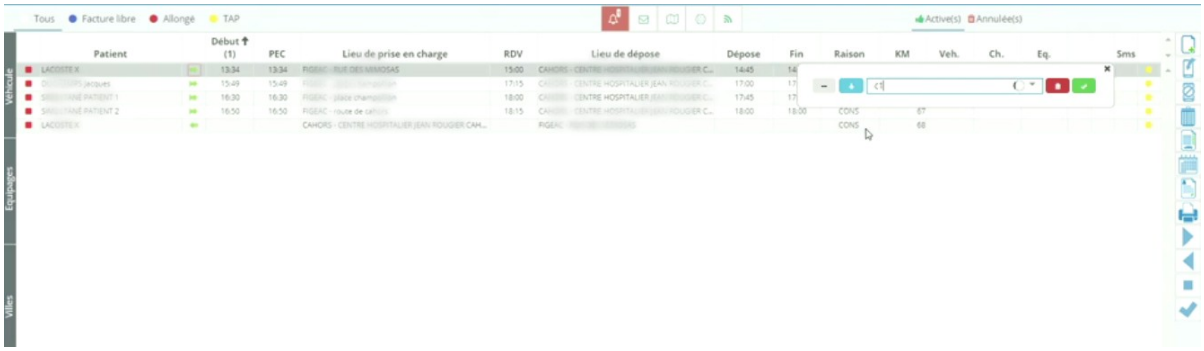

FIGURE 4.5 – Affectation manuelle d'une requête de transport

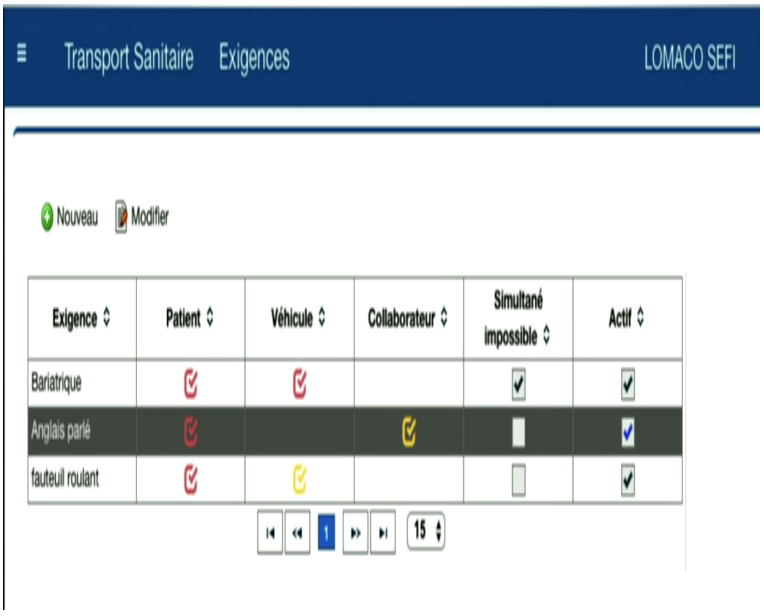

FIGURE 4.6 – Types d'exigence

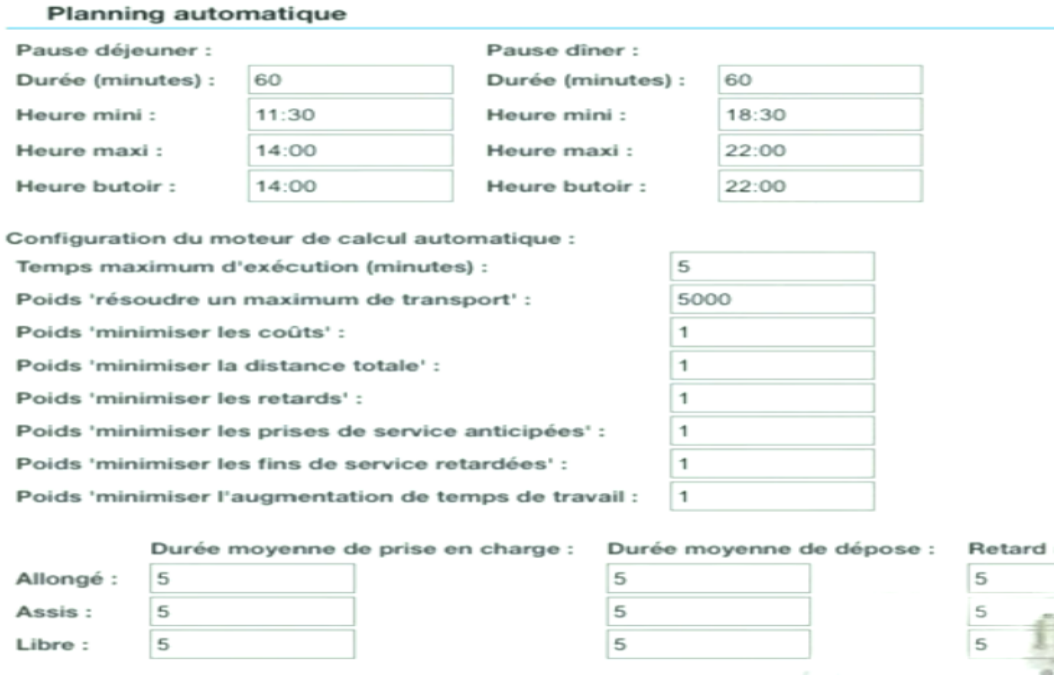

FIGURE 4.7 – Paramétrage de la planification automatique

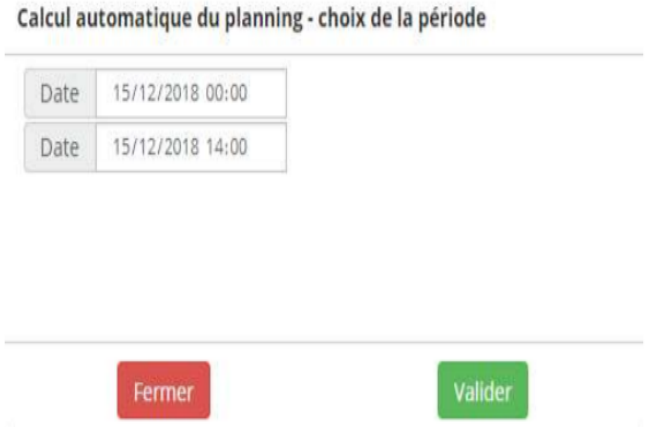

FIGURE 4.8 – Paramétrage de la plage horaire de calcul

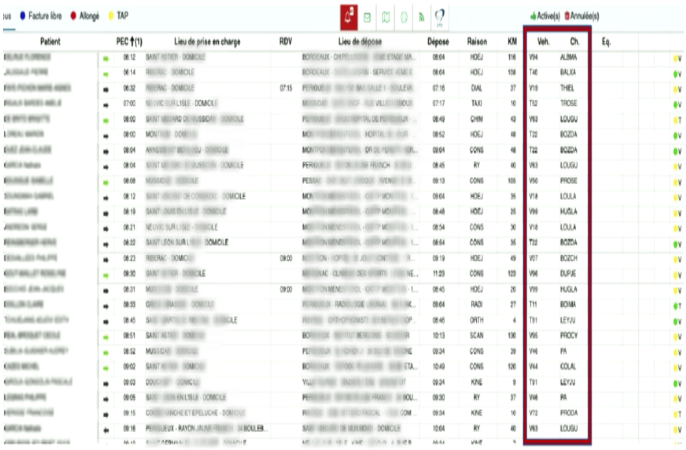

FIGURE 4.9 – Affectation automatique

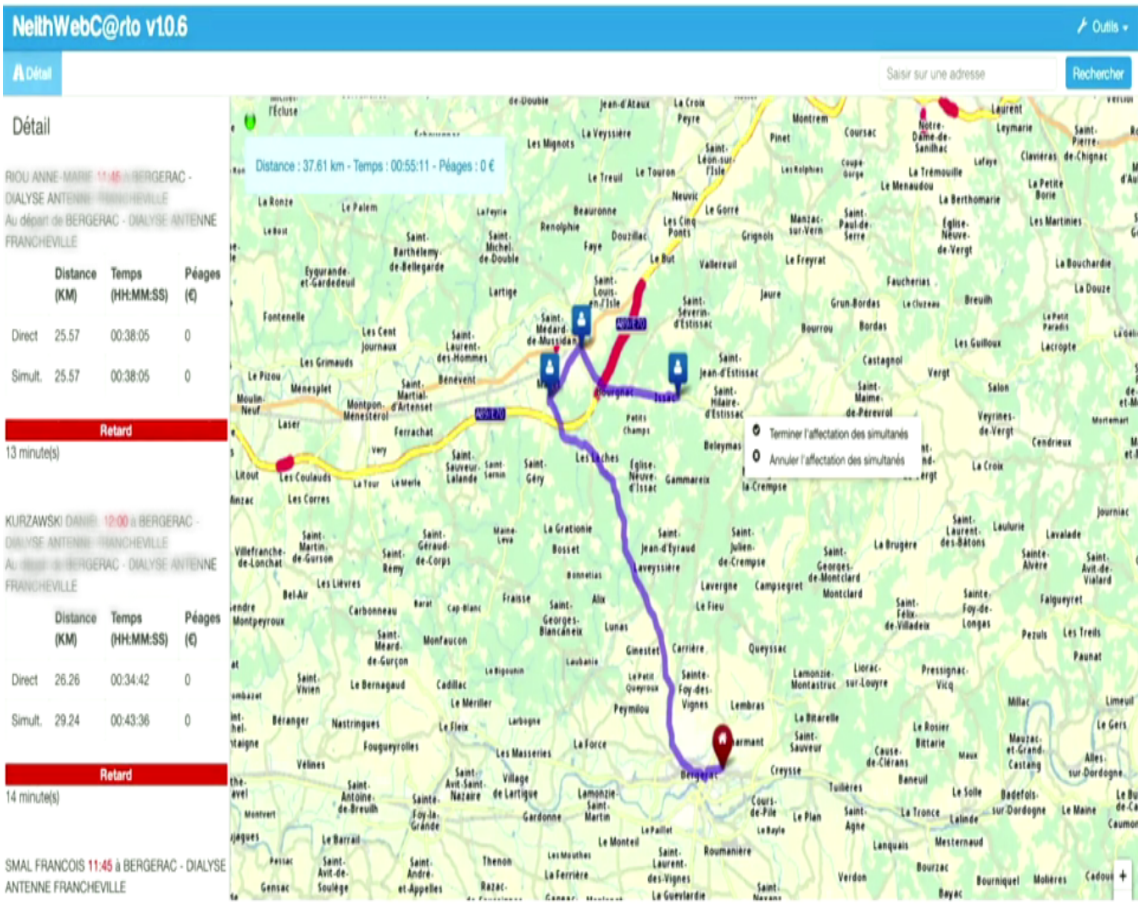

FIGURE 4.10 – Capture LOL aides aux simultanés

Chapitre 5

Synthèse et Conclusion

Depuis quelques années, le marché du transport sanitaire est en pleine croissance économique. Toutefois, le manque de productivité empêche certaines entreprises de transport sanitaire à faire face à cette croissance. Le manque d'efficacité s'explique, essentiellement, par le fait que 40% des trajets réalisés sont à vide (non facturables et à la charge de l'entreprise car ils ne concernent pas le déplacement des patients), et souvent les trajets des ambulanciers sont mal planifiés (cela génère une sous ou suractivité). Face à cette expansion et ces difficultés, les entreprises de transport sanitaire se retrouvent obligées d'adapter leur organisation. Pour accompagner ces entreprises, plusieurs éditeurs de logiciels ont investi dans la recherche et le développement afin de leur proposer des solutions qui répondent à leurs besoins. La société Lomaco est l'une des pionnières à proposer à ces clients une panoplie d'outils innovants.

En collaboration avec la société Lomaco, cette thèse s'intéresse au module relatif à la régulation des tournées de véhicules qui permet d'affecter les demandes de transport sanitaire (ambulance, VSL, taxi, ...) aux équipages de la société (véhicules et ambulanciers). Une partie de ces demandes est connue la veille pour le lendemain. Le reste est connu au fil de la journée.

L'objet de cette thèse est le développement de méthodes d'optimisation pour la planification proactive et réactive des plans de routes d'un grand réseau d'ambulances. Les solutions proactives correspondent au plan de route du matin qui planifie l'heure de début et de fin du travail des chauffeurs ainsi que les premiers trajets à réaliser. Quant aux solutions réactives, elles correspondent à l'intégration, au plan courant, des nouvelles requêtes qui proviennent au fur et à mesure durant la journée.

Les travaux de cette thèse portent sur 3 axes principaux :

Planification de la veille pour le lendemain (proactive) : Le problème consiste à établir un premier planning aux ambulanciers élaboré à partir des demandes connues. Pour résoudre ce problème, nous avons établi des méthodes de résolution statique qui vont être exécutées pour un scénario complet généré à partir des données historiques.

La solution du problème est un planning qui consiste à affecter les ressources (Véhicules et ambulanciers) aux demandes tout en minimisant les déviations des horaires de travail des ambulanciers et les coûts de transport et de maximiser les bénéfices relatifs aux demandes.

La grosse difficulté du problème est la synchronisation entre les tournées, les ambulanciers et véhicules. En effet, plusieurs ambulanciers peuvent être affectés à une tournée, et un ambulancier et véhicule peuvent être affectés à plusieurs tournées dans la journée. D'autres contraintes sont prises en compte dans le problème alors qu'aucun travail scientifique ne les a traités auparavant (affectation des pauses aux ambulanciers avec des fenêtres de temps, considération des pauses au dépôt ou à l'extérieur, minimisation des déviations de temps de travail des ambulanciers, ...)

Pour la résolution du problème, nous avons développé une métha-heuristique du grand voisinage "Adaptive Large Neighberhood Search" (ALNS), à laquelle nous avons adapté, modifié et rajouté plusieurs mécanismes pour l'utiliser en fonction de notre problème.

L'algorithme ALNS utilise une procédure de planification manuelle (HLA) pour la construction de la solution initiale. L'algorithme ALNS a été comparé avec l'algorithme HLA^+ qui est la version améliorée de HLA grâce à une recherche locale. Les résultats montrent que l'algorithme ALNS améliore bien la solution initiale et dépasse la solution obtenue avec l'algorithme HLA^+. L'algorithme ALNS a été testé sur des problèmes connus dans la littérature (MTPTW-MP) et(MTPTW). Les résultats ont montré que l'algorithme ALNS réalise des solutions très compétitives par rapport aux travaux existants dans la littérature.

Planification en temps réel (réactive) : Le problème consiste à réguler automatiquement et en temps réel les demandes de transport sanitaire. Le problème à été décortiqué en plusieurs sous-problèmes : (P_1 DARP-dynamique, P_2 DARP-dynamique avec temps de travail des ambulanciers, P_3 DARP-dynamique avec affectation des pauses aux ambulanciers, P_4 DARP-dynamique avec plusieurs types de véhicules et P_5 le problème complet qui regroupe toutes les contraintes). Pour résoudre le problème nous avons proposé 3 stratégies d'insertion :

- $Strategie_1$, algorithme glouton d'insertion, avec F comme fréquence de re-planification ;
- $Strategie_2$ (Méthode de petit pas) algorithme d'insertion avec un PLNE, l'insertion ne se fait qu'à la fin des tournées et la re-planification se fait à chaque fois qu'un chauffeur termine une course ;
- $Strategie_3$ (Méthode d'anticipation) algorithme glouton d'insertion : l'insertion ne se fait qu'à la fin des tournées avec F comme fréquence de re-planification.

Les trois stratégies ont été comparées sur les trois premiers problèmes, avec plusieurs paramétrages de H_{fix} , H_{libre} et F. Les résultats ont montré que la $strategie_1$ est la stratégie efficace, dans plus de 93% des cas en moyenne. Ce qui implique que la stratégie

1 serait normalement la stratégie la plus efficace pour le problème général P_5.

La deuxième série de tests se concentre exclusivement sur le problème général P_5. Le but de ces tests est de comparer la $strategie_1$ avec les bornes à informations parfaites. La solution de $strategie_1$ dans un environnement dynamique à été comparée avec deux bornes à savoir :

— Solution en utilisant le planning de la veille pour le lendemain dans le contexte où toutes les informations sont révélées au début de la journée.

— Solution en utilisant $strategie_1$ dans le contexte où toutes les informations sont révélées au début de la journée.

Les résultats ont montré que le fait de savoir toutes les informations à l'avance améliore la qualité de la solution entre 33 à 98%. Le grand écart entre les deux bornes à informations parfaites montre que la $strategie_1$ est beaucoup moins efficace que l'algorithme ALNS, la chose qui est normale en comparaison avec le temps de calcul de chaque algorithme.

La dernière série de tests traite le problème complet de la thèse (partie proactive et partie réactive). Autrement dit, nous étudions l'impact du planning de la veille pour le lendemain dans les décisions prises en temps réel. Les résultats montrent que l'utilisation du planning de la veille pour le lendemain à un impact positif sur les décisions du jour même. Une amélioration est de $6,9\%$ en moyenne si l'algorithme de la veille est utilisé sur un scénario complet et encore une amélioration de 3% en moyenne si l'algorithme de la veille minimise le nombre de chauffeurs utilisés.

Implémentation industrielle : Dans ce chapitre, nous avons présenté l'application Lomaco-OnLine (LOL), ses fonctionnalités et comment les algorithmes de la thèse ont été implémentés. Des tests ont été effectués chez les clients afin de calibrer les poids des objectifs pour une meilleure prise de décision. Durant ces tests, des commentaires ont été formulés par les clients, ce qui a nécessité des modifications sur les algorithmes développés.

Le travail de cette thèse a bien répondu aux attentes de l'entreprise Lomaco et à l'objectif qui était fixé au tout début de la thèse avec un algorithme unique qui traite les cas de plusieurs clients finaux aux profils variés et aux exigences pointues. Néanmoins plusieurs perspectives doivent être exploitées pour bien affiner les algorithmes déjà développés :

Perspectives :

La première perspective sur laquelle nous allons commencer à travailler est la génération d'un ou plusieurs scénarios complets à partir des informations connues la veille. Comme montré dans les résultats du Chapitre 3, le fait de générer un scénario complet proche de la réalité améliore les décisions de la planification en temps réel. L'étude des données historiques des clients est la première étape sur laquelle nous allons nous pencher pour voir s'il y a des jours de la semaine où les réalisations se ressemblent, des endroits très

sollicités ... L'apport de techniques d'analyse de données et d'apprentissage peut être intéressant pour traiter cet aspect.

Caractériser des aspects stochastiques de ce problème peut permettre d'améliorer la robustesse du plan initial, le résoudre par rapport à plusieurs scénarios, mais comment agréger des solutions différentes reste un challenge. Traiter le problème directement comme un problème stochastique est aussi difficile vu la taille déjà importante des instances à traiter.

Des pistes sont déjà explorées avec des requêtes partiellement déterminées (ex : requêtes Retour) et des solutions prometteuses (ex : avoir des ambulances disponibles à l'heure des sorties massives de l'hôpital, voir des équipages totalement libres pour mieux accepter la demande plutôt que d'équilibrer la charge).

Une autre piste d'amélioration est de considérer des approches de division des espaces temporels et géographiques pour traiter des sous-problèmes de plus petite taille. Le challenge ici est de parvenir à optimiser les ressources partagées aux interfaces et les longs trajets. En effet un véhicule envoyé dans un secteur lointain peut travailler localement puis revenir à sa base de départ. À ce jour, ce type de comportement est géré en déléguant la responsabilité d'un équipage à une nouvelle régulation pendant un certain temps.

D'un point de vue de possibles améliorations locales aux algorithmiques déjà développés, deux axes de travail sont en cours d'étude avec Lomaco.

La prise de la pause déjeuner/diner se fait généralement au dépôt avec une durée d'une heure, le régulateur peut réellement en décider autrement (i.e. d'attribuer une pause sur place ou avec une durée moins d'une heure), en indemnisant l'ambulancier concerné. L'ajout d'autres objectifs qui dépendent des positions et des durées des pauses est l'une des prochaines perspectives pour l'amélioration des algorithmes développés (Veille pour le lendemain ou temps réel).

De plus en plus, un ambulancier peut être attribué à plusieurs dépôts dans la journée. Autrement dit, chaque ambulancier e est attaché à un ensemble de dépôts possibles De_e. L'ambulancier e peut démarrer sa journée dans un endroit $v_{1,e} \in D_e$, finir la journée dans un endroit $v_{2,e} \in De_e$ et chaque changement de véhicule i peut se faire dans un endroit $v_{3_i,e} \in De_e$. Les endroits $v_{1,e}, v_{1,e}$ et $v_{3_i,e}$ peuvent être différents. La considération de cette contrainte est plus ou moins facile à implémenter dans la stratégie d'insertion en temps réel. Par contre pour le planning de la veille pour le lendemain, elle nécessite un grand changement de l'algorithme ALNS à cause des opérateurs d'insertion et de suppression.

Bibliographie

[Ameli.fr, a] Ameli.fr. Tarifs conventionnels ambulances. https ://www.ameli.fr/loire/transporteur-sanitaire/exercice-professionnel/facturation/tarifs/ambulances-tarifs-conventionnels.

[Ameli.fr, b] Ameli.fr. Tarifs conventionnels vsl. https ://www.ameli.fr/loire/transporteur-sanitaire/exercice-professionnel/facturation/tarifs/vsltarifsconventionnels.

[Andrew et al., 2020] Andrew, E., Nehme, Z., Cameron, P., and Smith, K. (2020). Drivers of increasing emergency ambulance demand. *Prehospital Emergency Care*, 24(3) :385. PMID : 31237460.

[Assurance-Maladie, 2020] Assurance-Maladie (2020). Améliorer la qualité du système de santé et matrîser les déponses. https ://assurance-maladie.ameli.fr/sites/default/files/2020-07_rapport-propositions-pour-2021_assurance-maladie_1.pdf.

[Attanasio et al., 2004] Attanasio, A., Cordeau, J.-F., Ghiani, G., and Laporte, G. (2004). Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing*, 30(3) :377–387.

[Azi et al., 2014] Azi, N., Gendreau, M., and Potvin, J.-Y. (2014). An adaptive large neighborhood search for a vehicle routing problem with multiple routes. *Computers & Operations Research*, 41 :167–173.

[Baldacci et al., 2008] Baldacci, R., Christofides, N., and Mingozzi, A. (2008). An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Math. Program.*, 115 :351–385.

[Beaudry et al., 2010] Beaudry, A., Laporte, G., Melo, T., and Nickel, S. (2010). Dynamic transportation of patients in hospitals. *OR Spectrum*, 32 :77–107.

[Benyahia and Potvin, 1998] Benyahia, I. and Potvin, J.-Y. (1998). Decision support for vehicle dispatching using genetic programming. *IEEE Transactions on Systems, Man, and Cybernetics - Part A : Systems and Humans*, 28(3) :306–314.

[Berbeglia et al., 2010] Berbeglia, G., Cordeau, J.-F., and Laporte, G. (2010). Dynamic pickup and delivery problems. *European journal of operational research*, 202(1) :8–15.

[Cattaruzza et al., 2016] Cattaruzza, D., Absi, N., and Feillet, D. (2016). Vehicle routing problems with multiple trips. *4OR*, 14(3) :223–259.

[Chang et al., 2003] Chang, M.-S., Chen, S.-r., and Hsueh, C.-F. (2003). Real-time vehicle routing problem with time windows and simultaneous delivery/pickup demands. *Journal of the Eastern Asia Society for Transportation Studies*, 5 :2273–2286.

[Cheung et al., 2008] Cheung, B., Choy, K., Li, C.-L., Shi, W., and Tang, J. (2008). Dynamic routing model and solution methods for fleet management with mobile technologies. *International Journal of Production Economics*, 113 :694–705.

[Cordeau and Laporte, 2003] Cordeau, J.-F. and Laporte, G. (2003). The dial-a-ride problem (darp) : Variants, modeling issues and algorithms. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1(2) :89–101.

[Coslovich et al., 2006] Coslovich, L., Pesenti, R., and Ukovich, W. (2006). A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. *European Journal of Operational Research*, 175(3) :1605–1615.

[Crainic et al., 2009] Crainic, T. G., Crisan, G. C., Gendreau, M., Lahrichi, N., and Rei, W. (2009). A concurrent evolutionary approach for rich combinatorial optimization. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference : Late Breaking Papers*, GECCO '09, pages 2017–2022, New York, NY, USA. Association for Computing Machinery.

[Drexl, 2012] Drexl, M. (2012). Synchronization in Vehicle Routing—A Survey of VRPs with Multiple Synchronization Constraints. *Transportation Science*, 46(3) :297–316.

[Gehring and Homberger, 1999] Gehring, H. and Homberger, J. (1999). A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows. In *Proceedings of EUROGEN99*, volume 2, pages 57–64. Citeseer.

[Hernandez et al., 2016] Hernandez, F., Feillet, D., Giroudeau, R., and Naud, O. (2016). Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows. *European Journal of Operational Research*, 249(2) :551–559.

[Ho et al., 2018] Ho, S. C., Szeto, W., Kuo, Y.-H., Leung, J. M., Petering, M., and Tou, T. W. (2018). A survey of dial-a-ride problems : Literature review and recent developments. *Transportation Research Part B : Methodological*, 111 :395–421.

[Karami et al., 2018] Karami, F., Vancroonenburg, W., and Vanden Berghe, G. (2018). A dynamic scheduling approach to internal hospital logistics. In *International Workshop on Dynamic Scheduling Problems, Date : 2018/06/26-2018/06/28, Location : Poznań, Poland*.

[Kergosien et al., 2013] Kergosien, Y., Lenté, C., Billaut, J.-C., and Perrin, S. (2013). Metaheuristic algorithms for solving two interconnected vehicle routing problems in a hospital centre. *Computers & Operations Research*, 40 :2508–2518.

[Kilby et al., 1998] Kilby, P., Prosser, P., and Shaw, P. (1998). Dynamic vrps : A study of scenarios. *University of Strathclyde Technical Report*, 1(11).

[KPMG, 2019] KPMG (2019). Observatoire du transport sanitaire 2019 : dernières évolutions. https ://home.kpmg/fr/fr/home/insights/2019/12/observatoire-activite-entreprises-transport-sanitaire.html.

[Laporte and Semet, 2002] Laporte, G. and Semet, F. (2002). Classical heuristics for the capacitated vrp. In *The vehicle routing problem*, pages 109–128. SIAM.

[Lim et al., 2017] Lim, A., Zhang, Z., and Qin, H. (2017). Pickup and delivery service with manpower planning in hong kong public hospitals. *Transportation Science*, 51(2) :688–705.

[Liu et al., 2015] Liu, M., Luo, Z., and Lim, A. (2015). A branch-and-cut algorithm for a realistic dial-a-ride problem. *Transportation Research Part B : Methodological*, 81 :267–288.

[LivreBlanc, 2018] LivreBlanc (2018). Chambre nationale des services d'ambulances. https ://www.cnsa-ambulances.com/wp-content/uploads/2019/05/Livre_blanc_CNSA.pdf.

[Luo et al., 2019] Luo, Z., Liu, M., and Lim, A. (2019). A two-phase branch-and-price-and-cut for a dial-a-ride problem in patient transportation. *Transportation Science*, 53(1) :113–130.

[Maligorne and Marcadé, 2018] Maligorne, C. and Marcadé, A. (2018). Les ambulanciers en colère bloquent le centre de paris. https ://www.lefigaro.fr/social/2018/12/03/20011-20181203ARTFIG00063-les-ambulanciers-protestent-contre-une-refonte-du-financement-des-transports-sanitaires.php.

[Markovic et al., 2015] Markovic, N., Nair, R., Schonfeld, P., Miller-Hooks, E., and Mohebbi, M. (2015). Optimizing dial-a-ride services in maryland : Benefits of computerized routing and scheduling. *Transportation Research Part C : Emerging Technologies*, 55 :156–165. Engineering and Applied Sciences Optimization (OPT-i) - Professor Matthew G. Karlaftis Memorial Issue.

[Mitrovic-Minic et al., 2004] Mitrovic-Minic, S., Krishnamurti, R., and Laporte, G. (2004). Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B : Methodological*, 38(8) :669–685.

[Parragh et al., 2009] Parragh, S., Cordeau, J.-F., Doerner, K., and Hartl, R. (2009). Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints. *OR Spectrum*, 34 :1–41.

[Pillac et al., 2013] Pillac, V., Gendreau, M., Guéret, C., and Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1) :1–11.

[Pisinger and Ropke, 2005] Pisinger, D. and Ropke, S. (2005). A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34 :2403–2435.

[Psaraftis, 1980] Psaraftis, H. N. (1980). A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science*, 14(2) :130–154.

[Psaraftis et al., 2016] Psaraftis, H. N., Wen, M., and Kontovas, C. A. (2016). Dynamic vehicle routing problems : Three decades and counting. *Networks*, 67(1) :3–31.

[Santé-Mobilité-Services, 2017] Santé-Mobilité-Services (2017). Du transport sanitaire à la mobilité du patient. https ://www.transdev.com/wp-content/uploads/2018/10/Livre-Blanc-Santee-Mobilite.pdf.

[Schilde et al., 2011] Schilde, M., Doerner, K., and Hartl, R. (2011). Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. *Computers & Operations Research*, 38(12) :1719–1730.

[Schyns, 2015] Schyns, M. (2015). An ant colony system for responsive dynamic vehicle routing. *European Journal of Operational Research*, 245.

[Shaw, 1997] Shaw, P. (1997). A new local search algorithm providing high quality solutions to vehicle routing problems. *APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK.*

[Swihart and Papastavrou, 1999] Swihart, M. R. and Papastavrou, J. D. (1999). A stochastic and dynamic model for the single-vehicle pick-up and delivery problem. *European Journal of Operational Research*, 114(3) :447–464.

[Sörensen, 2013] Sörensen, K. (2013). Metaheuristics – the metaphor exposed. *International Transactions of Operations Research*, In Press.

[Téllez et al., 2018] Téllez, O., Vercraene, S., Lehuédé, F., Péton, O., and Monteiro, T. (2018). The fleet size and mix dial-a-ride problem with reconfigurable vehicle capacity. *Transportation Research Part C : Emerging Technologies*, 91 :99–123.

[Vallee et al., 2020] Vallee, S., Oulamara, A., and Cherif-Khettaf, W. R. (2020). New online reinsertion approaches for a dynamic dial-a-ride problem. *Journal of Computational Science*, 47 :101199.

[Vidal et al., 2014] Vidal, T., Crainic, T. G., Gendreau, M., and Prins, C. (2014). A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research*, 234 :658–673.

[Zhang et al., 2015] Zhang, Z., Liu, M., and Lim, A. (2015). A memetic algorithm for the patient transportation problem. *Omega*, 54 :60–71.

Annexe A

Lunch break counterexample

TABLE A.1 – Computation details of the counterexample

σ	σ'	$\sigma \oplus \sigma'$
$E(\sigma) = 150$	$E(\sigma') = 160$	$E(\sigma \oplus \sigma') = 160$
$L(\sigma) = 20$	$L(\sigma') = 170$	$L(\sigma \oplus \sigma') = 20$
$F(\sigma) = True$	$F(\sigma') = True$	$F(\sigma \oplus \sigma') = True$
$T(\sigma) = 30$	$T(\sigma') = 0$	$T(\sigma \oplus \sigma') = 40$
		$E'_{case1} = 160$
		$E'_{case2} = 220$
		$E'_{case3} = +\infty$
		$L'_{case1} = 20$
		$L'_{case2} = 20$
		$L'_{case3} = -\infty$
		$F'_{case1} = True$
		$F'_{case2} = False$
		$F'_{case3} = False$
$E'(\sigma) = 150$	$E'(\sigma') = +\infty$	$E'(\sigma \oplus \sigma') = 160$
$L'(\sigma) = 20$	$L'(\sigma') = -\infty$	$\underline{L'(\sigma \oplus \sigma') = 20}$
$F'(\sigma) = True$	$F'(\sigma') = False$	$F'(\sigma \oplus \sigma') = True$
$T'(\sigma) = 90$	$T'(\sigma') = 60$	$T'(\sigma \oplus \sigma') = 100$

Annexe B

Results tables

TABLE B.1 – ALNS-Best

Instance	Ob.V	U.R	Distance	Ealier	Later	Longer	Loss
A-10-1	11729,4	1	4885,2	1200	61	372	211,2
A-10-2	4286,6	0	3068,6	911	111	196	0,0
A-10-3	9756,4	1	3782,5	881	0	0	92,9
A-10-4	3882,6	0	3047,6	533	302	0	0,0
A-10-5	7477,2	0	4813,2	1615	248	801	0,0
A-06-1	5208,6	0	4848,6	358	2	0	0,0
A-06-2	7969,8	0	5628,8	1091	839	411	0,0
A-06-3	11821,2	1	4501,2	1141	421	568	190,0
A-06-4	6197,7	0	5016,7	1155	1	25	0,0
A-25-1	16253,6	1	7327,9	2293	497	1048	87,6
A-25-2	5763,6	0	5166,1	589,5	2	6	0,0
A-25-3	11312,3	0	8591,1	1993,52	66,84	660,84	0,0
A-61-1	72466,6	2	42348,8	11254	2475	6250	138,9
A-61-2	97542,3	1	56338,9	20128	5329	10666	80,4
A-61-3	112216,0	3	62879,0	18303	5191	10730	113,0
A-61-4	67020,9	1	37471,3	13894	2653	7960	42,6
A-61-5	62867,3	1	40693,6	10893	1376	4824	80,7

TABLE B.2 – ALNS avg

Instance	Ob.V	U.R	Distance	Ealier	Later	Longer	Loss
A-10-1	13562,8	1,3	4763,2	1139,0	181,3	480,0	332,7
A-10-2	4415,7	0,0	3142,4	934,0	109,0	230,3	0,0
A-10-3	9917,4	1,0	3747,5	937,0	0,0	140,0	92,9
A-10-4	3954,7	0,0	3110,7	672,3	171,7	0,0	0,0
A-10-5	8183,1	0,0	4730,4	2464,7	242,7	745,3	0,0
A-06-1	8850,7	0,7	4911,6	261,0	66,3	88,0	190,4
A-06-2	8058,3	0,0	5621,6	1170,7	845,0	421,0	0,0
A-06-3	12032,1	1,0	4484,8	1249,0	430,0	678,3	190,0
A-06-4	9738,9	0,7	5076,9	928,3	146,7	174,7	79,0
A-25-1	16319,5	1,0	7322,2	2302,0	493,7	1114,0	87,6
A-25-2	5947,5	0,0	5312,5	624,2	1,0	9,8	0,0
A-25-3	12886,5	0,3	8382,1	2037,3	65,0	717,3	18,0
A-61-1	75444,5	2,3	43024,4	11324,7	2714,0	6571,7	143,0
A-61-2	103783,6	2,3	57848,2	18959,3	4900,3	10250,0	159,1
A-61-3	117713,5	3,3	64187,9	20136,3	5380,3	11199,7	142,5
A-61-4	67471,5	1,3	36916,8	13981,7	2412,0	7427,3	67,0
A-61-5	66815,7	1,3	41290,8	11225,0	2049,0	5481,0	103,1

Table B.3 – HLA

Instance	Ob.V	U.R	Distance	Ealier	Later	Longer	Loss
A-10-1	74414,7	13	6196,5	1045	396	865	912,2
A-10-2	32139,2	5	4222,2	1146	267	984	520,0
A-10-3	59459,4	10	5862,8	1266	525	1030	775,6
A-10-4	30765,1	5	4115,4	914	37	426	272,8
A-10-5	69285,3	12	5616,2	1577	228	1277	587,1
A-06-1	70820,7	12	7166,0	830	831	1264	729,7
A-06-2	29200,3	4	6056,7	1305	592	859	387,6
A-06-3	111758,9	20	6014,2	2596	421	1739	988,7
A-06-4	73995,7	13	6127,9	1365	4	667	831,8
A-25-1	21977,2	2	7628,0	2749	441	964	195,2
A-25-2	8365,6	0	6409,4	1207,1	67,5	681,6	0,0
A-25-3	45705,9	6	11325,1	2413,96	0	1521,16	445,6
A-61-1	201746,0	21	65631,4	14574	3247	11674	1619,6
A-61-2	248580,9	23	85058,6	21908	5958	18510	2146,2
A-61-3	395711,3	48	100628,6	22661	7819	20832	3770,8
A-61-4	160841,1	13	58558,4	16920	4800	14648	914,7
A-61-5	225313,4	25	68119,2	15413	2446	12289	2046,2

Table B.4 – HLA+

Instance	Ob.V	U.R	Distance	Ealier	Later	Longer	Loss
A-10-1	39602,3	6	6017,2	1146,0	543,0	1267,0	629,1
A-10-2	11307,3	1	4011,5	1092,0	279,0	821,0	103,8
A-10-3	18309,0	2	5889,9	1377,0	350,0	647,0	45,0
A-10-4	10496,2	1	3970,4	1006,0	0,0	441,0	78,8
A-10-5	39162,4	6	5721,7	1637,0	228,0	1271,0	304,6
A-06-1	30202,6	4	6685,4	554,0	1344,0	1417,0	202,2
A-06-2	22770,2	3	5248,3	1043,0	585,0	575,0	318,9
A-06-3	49467,0	8	5319,8	2068,0	564,0	1074,0	441,2
A-06-4	57673,0	10	5399,2	1118,0	0,0	417,0	738,8
A-25-1	21010,0	2	7260,8	2693,0	268,0	593,0	195,2
A-25-2	7595,1	0	6060,1	982,5	1,0	551,5	0,0
A-25-3	20425,8	1	10859,9	2891,4	0,0	1631,8	42,8
A-61-1	152596,9	12	62301,8	14187,0	3328,0	11835,00	945,1
A-61-2	203168,1	14	84038,4	22277,0	6353,0	19243,00	1256,7
A-61-3	274819,6	26	96284,0	22367,0	6266,0	17762,00	262140,6
A-61-4	108297,6	4	54265,6	17003,0	3654,0	13173,00	202,0
A-61-5	159576,0	12	67316,9	15205,0	3250,0	12957,00	847,1

TABLE B.5 – ALNS 1-trip

Instance	Ob.V	U.R	Distance	Ealier	Later	Longer	Loss
A-10-1	17330,8	2	4419,1	1226,0	429,0	681,0	575,6
A-10-2	4875,9	0	3137,9	1398,0	101,0	239,0	0,0
A-10-3	10244,1	1	3914,2	1036,0	0,0	201,0	92,9
A-10-4	4960,1	0	3221,1	1247,0	11,0	481,0	0,0
A-10-5	7803,0	0	4995,0	1583,0	262,0	963,0	0,0
A-06-1	11447,2	1	5240,1	648,0	45,0	431,0	83,1
A-06-2	8466,7	0	5665,7	901,0	1152,0	748,0	0,0
A-06-3	14323,4	1	4293,4	1668,0	1860,0	1312,0	190,0
A-06-4	11067,6	1	5066,0	847,0	8,0	28,0	118,6
A-25-1	16345,6	1	7339,0	2409,0	484,0	1026,0	87,6
A-25-2	6273,3	0	5227,3	825,5	174,0	46,5	0,0
A-25-3	11982,1	0	8797,3	2843,3	5,0	336,5	0,0
A-61-1	90578,1	5	44557,8	12089,0	3072,0	5591,0	268,3
A-61-2	115585,8	4	56673,6	21266,0	4683,0	12721,0	242,1
A-61-3	143926,3	7	62072,4	24434,0	7108,0	15005,0	307,0
A-61-4	75874,6	3	36703,4	13923,0	3154,0	6954,0	140,2
A-61-5	74108,8	3	39532,8	10842,0	3146,0	5394,0	194,0

Table B.6 – ALNS BOW

Instance	Ob.V	U.R	Distance	Ealier	Later	Longer	Loss
A-10-1	11168,9	1	4546,7	1021,0	54,0	336,0	211,2
A-10-2	4476,7	0	2963,7	1209,0	30,0	274,0	0,0
A-10-3	9777,1	1	3622,1	961,0	0,0	101,0	92,9
A-10-4	4158,0	0	3070,0	624,0	304,0	160,0	0,0
A-10-5	7556,3	0	4696,3	2261,0	98,0	501,0	0,0
A-06-1	10172,5	1	4926,8	140,0	26,0	12,0	67,8
A-06-2	8114,9	0	5401,9	1082,0	940,0	691,0	0,0
A-06-3	12307,6	1	4066,6	1709,0	888,0	454,0	190,0
A-06-4	10689,3	1	4674,7	790,0	40,0	66,0	118,6
A-25-1	15906,4	1	7025,8	2305,0	462,0	1026,0	87,6
A-25-2	6019,2	0	5367,7	604,0	9,5	38,0	0,0
A-25-3	13591,5	0	8246,8	2371,1	1359,5	1614,0	0,0
A-61-1	65471,5	1	40978,8	9741,0	3266,0	6405,0	80,7
A-61-2	98196,5	2	58039,7	17149,0	3820,0	9018,0	169,8
A-61-3	104917,1	1	66652,7	18372,0	5425,0	9410,0	57,4
A-61-4	61916,2	0	38770,2	12584,0	3429,0	7133,0	0,0
A-61-5	77309,7	4	40017,8	10667,0	1715,0	4698,0	211,8

TABLE B.7 – ALNS with customers preferences

Instance	Ob.V	U.R	Distance	Ealier	Later	Longer	Loss
A-10-1	27279,0	4	4353,1	1273,0	197,0	439,0	1016,9
A-10-2	9800,6	1	3359,4	1149,0	67,0	218,0	7,1
A-10-3	10641,4	1	4047,5	1033,0	50,0	418,0	92,9
A-10-4	4186,5	0	3225,5	768,0	107,0	86,0	0,0
A-10-5	8968,1	0	5252,1	1845,0	628,0	1243,0	0,0
A-06-1	17423,5	2	5247,3	109,0	953,0	543,0	571,3
A-06-2	8866,5	0	5652,5	1259,0	1149,0	806,0	0,0
A-06-3	22900,5	3	4654,8	2580,0	8,0	389,0	268,7
A-06-4	16242,4	2	5115,1	876,0	0,0	65,0	186,3
A-25-1	16977,6	1	7453,0	2905,0	466,0	1066,0	87,6
A-25-2	5985,1	0	5301,5	675,6	1,0	7,0	0,0
A-25-3	14421,9	0	8744,0	1951,5	1593,5	2132,8	0,0
A-61-1	94293,9	5	45130,2	12538,0	3601,0	7704,0	320,7
A-61-2	143406,1	8	59938,0	22173,0	6215,0	14613,0	467,1
A-61-3	147789,1	6	70530,0	23329,0	8862,0	14770,0	298,2
A-61-4	94668,1	4	41816,9	16179,0	5075,0	11407,0	190,2
A-61-5	90652,2	5	43340,1	12757,0	2150,0	7149,0	256,2

École Nationale Supérieure des Mines

de Saint-Étienne

NNT : 2021LYSEL022

Mohammed SKIREDJ

Proactive and reactive planning of ambulance networks

Specialty: Industrial Engineering

Keywords: Medical transport, dynamic planning, vehicle routing problems, heuristics, metaheuristics, simulation, …

Abstract:

This thesis deals with optimization methods for proactive planning of large ambulance networks and reactive planning to update them during the day. The proactive solutions corresponding to the decisions of the morning, which decide the starting times of the drivers and their first visits. This plan must be robust in the sense of the best acceptance of requests that appear during the day. When new requests are revealed, reactive methods must offer solutions to integrate them into the current planning by balancing the efficiency and the acceptability of future requests.

The principal scientific challenges in the thesis are: (1) the integration of complex constraints and (2) the resolution of a very large problems (several thousand) compared to those treated by the classical approaches. Complex business constraints involve staff recombination with vehicle change, working time management (start and end times, breaks, …).

Regarding the proactive part, a metaheuristic (Adaptive Large Neighborhood Search) with specific procedures to manage the aforementioned locks is proposed. The comparison of our ALNS algorithm with a manual planning method shows the importance of the developed method. Our algorithm is also very competitive with the state of the art of related problems such as Resource Constrained Dial-a-ride Problem, Multi-Trip Vehicle Routing Problem with Time Windows and Pickup-and-Delivery Problem with Time Windows.

As for reactive planning, a general framework of re-optimization strategies is proposed and allows from three parameters establishing different strategies in terms of anticipation, frequency and conservatism. The best configurations found are compared to two manual approaches. Finally, several key elements are analyzed through a large numerical experiments carried out by a specific simulation platform.

Finally, the integration of these algorithms into an operational tool is presented with the first adjustments and future perspectives following the feedback of ambulance dispatchers.

École Nationale Supérieure des Mines

de Saint-Étienne

NNT : 2021LYSEL022

Mohammed SKIREDJ

Planification proactive et réactive des réseaux d'ambulances

Spécialité : Génie industriel

Mots clefs : Transport sanitaire, Planification dynamique, Tournées de véhicules, heuristiques, Métaheuristiques, Simulation, …

Résumé :

Cette thèse traite les méthodes d'optimisation pour la planification proactive des plans de routes d'un grand réseau d'ambulances et celle réactive pour leur mise à jour durant la journée. Les solutions proactives correspondent au plan du matin qui fixe les heures de début de travail des chauffeurs et leurs premières missions. Ce plan doit être robuste afin d'optimiser la gestion des requêtes qui apparaissent au cours de la journée. Lorsque ces dernières sont reçues, les méthodes réactives doivent proposer de les intégrer dans le plan courant en équilibrant l'efficacité et le potentiel d'acceptabilité des futures requêtes.

Les verrous scientifiques abordés dans la thèse portent sur deux points : (1) l'intégration des contraintes métiers complexes et résistantes aux approches existantes de la littérature et (2) la résolution des problèmes de très grandes tailles (plusieurs milliers) par rapport à ceux traités par les approches classiques. Les contraintes métiers complexes comportent la recombinaison d'équipages avec le changement des véhicules et la gestion des temps de travail (les horaires de début et de fin de la journée et les horaires des pauses).

Concernant la partie proactive, une métaheuristique (Adaptive Large Neighborhood Search) avec des procédures spécifiques est proposée afin de gérer les verrous précités. La comparaison de notre algorithme ALNS avec une méthode de planification manuelle montre l'importance de la méthode développée. Notre algorithme se montre aussi très compétitif avec l'état de l'art sur des cas particuliers de notre problème tel que Resource Constrained Dial-a-ride Problem, Multi-Trip Vehicle Routing Problem with Time Windows et Pickup-and-Delivery Problem with Time Windows.

Quant à la planification réactive, un canevas général de stratégies de ré-optimisation est proposé et permet d'élaborer, à partir de trois paramètres, des stratégies diverses en termes d'anticipation, de fréquence et de conservatisme. Les meilleures configurations trouvées sont comparées à deux approches manuelles de régulation. A travers un large plan d'expérience mené sur une plateforme de simulation, des éléments clefs sont analysés, tel quel le niveau d'utilisation et l'objectif à optimiser lors de la planification proactive et en fonction du degré de dynamise (de peu à très dynamique).

Enfin, l'intégration de ces algorithmes dans un outil commercial est présentée avec les premiers ajustements et perspectives futures, suite au remarques et retours des régulateurs d'ambulances.