



HAL
open science

New variants and solution methods for the dial-a-ride problem: An application to the transportation of disabled children

Timothée Chane-Haï

► To cite this version:

Timothée Chane-Haï. New variants and solution methods for the dial-a-ride problem: An application to the transportation of disabled children. Operations Research [math.OC]. INSA de Lyon, 2023. English. NNT: 2023ISAL0125 . tel-04523367

HAL Id: tel-04523367

<https://theses.hal.science/tel-04523367>

Submitted on 27 Mar 2024

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

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



N°d'ordre NNT : 2023ISAL0125

**THESE DE DOCTORAT DE L'INSA LYON,
membre de l'Université de Lyon**

**Ecole Doctorale N° 512
Informatique et Mathématiques**

Spécialité : Génie industriel

Soutenue publiquement le 13/12/2023, par :
Timothée CHANE-HAÏ

**Nouvelles variantes et méthodes de
résolution pour le problème de
transport à la demande**

**Application au transport d'enfants en
situation de handicap**

Devant le jury composé de :

| | |
|---|------------------------|
| Sylvie NORRE, professeur des universités, Université de Clermont-Auvergne | Présidente |
| Maria DI-MASCOLO, directrice de recherche, Grenoble INP CNRS | Rapporteuse |
| Samir LOUDNI, professeur, IMT Atlantique | Rapporteur |
| Sylvie NORRE, professeur des universités, Université de Clermont-Auvergne | Examinatrice |
| Dominique FEILLET, professeur, Écoles des mines de Saint-Etienne | Examineur |
| Thibaud MONTEIRO, professeur des universités, INSA Lyon | Directeur de thèse |
| Samuel VERCRAENE, maître de conférences HDR, INSA Lyon | Co-directeur de thèse |
| Céline ROBARDET, professeur des universités, INSA Lyon | Co-directrice de thèse |
| thèse | |
| Geovanny MONTOYA, docteur en génie industriel, Nomad | Invité |

Référence : TH1026_CHANE-HAI Timothée

L'INSA Lyon a mis en place une procédure de contrôle systématique via un outil de détection de similitudes (logiciel Compilatio). Après le dépôt du manuscrit de thèse, celui-ci est analysé par l'outil. Pour tout taux de similarité supérieur à 10%, le manuscrit est vérifié par l'équipe de FEDORA. Il s'agit notamment d'exclure les auto-citations, à condition qu'elles soient correctement référencées avec citation expresse dans le manuscrit.

Par ce document, il est attesté que ce manuscrit, dans la forme communiquée par la personne doctorante à l'INSA Lyon, satisfait aux exigences de l'Etablissement concernant le taux maximal de similitude admissible.

Département FEDORA – INSA Lyon - Ecoles Doctorales

| SIGLE | ECOLE DOCTORALE | NOM ET COORDONNEES DU RESPONSABLE |
|------------------|---|--|
| CHIMIE | <u>CHIMIE DE LYON</u> https://www.edchimie-lyon.fr Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage secretariat@edchimie-lyon.fr | M. Stéphane DANIELE C2P2-CPE LYON-UMR 5265 Bâtiment F308, BP 2077 43 Boulevard du 11 novembre 1918 69616 Villeurbanne directeur@edchimie-lyon.fr |
| E.E.A. | <u>ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE</u> https://edeea.universite-lyon.fr Sec. : Stéphanie CAUVIN Bâtiment Direction INSA Lyon Tél : 04.72.43.71.70 secretariat.edeea@insa-lyon.fr | M. Philippe DELACHARTRE INSA LYON Laboratoire CREATIS Bâtiment Blaise Pascal, 7 avenue Jean Capelle 69621 Villeurbanne CEDEX Tél : 04.72.43.88.63 philippe.delachartre@insa-lyon.fr |
| E2M2 | <u>ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION</u> http://e2m2.universite-lyon.fr Sec. : Bénédicte LANZA Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.e2m2@univ-lyon1.fr | Mme Sandrine CHARLES Université Claude Bernard Lyon 1 UFR Biosciences Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69622 Villeurbanne CEDEX sandrine.charles@univ-lyon1.fr |
| EDISS | <u>INTERDISCIPLINAIRE SCIENCES-SANTÉ</u> http://ediss.universite-lyon.fr Sec. : Bénédicte LANZA Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.ediss@univ-lyon1.fr | Mme Sylvie RICARD-BLUM Institut de Chimie et Biochimie Moléculaires et Supramoléculaires (ICBMS) - UMR 5246 CNRS - Université Lyon 1 Bâtiment Raulin - 2ème étage Nord 43 Boulevard du 11 novembre 1918 69622 Villeurbanne Cedex Tél : +33(0)4 72 44 82 32 sylvie.ricard-blum@univ-lyon1.fr |
| INFOMATHS | <u>INFORMATIQUE ET MATHÉMATIQUES</u> http://edinfomaths.universite-lyon.fr Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 infomaths@univ-lyon1.fr | M. Hamamache KHEDDOUCI Université Claude Bernard Lyon 1 Bât. Nautibus 43, Boulevard du 11 novembre 1918 69 622 Villeurbanne Cedex France Tél : 04.72.44.83.69 hamamache.kheddouci@univ-lyon1.fr |
| Matériaux | <u>MATÉRIAUX DE LYON</u> http://ed34.universite-lyon.fr Sec. : Yann DE ORDENANA Tél : 04.72.18.62.44 yann.de-ordenana@ec-lyon.fr | M. Stéphane BENAYOUN Ecole Centrale de Lyon Laboratoire LTDS 36 avenue Guy de Collongue 69134 Ecully CEDEX Tél : 04.72.18.64.37 stephane.benayoun@ec-lyon.fr |
| MEGA | <u>MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE</u> http://edmega.universite-lyon.fr Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bâtiment Direction INSA Lyon mega@insa-lyon.fr | M. Jocelyn BONJOUR INSA Lyon Laboratoire CETHIL Bâtiment Sadi-Carnot 9, rue de la Physique 69621 Villeurbanne CEDEX jocelyn.bonjour@insa-lyon.fr |
| ScSo | <u>ScSo*</u> https://edsciencessociales.universite-lyon.fr Sec. : Mélina FAVETON INSA : J.Y. TOUSSAINT Tél : 04.78.69.77.79 melina.faveton@univ-lyon2.fr | M. Bruno MILLY Université Lumière Lyon 2 86 Rue Pasteur 69365 Lyon CEDEX 07 bruno.milly@univ-lyon2.fr |

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

New variants and solution methods for the dial-a-ride problem:
an application to the transportation of disabled children

Timothée CHANE-HAÏ

December 2023

Remerciements

De par ce travail de thèse, je rajoute ma modeste contribution à l'ensemble des connaissances humaines. Bien que modeste, cette contribution a une valeur singulière à mes yeux car elle est le résultat des heureux hasards qui ont façonné ma vie. J'ai eu la chance de naître au bon endroit, au bon moment, dans la bonne famille, de rencontrer les bonnes personnes, de prendre les bonnes décisions... Bien qu'on ne puisse nier les efforts personnels que j'ai dû déployer pendant ces trois années de recherche, je reste convaincu que mon mérite est essentiellement dû au hasard. Un hasard qui s'incarne en les personnes qui ont contribué directement ou indirectement au succès de cette thèse.

Je remercie mes directeurs de thèse : Thibaud, Samuel, Céline. Merci de m'avoir accompagné durant ces trois années. J'en ressors grandi intellectuellement. Votre bienveillance a été sans nul doute une des clés de la réussite de cette thèse. Grâce à cela, j'ai vécu cette aventure avec sérénité. Le doctorat met à rude épreuve la santé mentale des jeunes chercheurs. Je l'ai lu dans des articles, je l'ai vu chez d'autres doctorants, mais je ne l'ai (presque) jamais vécu personnellement. Merci à vous de m'avoir donné la liberté de réfléchir, d'échouer, puis de réfléchir encore et de réussir enfin.

Je remercie mes collègues de laboratoire pour leurs conseils, leur aide, et pour leur capacité à déformer la trame de l'espace-temps lors des pauses-café. J'ajoute une mention spéciale pour mon maître de l'informatique : Guy. Toute valeur numérique présente dans ce manuscrit n'existe que grâce à ta capacité à expliquer et réexpliquer patiemment le fonctionnement d'un serveur de calculs. Exécuter mille heures de calcul en une heure, voilà une méthode efficace pour rattraper les pauses-café.

Je remercie mes collègues enseignants. Merci de m'avoir fait confiance en me donnant la responsabilité d'enseigner à des classes entières d'étudiants. Enseigner, c'est difficile. Comment transmettre un maximum de connaissances à un maximum d'élèves ? Voici la question que je me suis posée et que je me pose encore. Mais grâce à vos conseils, je pense m'être un peu rapproché d'une réponse. Bien qu'incertaine, cette réponse a indéniablement guidé ma manière d'enseigner. Merci d'avoir fait passer l'élève qui s'asseyait au fond de la classe au-devant du tableau.

Je remercie mes élèves. Merci d'avoir fait confiance à l'enseignant débutant que j'étais. En particulier, merci pour le respect dont vous avez fait preuve envers un professeur souvent à peine plus âgé et parfois même plus jeune que vous. Merci pour votre curiosité intellectuelle, merci d'avoir appris.

Je remercie mes parents. Merci de m'avoir donné les clés pour réussir à la fois sur le plan académique, professionnel et social. Beaucoup de ce que je suis, de ce que je fais, c'est grâce à vous. Afin de contenir la taille de ce paragraphe, toutes les énumérations seront abrégées par "etc.". Merci pour les valeurs que vous m'avez transmises : la persévérance, l'humilité, le respect d'autrui, l'entraide, la bienveillance, etc. Merci de m'avoir guidé dans mon rapport à moi-même et aux autres. Si aujourd'hui je me sens à

l'aise en société, c'est parce que vous m'avez habitué aux interactions sociales dès le plus jeune âge : aller chercher seul le pain, demander l'addition au restaurant, aller voir les autres enfants à la plage, etc. Merci d'avoir investi autant de temps, d'argent, d'énergie pour que je puisse aller au karaté, au piano, au tennis, à la natation, à la danse, au saxophone, aux cours particuliers, à l'équitation, au tennis, au centre aéré, aux compétitions, aux concerts, aux soirées, à la plage, chez les cousins, chez mes amis, etc. Papa, merci d'avoir été mon spécialiste des sciences naturelles et formelles. Je dois ma curiosité intellectuelle à nos longues discussions dans la voiture et à nos heures de visionnage d'émissions scientifiques. On a abordé beaucoup de sujets : les dinosaures, les avions, les tornades, la matière, l'espace, la théologie, etc. Merci de m'avoir donné l'envie de comprendre le monde. Maman, merci d'avoir été ma spécialiste des sciences humaines et sociales. Lorsque j'ai des problèmes avec d'autres personnes, tu m'écoutes, tu comprends ma situation, tu canalises mes émotions, et surtout, tu sais me faire prendre du recul. Grâce à toi, j'arrive à comprendre, analyser et gérer toutes situations impliquant une ou plusieurs personnes. Merci de m'avoir donné l'envie de comprendre les autres. Merci à vous deux de m'avoir aidé à devenir adulte. Grâce à vous, j'ai aujourd'hui une pensée qui m'est propre et la capacité d'assumer les décisions qui découlent de cette pensée. Malheureusement pour vous, votre aura sacrée à mes yeux d'enfants, a disparu au fur et à mesure que je devenais adulte. Cependant, même si je ne suis plus votre petit garçon et que je pense par moi-même, merci de continuer à m'écouter. Nos discussions me font toujours grandir comme au premier jour.

Je remercie mes deux petites soeurs. Merci de ne pas m'avoir oublié quand j'ai commencé mes études à 10000 km de la maison. Même à distance, aucun être humain ne me fait plus rire que vous deux. Merci pour tous ces messages, ces appels, ces dessins, et ces photos qui ont réchauffé les hivers continentaux.

Je remercie ma famille (ama, aye, papi, mamie, tatie, tonton, cousins, cousines, et amis de la famille). Merci de m'avoir gardé les mercredis après-midi quand j'étais au primaire, pour m'avoir accueilli pendant les vacances scolaires, pour les repas de famille, pour l'aide aux devoirs, pour la bienveillance, pour les joies, et pour les peines. Sans nul doute, nos liens familiaux très forts ont eu une influence positive sur mon développement. On ne choisit pas sa famille, mais si j'avais à choisir, je choisirais la nôtre.

Je remercie ma belle-famille. Vous m'avez accueilli à bras ouverts sans jamais broncher devant mon appétit glouton, mes gammes interminables, ou ma présence à chaque confinement. Votre foyer est parfois l'open-space idéal pour le télétravail, mais plus souvent le petit îlot paradisiaque sur lequel je peux me ressourcer.

Merci Julie pour tout ce que tu m'apportes, pour tout ce que tu représentes, pour tout ce que tu es.

Je remercie mes amis. Vous avez été essentiels dans la construction d'une pensée qui m'est propre. Grâce à vous, mon désir de richesse financière a laissé place à un désir de richesses humaines, artistiques et intellectuelles. Cette richesse immatérielle fait de moi une bien meilleure personne, pour moi-même et pour les autres. Je remercie particulièrement M. Johan, mon camarade de thèse, mon maître à penser, mon confident, mon meilleur ami. Dans la quête de recherche de la vérité et du bien commun, nul n'est meilleur compagnon que toi.

Merci aux communautés scientifiques passées, présentes, et à venir, aux communautés des sciences naturelles, fondamentales, humaines et sociales. Sans sciences, je n'aurais pas pu prendre l'avion pour venir faire mes études en Métropole. Sans sciences, je n'aurais pas eu accès d'un simple clic à l'ensemble des connaissances humaines. Sans sciences, je n'aurais pas pu appeler ma famille pour garder un lien avec eux. Sans sciences, je n'aurais pas pu évoluer dans ce monde que j'apprécie tant. Je suis fier d'appartenir à une communauté qui privilégie la production intellectuelle à la production matérielle. Bien que la positivité de l'impact des sciences soit discutable, je tends à croire que globalement, elles ont rendu le monde un peu plus agréable pour l'humanité et pour moi-même.

Je remercie l'État français, en particulier pour m'avoir sauvé la vie gratuitement lorsque j'ai eu l'appendicite, et pour avoir financé en grande partie mes études. Pouvoir étudier sans être inquiété par ses finances et sa santé est un luxe que je souhaite à chaque étudiant.

Je remercie tous ceux qui ont rendu, qui rendent, et qui rendront ce monde un peu meilleur pour les générations passées, présentes, et futures.

Je souhaite à présent une bonne lecture à ceux qui s'aventureront plus loin dans ce manuscrit, et je remercie les autres d'avoir lu jusqu'ici.

“L'école, c'est l'avenir.” - *Papi Catherine*

Résumé en Français

Un système de transport durable pour les passagers et les marchandises assure une mobilité sûre, inclusive, abordable et propre pour les générations actuelles et futures.

*GIEC - Groupe d'experts Intergouvernemental sur l'Évolution du Climat
(traduit de l'anglais)*

Le transport est un domaine particulier de nos sociétés. Rien n'en est produit, pourtant il est omniprésent. De l'échelle la plus petite à la plus grande, le transport joue un rôle de support qui façonne nos interactions. Cependant, le transport a deux facettes. D'un côté, il ne fait nul doute qu'il a contribué à l'amélioration de la qualité de vie autour du globe. D'un autre côté, il exerce une pression croissante sur l'environnement et la santé humaine. Par conséquent, il est nécessaire de trouver de meilleures manières de se déplacer et de transporter nos biens pour construire un futur désirable et durable. Le partage de véhicules fait partie des solutions possibles. Dans le contexte du transport de passagers, les transports publics et les taxis sont les alternatives les plus courantes à la voiture individuelle. Cependant, les transports publics ne sont pas toujours pratiques (ex : horaires d'ouverture, foules importantes, faible couverture dans les zones peu denses, ...) et les taxis sont chers. Ces caractéristiques les rendent peu commodes pour les personnes en situation de vulnérabilité comme les personnes âgées et les personnes handicapées. En réponse à ces problèmes, les services de transport à la demande sont apparus comme des solutions intermédiaires, équilibrant accessibilité et coût.

Dans un système de transport à la demande, une flotte de véhicules transporte un ensemble d'utilisateurs de leur point de départ individuel jusqu'à leur point d'arrivée. Dans ce système, les utilisateurs se partagent les véhicules. Afin de résoudre le problème de transport à la demande (*Dial-A-Ride Problem* - DARP), une tournée de véhicules est organisée. Cette tournée de véhicules doit satisfaire les demandes de trajets des utilisateurs, respecter les contraintes de qualité de service, et minimiser un objectif cible (ex : coûts de transport, pollution, ...). Les systèmes de transport à la demande mettent l'accent sur la qualité de service. Par conséquent, ils peuvent prendre des formes différentes pour intégrer les besoins spécifiques de leurs utilisateurs.

Cette thèse explore des variantes du DARP et les résout en utilisant de nouvelles méthodes de résolution. Des cas d'étude pratiques sur le transport d'enfants en situation de handicap sont fournis.

La première partie de ce travail étudie l'utilisation de techniques d'apprentissage automatique pour résoudre le DARP. Deux méthodes ont été développées.

La première méthode identifie les caractéristiques des bonnes solutions du DARP par un processus de classification *offline*, et ensuite crée de nouvelles heuristiques qui exploitent les caractéristiques extraites. Finalement, les nouvelles heuristiques sont testées sur des instances classiques de la littérature. Malgré un an et demi de travail sur ce sujet, aucun résultat comparable à l'état de l'art actuel n'a été obtenu. En raison du temps limité pour réaliser cette thèse, nous sommes passés à d'autres pistes de recherche. Néanmoins, nous apportons des éléments de réponses et des perspectives sur le travail initial de cette thèse.

La deuxième méthode basée sur l'apprentissage automatique adapte une méthode *online*, appelée *Nested Rollout Policy Adaptation* (NRPA), au DARP classique. Le NRPA est intégré dans une heuristique gloutonne. Cette intégration exploite les capacités d'apprentissage du NRPA en déléguant les tests de réalisabilité à l'heuristique gloutonne. À nouveau, la méthode de résolution est testée sur les instances classiques de la littérature. Des conclusions et pistes de recherche sont obtenues de l'analyse détaillée des résultats.

La deuxième partie de cette thèse introduit une nouvelle variante : le problème journalier de transport à la demande (*Commuting Dial-A-Ride Problem* - Com-DARP). Dans cette variante, chaque usager a une demande de trajet le matin, une demande de trajet le soir, et un temps de trajet maximal à la journée. Par conséquent, un long trajet le matin est compensé par un trajet court le soir. Au lieu de considérer les demandes du matin et du soir de manière indépendante, le Com-DARP optimise les routes à l'échelle de la journée. Ce problème est inspiré du transport d'enfants en situation de handicap en France. Cependant, il s'applique à tous les systèmes de transport impliquant des flux pendulaires. Le Com-DARP est résolu par une métaheuristique de recherche à petit et grand voisinage couplée à un filtre de précedence (*Small and Large Neighborhood Search with Precedence Filter* - SLNS-PF). Il s'agit de la première mise en application d'un SLNS pour résoudre le DARP. De plus, le filtre de précédences agit comme mécanisme d'accélération de la résolution qui exploite les relations de dépendance entre les trajets du matin et du soir. Des recommandations pratiques et des perspectives de recherches sont déduites des résultats expérimentaux.

La troisième partie de ce manuscrit introduit le problème d'affectation et de transport à la demande (*Assignment Dial-A-Ride Problem* - ADARP). Le ADARP est la variante la plus sophistiquée abordée lors de cette thèse. Ce problème est à nouveau inspiré du transport d'enfants en situation de handicap en France. La nouveauté du ADARP repose dans l'intégration simultanée de trois aspects de ce système de transport : l'affectation des personnes handicapées dans des établissements de santé, la gestion du personnel en fonction des besoins des usagers, et la construction des tournées de véhicules. La combinaison de ces trois aspects permet une optimisation plus globale du système. Cependant, cela accroît la taille du problème de manière significative. Afin de résoudre ce problème, une matheuristique a été développée : la recherche itérative de routes (*Iterative Route Search* - IRS). Cette méthode de résolution construit un ensemble de routes en utilisant une recherche à large voisinage et sélectionne les meilleures routes avec une méthode exacte. Testée sur des instances réelles, cette étude apporte des connaissances pratiques pour les professionnels du secteur et des pistes de recherche pour les chercheurs.

Pour résumé, cette thèse étudie trois variantes du DARP. Premièrement, la variante classique du DARP est résolue en utilisant des méthodes de résolution basée sur l'apprentissage automatique. Ensuite, deux nouvelles variantes du DARP sont introduites, toutes les deux inspirées par le transport d'enfants en situation de handicap. Ces deux variantes sont traitées avec de nouvelles méthodes de résolution, respectivement une métaheuristique et une matheuristique. Chaque étude permet d'obtenir des éléments de réponse pratiques et théoriques.

La première étude au chapitre 4 a été présentée à la conférence Chane-Haï et al. (2022). Les chapitres 5 et 6 sont des versions mises à jour des articles de journaux Chane-Haï et al. (2023b) (soumis) et Chane-Haï et al. (2023a) (à paraître dans *Health Care Management Science*).

Contents

| | |
|--|-----------|
| Remerciements | 3 |
| Résumé en Français | 6 |
| List of Tables | 12 |
| List of Figures | 15 |
| List of publications | 16 |
| List of acronyms | 18 |
| 1 Introduction | 21 |
| 2 Context | 23 |
| 2.1 Transportation systems | 23 |
| 2.2 The sanitary transportation | 24 |
| 2.2.1 Taxonomy of transportation systems | 24 |
| 2.3 The regular non-urgent transportation of disabled children in France | 28 |
| 2.3.1 Handicap in France | 28 |
| 2.3.2 Stakeholders | 29 |
| 2.3.3 Challenges faced by the regular non-urgent transportation of disabled children in France | 30 |
| 2.4 Conclusion | 32 |
| 3 Literature review | 35 |
| 3.1 Operations research | 35 |
| 3.2 Routing problems | 35 |
| 3.3 The Dial-A-Ride Problem (DARP) | 36 |
| 3.3.1 The classic DARP | 37 |
| 3.3.2 Taxonomy of DARPs | 38 |
| 3.3.3 Solution methods | 44 |

| | | |
|----------|---|------------|
| 3.4 | Research gaps relative to research questions | 48 |
| 3.4.1 | RQ1: How can we build an algorithm that computes fast and optimal solutions for the regular non-urgent sanitary transportation? | 48 |
| 3.4.2 | RQ2: How can we extend the scope of the optimization and consider the different aspects related to all the stakeholders? | 49 |
| 3.4.3 | RQ3: How the dependence between requests can be used to improve the transportation? | 49 |
| 3.5 | Contributions | 49 |
| 4 | Machine learning applied to the dial-a-ride problem | 51 |
| 4.1 | Introduction | 51 |
| 4.2 | Literature review | 52 |
| 4.2.1 | A brief overview of machine learning | 52 |
| 4.2.2 | Machine learning applied to operations research | 53 |
| 4.2.3 | Machine learning for routing problems | 54 |
| 4.3 | Knowledge-guided search for the DARP using offline machine learning | 57 |
| 4.3.1 | Introduction | 57 |
| 4.3.2 | Solution method | 58 |
| 4.3.3 | Presentation of the numerical experiments | 59 |
| 4.3.4 | Data generation | 59 |
| 4.3.5 | Characteristic extraction | 64 |
| 4.3.6 | Building knowledge-guided operators | 70 |
| 4.3.7 | Conclusion and perspectives | 72 |
| 4.4 | Nested rollout policy adaptation applied to the dial-a-ride problem | 75 |
| 4.4.1 | Introduction | 75 |
| 4.4.2 | Solution method | 76 |
| 4.4.3 | Computational experiments | 79 |
| 4.4.4 | Conclusion and perspectives | 83 |
| 5 | The Commuting Dial-A-Ride Problem (Com-DARP) | 85 |
| 5.1 | Introduction | 85 |
| 5.2 | Literature review | 87 |
| 5.2.1 | The Dial-A-Ride Problem | 87 |
| 5.2.2 | Contributions | 88 |
| 5.3 | Problem description | 89 |
| 5.3.1 | General problem setting | 89 |
| 5.3.2 | Mathematical model | 89 |
| 5.4 | Solution method | 91 |
| 5.4.1 | Small and Large Neighborhood Search (SLNS) framework | 93 |
| 5.4.2 | Operators | 93 |
| 5.4.3 | Precedence filter | 96 |
| 5.4.4 | Feasibility testing | 98 |
| 5.5 | Computational experiments | 99 |
| 5.5.1 | Benchmark instances | 99 |
| 5.5.2 | Performance evaluation | 100 |
| 5.5.3 | Managerial insights | 101 |
| 5.6 | Conclusion | 104 |
| 6 | The Assignment Dial-a-Ride Problem (ADARP) | 105 |
| 6.1 | Introduction | 105 |
| 6.2 | Literature review | 107 |
| 6.3 | Model | 108 |
| 6.3.1 | Notations | 108 |
| 6.3.2 | Single-route problem | 109 |

| | | |
|----------|---|------------|
| 6.3.3 | Route selection problem and staff sizing | 110 |
| 6.4 | Solution method | 110 |
| 6.4.1 | General framework: Iterative Route Selection (IRS) | 111 |
| 6.4.2 | Large Neighborhood Search (LNS) | 111 |
| 6.4.3 | Assignment problem | 113 |
| 6.5 | Case study | 114 |
| 6.6 | Validation of the method | 116 |
| 6.6.1 | Validation of the components and tuning of parameters | 116 |
| 6.6.2 | Comparison with the literature | 119 |
| 6.7 | Managerial results | 119 |
| 6.7.1 | From the MSIs' perspective | 120 |
| 6.7.2 | From people with disabilities and other vulnerabilities perspective | 122 |
| 6.8 | Conclusion | 122 |
| 7 | Conclusions and perspectives | 125 |
| 7.1 | Conclusions | 125 |
| 7.2 | Perspectives | 127 |
| 8 | Final words | 129 |
| | Bibliography | 130 |
| | Appendix | 145 |
| A.1 | List of extended arc metrics | 145 |
| A.2 | List of notations for the Com-DARP | 145 |
| A.3 | Precedences for the Com-DARP | 147 |
| A.3.1 | Pickup/pickup | 147 |
| A.3.2 | Drop-off/drop-off | 148 |
| A.3.3 | Pickup/drop-off | 149 |
| A.3.4 | Drop-off/pickup | 149 |
| A.4 | List of notations for the ADARP | 150 |
| A.5 | Best-known solutions for the ADARP | 151 |

List of Tables

| | | |
|------|---|-----|
| 1 | list of acronyms | 19 |
| 2.1 | comparison of urgent and non-urgent patient transportation (Tellez, 2019) | 26 |
| 2.2 | comparison of occasional and regular non-urgent patient transportation (Tellez, 2019) | 27 |
| 3.1 | comparison of public transportation services (Ho et al., 2018) | 37 |
| 3.2 | classification of DARP relative to the nature of the information | 43 |
| 4.1 | LNS parameter setting | 59 |
| 4.2 | computation time (in s) to reach a 1% gap | 60 |
| 4.3 | list of instance metrics | 61 |
| 4.4 | list of solution metrics | 63 |
| 4.5 | list of arc metrics | 63 |
| 4.6 | prediction accuracy (in %) when using all the metrics at solution level | 65 |
| 4.7 | prediction accuracy (in %) when using all the metrics at arc level | 65 |
| 4.8 | prediction accuracy (in %) of relative to the number of leaf nodes in the decision tree at solution level | 66 |
| 4.9 | prediction accuracy (in %) relative to the number of leaf nodes in the decision tree at arc level | 66 |
| 4.10 | average gaps (in %) on the Cordeau 2006 A benchmark | 73 |
| 4.11 | values for tuning the parameters of the NRPA | 80 |
| 4.12 | parameter tuning results relative to α (gaps in %) | 80 |
| 4.13 | parameters tuning results relative to the number of levels and iterations (gaps in %) | 80 |
| 4.14 | parameter tuning results (gaps in %) | 81 |
| 4.15 | LNS parameter setting | 82 |
| 4.16 | comparison between NRPA-3-37, NRPA-3-100 and LNS | 82 |
| 5.1 | parameter setting for the metaheuristics | 99 |
| 5.2 | results on the Com-DARP benchmark | 100 |
| 5.3 | comparison between the Com-DARP and the corresponding 2-DARP | 102 |
| 6.1 | number of users, MSIs, and depots for the three instances | 114 |
| 6.2 | number of users relative to the width of the time windows for a DTF = 1.75 | 115 |

| | | |
|------|---|-----|
| 6.3 | average number of feasible itineraries (i.e., MSIs) per user depending on DTF | 115 |
| 6.4 | number of users with i feasible itineraries (i.e., MSIs) depending on DTF | 116 |
| 6.5 | evaluation of the components, average gap with the BKS in % for 5 runs | 117 |
| 6.6 | number of times the assignment problem is solved during an iteration of the LNS, average gap with the BKS in % for 5 runs per line | 118 |
| 6.7 | metrics relative to the number of times the assignment problem is solved during an iteration of the LNS | 118 |
| 6.8 | comparison of costs for Pinson et al. (2016) instances with 500 iterations for GRASP and IRS | 119 |
| 6.9 | comparison with the best solutions in the literature Dumez et al. (2021a) and Dumez et al. (2021b) and the best solutions for 5 runs of IRS | 120 |
| 6.10 | effect of user assignment and sharing of vehicles on the cost and the number of routes | 121 |
| 6.11 | evolution of the two main expenses for MSIs: routes and wages between Scenarios (A) and (D) | 121 |
| 6.12 | impact of increasing the number of user places (Capacity C5) | 122 |
| 6.13 | ride time evolution and proportion of users moved from their current assignment | 123 |
| A.1 | list of extended arc metrics | 145 |
| A.2 | Best-Known Solution (BKS) | 151 |

List of Figures

| | | |
|------|--|-----|
| 2.1 | partial taxonomy of sanitary transportation (Tellez, 2019) | 25 |
| 2.2 | stakeholders of the non-urgent sanitary transportation in France (Tellez, 2019) | 29 |
| 3.1 | taxonomy of the DARP | 39 |
| 4.1 | learning process (Wang et al., 2009) | 51 |
| 4.2 | number of articles related to machine learning and vehicle routing problems per year | 55 |
| 4.3 | route metrics | 62 |
| 4.4 | decision tree at solution level on the 2% gap dataset with 5 leaves | 68 |
| 4.5 | decision tree at solution level on the 4% gap dataset with 5 leaves | 69 |
| 4.6 | decision tree at solution level on the 2% gap dataset with 3 leaves without the distance-related metrics | 69 |
| 4.7 | decision tree at arc level on the 2% gap 1000 iterations dataset with 3 leaves | 70 |
| 4.8 | decision tree at arc level on the 2% gap 5000 iterations dataset with 3 leaves | 71 |
| 4.9 | decision tree at arc level on the 2% gap 5000 iterations dataset with only one split on the ratio between the number of apparitions and the number of scheduling tests succeeded | 71 |
| 4.10 | the four steps of the Monte Carlo Tree Search (MCTS) | 75 |
| 4.11 | NRPA with 1 level and 2 iterations | 78 |
| 5.1 | daily maximum ride time decomposition for the Com-DARP | 86 |
| 5.2 | column to update after an insertion | 94 |
| 5.3 | k-regret matrix | 95 |
| 5.4 | precedence relation | 96 |
| 5.5 | convergence speed of the LNS, SLNS, and SLNS-PF (200 users, MRTC = 1.5) | 101 |
| 5.6 | best cost relative to the maximum ride time coefficient for instances with 100 users | 103 |

List of publications

Journal articles

- Timothée Chane-Haï, Samuel Vercaene, Céline Robardet, Thibaud Monteiro. The Commuting Dial-A-Ride Problem. EURO Journal on Transportation and Logistics, 2023 (submitted)
- Timothée Chane-Haï, Samuel Vercaene, Thibaud Monteiro. The Assignment Dial-A-Ride Problem. Health Care Management Science, 2023

Conferences with proceedings

- Timothée Chane-Haï, Samuel Vercaene, Céline Robardet and Thibaud Monteiro. Improving the non-urgent sanitary transportation. 7th International Conference on Control, Automation and Diagnosis (ICCAD'23), Rome - Italy, 2023
- Timothée Chane-Haï, Samuel Vercaene, Thibaud Monteiro. Sharing a ride time constraint in a multi-trip dial-a-ride problem, an application to the non-urgent patient transportation problem. 13th International Conference on Modeling, Optimization and Simulation (MOSIM 2020), Agadir - Morocco, 2020

Conferences without proceedings

- Timothée Chane-Haï, Samuel Vercaene, Céline Robardet, Thibaud Monteiro. Guidage des métaheuristiques par machine-learning, application au transport d'enfants en situation de handicap. 23ème congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF), Lyon - France, 2022
- Timothée Chane-Haï, Samuel Vercaene, Céline Robardet, Thibaud Monteiro. Knowledge-guided search applied to the transportation of disabled people. Smart Healthcare International Conference (SHeIC), Troyes - France, 2021

List of acronyms

| acronym | definition |
|----------|---|
| ADARP | Assignment Dial-A-Ride Problem |
| ALNS | Adaptive Large Neighborhood Search |
| BKS | Best-Known Solution |
| Com-DARP | Commuting Dial-A-Ride Problem |
| DAR | Dial-A-Ride |
| DARP | Dial-A-Ride Problem |
| GDARP | Generalized Dial-A-Ride Problem |
| GRASP | Greedy Randomized Adaptive Search Procedure |
| GVRPTW | Generalized Vehicle Routing Problem with Time Windows |
| LNS | Large Neighborhood Search |
| MCTS | Monte Carlo Tree Search |
| MIP | Mixed-Integer Programming |
| MILP | Mixed-Integer Linear Programming |
| ML | Machine Learning |
| MSI | Medico-Social Institution |
| NMCS | Nested Monte Carlo tree Search |
| NRPA | Nested Rollout Policy Adaptation |
| OR | Operations Research |
| PDP | Pickup and Delivery Problem |
| RG | Research Gap |
| RQ | Research Question |
| SLNS | Small and Large Neighborhood Search |
| SPP | Shortest Path Problem |
| TSP | Traveling Salesman Problem |
| VRP | Vehicle Routing Problem |

Table 1: list of acronyms

Chapter 1

Introduction

A sustainable transport system provides safe, inclusive, affordable, and clean passenger and freight mobility for current and future generations.

IPCC - Intergovernmental Panel on Climate Change

Transportation is a peculiar sector of our societies. It does not produce anything, yet it is everywhere. From the smallest to the largest scale, transportation plays a supporting role that shapes our interactions. However, it is a double-edged sword. Without a doubt, transportation has improved the quality of life around the globe. But at the same time, it exerts increasing pressure on the environment and human health. Consequently, finding better ways to move people and goods is necessary for building a desirable and sustainable future. Vehicle sharing is one of the numerous solutions. In the context of passenger transportation, public transportation and taxis are the most common alternatives to the individual car. However, public transportation is not always practical (e.g., opening hours, dense crowds, low frequency in certain areas, ...) and taxis are expensive. These characteristics make them impractical for vulnerable people such as elderly and disabled persons. As a response, dial-a-ride services emerged as a middle-ground solution, balancing accessibility and cost.

In a dial-a-ride system, a fleet of vehicles transports a set of users from their individual pickup locations to their individual drop-off locations. The vehicles are shared by the users. To solve a Dial-A-Ride Problem (DARP), a routing plan is created. This routing plan must satisfy all user demands, respect the constraints related to the quality of service, and minimize a target objective (e.g., routing cost, pollution, ...). Dial-a-ride services put a great emphasis on the quality of service. As a result, they can take many forms in order to integrate the specific needs of their users.

This thesis explores variants of the DARP and solves them using novel solution methods. Practical applications are provided in the context of disabled children transportation.

The first part of this work investigates the use of Machine Learning (ML) techniques to solve the basic DARP. Two methods have been developed.

The first method identifies the characteristics of good DARP solutions using an offline classification

process, and then creates new heuristics that take advantage of the extracted characteristics. Finally, the new knowledge-guided heuristics are tested on classic DARP instances from the literature. Despite exploring various ideas for a year and a half, this research has not yielded satisfactory results that could challenge the current state of the art. Due to the limited duration of this thesis, we had to move on to other research subjects. Nevertheless, we provide some insights and perspectives on the initial work of this thesis.

The second ML-based method adapts an online method called Nested Rollout Policy Adaptation (NRPA) to the classic DARP. The NRPA is integrated into a greedy heuristic framework. This integration leverages the learning capabilities of the NRPA by delegating feasibility checks to the greedy heuristic. Again, the solution method is tested on classic DARP instances from the literature. Conclusions and perspectives are derived from the detailed analysis of the results.

The second part of this thesis introduces a new variant: the Commuting Dial-A-Ride Problem (Com-DARP). In this variant, each user has one morning request, one afternoon request, and a daily maximum ride time. As a result, long morning trips are compensated by short afternoon trips. Instead of considering the morning and afternoon requests independently, the Com-DARP optimizes routes at the scope of the entire day. This problem is inspired by the transportation of disabled children in France. Nevertheless, it is applicable to any transportation system involving commuting. The Com-DARP is solved using a Small and Large Neighborhood Search coupled with a Precedence Filter (SLNS-PF). This is the first implementation of the SLNS to the DARP. In addition, the precedence filter provides a speed-up mechanism that exploits the dependence between morning and afternoon routes. Managerial insights and research perspectives are derived from the experiments.

The third part of this manuscript introduces the Assignment Dial-A-Ride Problem (ADARP). The ADARP is the most sophisticated DARP variant tackled in this thesis. Again, it is inspired by the transportation of disabled children in France. The novelty of the ADARP lies in the simultaneous integration of three aspects of this transportation system: the assignment of disabled people to medical facilities, the staff sizing relative to the needs of the users, and the management of the shared door-to-door transportation service. Combining the three problems allows for a more holistic optimization. However, it significantly increases the intractability of the problem. To address this problem, a matheuristic has been developed: the iterative route search (IRS). This solution method builds a pool of routes using a large neighborhood search and selects the best routes using a MILP solver. Tested on real-life instances, this study provides managerial insights to practitioners and research perspectives to academics.

In summary, this thesis investigates three variants of the DARP. First, the classic DARP is solved using ML-based approaches. Then, two new DARP variants are introduced, both inspired by the transportation of disabled children. These two variants are solved with novel solution methods, respectively a metaheuristic and a matheuristic. For each study, valuable insights for academics and practitioners are derived.

The first study presented in Chapter 4 has been presented in the conference Chane-Haï et al. (2022). Chapter 5 and 6 are updated versions of the journal articles Chane-Haï et al. (2023b) (submitted) and Chane-Haï et al. (2023a) (to be published in Health Care Management Science).

Chapter 2

Context

2.1 Transportation systems

Transportation is vital for our society. Over the past century, the transportation of people, goods, and information has increased tremendously thanks to constant technological advances. Transportation systems have metaphorically contracted distances across the planet (Chapman, 2007). Many critical components of our society, such as international political institutions (e.g., European Union, United Nations, North Atlantic Treaty Organization, ...), expanded social networks, and the global economy, exist and are sustained by transportation. Banister (2010) describes transportation systems as the blood system of society, playing a central role in its development. However, they also generate problems such as air pollution (Anenberg et al., 2019), noise (Münzel et al., 2021), increased morbidity and mortality (James et al., 2020), congestion (Christodoulou and Christidis, 2020), impact on biodiversity (Walker et al., 2019; Owens et al., 2020), and so forth. These problems negatively impact human life, the environment, and the economy. In regard to the climate crisis, transportation was responsible for 15% of global carbon emissions (8.7 GtCO₂-eq) in 2019, making it the fourth largest emitting sector after energy, industry, and AFOLU¹ (Calvin et al., 2023).

While dealing with these negative impacts, transportation systems must also face incoming challenges related to population growth. The population has grown exponentially in recent history. From 1 billion people in 1804, the population has reached 8 billion in 2022. According to the projection of the United Nations², the population is expected to peak at 10 billion around 2080. Therefore, the demand for transportation has grown exponentially and will continue to grow in the coming years. In addition to the increasing in demand, it is also becoming more diverse to better meet the needs of individuals (e.g., commuting, sanitation, leisure, ...). For individuals, affordable transportation is crucial for their integration into their communities. As such, it plays a significant role in social equity and inclusion.

To face the current and incoming challenges, transportation must shift toward green, shared, and flexible mobility (Brůhová Foltýnová et al., 2020). In order to meet the goal set in the Paris Agreement, carbon emissions from transportation must be reduced to 2-3 Gt by 2050 (de la Rue du Can, Stephane et al., 2022). According to the European Environment Agency (EEA), CO₂ emissions related to

¹AFOLU: Agriculture, Forestry and Other Land Use

²<https://www.un.org/en/global-issues/population> (visited on 31/08/2023)

transportation systems in Europe must be reduced by 90% in order to achieve climate neutrality by 2050³. The electrification of transportation systems and shared mobility play key roles in this transition (Miskolczi et al., 2021; Jin et al., 2018).

2.2 The sanitary transportation

Most individuals rely on transportation systems as part of their daily routines. These systems are vital for maintaining an efficient functioning of society, enabling the movement of individuals and goods. However, their importance extends beyond everyday activities. Indeed, in the field of healthcare, medical transportation plays a critical role in ensuring that patients receive care and treatment on time. Despite adding no value to medical treatments, sanitary transportation is essential to health systems as it relates to the “ability to reach” medical care (Levesque et al., 2013). Therefore, it is important to maximize its efficiency in order to allocate as many resources as possible to care. Inadequate transportation translates to unnecessary costs and hinders individuals’ access to medical care. This may result in delayed treatment (Kironji et al., 2018), limited time for receiving care (Gaans and Dent, 2018), or even prevent individuals from receiving care altogether (Cyr et al., 2019). As a component of healthcare systems, an ideal sanitary transportation system should adhere to six key principles (Khanassov et al., 2016): “approachability, acceptability, availability, accommodation, affordability, appropriateness”.

In addition to facing the same challenges as general transportation systems, sanitary transportation is particularly impacted by the increase in the aging population and the aging-in-place demand. As a result of longer life expectancy and lower fertility rates, the aging population is growing. In the Asia-Pacific region, the percentage of individuals over the age of 60 is projected to increase from 11% in 2012 to 24% in 2050 (Holmes, 2021). As older people require more intense medical care, their impact on healthcare systems is increasing. Alongside the growth of the aging population, the increasing number of individuals choosing to age at home is contributing to a rise in demand for sanitary transportation. The possibility for individuals to age in place is being supported by advances in technology, evolving cultural and personal preferences, and changes in health policy (Alders and Schut, 2019; Wieczorek et al., 2022). For instance, the use of personal emergency systems enhances safety by allowing the elderly to easily notify caregivers when assistance is needed (De San Miguel and Lewin, 2008). The aging of the population and the trend toward aging-in-place leads to a higher demand for sanitary transportation.

As the demand for sanitary transportation grows, so does its cost and environmental impact. In France, the healthcare sector is responsible for 8% of total CO2 emissions, with transportation accounting for 16% of that amount (Marraud et al., 2023). At the same time, the increasing financial pressure threatens the long-term sustainability of healthcare systems (European Commission. Directorate General for Economic and Financial Affairs. and Economic Policy Committee of the European Communities., 2019). As transportation is a significant expense for healthcare structures, its efficiency has a large impact on the overall cost of healthcare systems (Hains et al., 2011).

Unfortunately, due to the diversity of healthcare structures and sanitary transportation, there is no universal solution to these challenges. To improve a sanitary transportation system, its unique characteristics must be taken into account.

2.2.1 Taxonomy of transportation systems

There is a lot of diversity in sanitary transportation. The structure of a transportation system depends on the nature of the transported item or individual, the degree of urgency, and the frequency of the transportation. Figure 2.1 presents a partial taxonomy of the sanitary transportation that gradually focus on the problem tackled in this thesis: the regular non-urgent patient transportation.

2.2.1.1 Transportation of patients, staff, and supplies.

Transportation of patients, staff, and supplies is crucial for the operation of healthcare systems. Although the various types of transportation are interconnected, each has its unique characteristics and challenges.

³<https://www.eea.europa.eu/publications/transport-increasing-oil-consumption-and/increasing-oil-consumption-and-ghg> (visited on 31/08/2023)

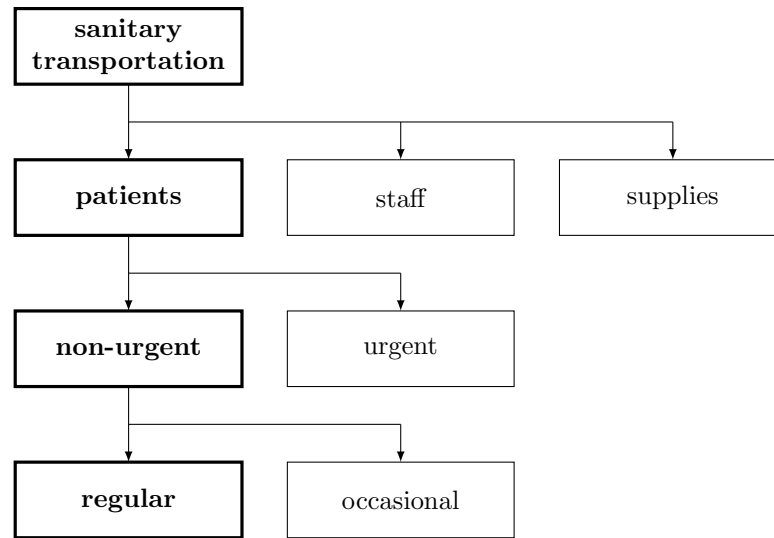


Figure 2.1: partial taxonomy of sanitary transportation (Tellez, 2019)

As a result, they can be examined both independently and in relation to one another.

Medical staff. The transportation of medical staff is often related to home healthcare services. On the one hand, short-term medical home healthcare often corresponds to a period of recovery after a hospitalization. On the other hand, long-term non-medical home healthcare primarily assists the elderly and individuals with disabilities in carrying out their daily activities (e.g., mobility care, food preparation, hygiene, shopping, swimming, cleaning, cooking, ...). Long-term non-medical care often works in conjunction with long-term medical care which provides at-home medical appointments, treatment management, emergency services, and more. Home healthcare has many advantages over more conventional care options such as hospitals or skilled nursing facilities. In most cases, it is cheaper, as effective, and more convenient. This makes it suitable for recovery after hospitalization, maintaining the current health conditions, and respecting the patient’s wish to stay at home.

As the nature of care depends on the individual, many practitioners are involved, including therapists, doctors, psychologists, nurses, and more. Organizing the transportation and scheduling for home healthcare services is challenging. As a consequence, a good organization is essential for home healthcare in order to be effective and provide high-quality services. We refer the reader to Euchi et al. (2022) for a recent review of home healthcare routing and scheduling.

Supplies. Hospitals manage a complex supply chain that deals with a large variety and quantity of items (e.g., pharmaceuticals, surgical medical products, medical equipment, sterile items, linen, food, ...). De Vries and Huijsman (2011) defines healthcare supply chain management as “the information, supplies, and finances involved with the acquisition and movement of goods and services from the supplier to the end user in order to improve clinical outcomes while controlling costs”. To achieve a good quality of care, the right items must be available at the right time and place. This task is challenging due to the number of items and the unpredictability of emergencies that make it difficult to keep track of them (Lewis et al., 2010). Poor supply chain management can impede the quality of care (e.g., delayed treatments). In addition, it can generate significant financial loss as the medical supply cost is the second largest expenditure after personnel cost in hospitals (Moons et al., 2019).

Patients. There are different types of patient transportation: intra-hospital, inter-hospital, and to-hospital transportation. First, intra-hospital transportation involves moving patients within a medical facility to receive specific treatments (e.g., x-rays, surgery, blood tests, ...). Because the distance traveled are short, intra-hospital transportation does not require motorized vehicles but mostly stretchers. Second,

inter-hospital transportation occurs when a hospitalized patient needs to be transferred to another hospital with more appropriate medical care. Finally, to-hospital transportation refers to the transportation of a patient from outside the hospital to the hospital. The return trip from the hospital back home can also be considered (e.g., disabled children going to socio-medical institutions in the morning and going back home in the afternoon). Organizing patient transportation can be challenging as it may require specific staff, specific materials, and may be unpredictable (Kulshrestha and Singh, 2016). For critically ill or injured patients, efficient transportation is a matter of life and death (Droogh et al., 2015).

This thesis focuses on patient transportation.

2.2.1.2 Urgent and non-urgent patient transportation.

The organization of patient transportation depends largely on the level of emergency. The main differences between urgent and non-urgent patient transportation are summarized in Table 2.1 (Tellez, 2019).

| characteristics | urgent | non-urgent |
|---------------------|----------------------|--|
| transport demand | immediate unknown | on-availability unknown or known in advance |
| transportation cost | high | medium - low |
| fleet utilization | low | medium - high |
| operating cost | high | medium - low |
| service quality | high | medium - low |
| response time | short | short - long |
| ride times | short | medium - long |

Table 2.1: comparison of urgent and non-urgent patient transportation (Tellez, 2019)

Urgent patient transportation. Urgent patient transportation corresponds to the transportation from the point of injury or illness to a hospital or place of definitive care (Sikka and Margolis, 2005; Huggins and Shugg, 2008). In the context of a life-threatening or time-critical situation, the response time must be as short as possible. Providing the optimal emergency service is difficult due to the high unpredictability in magnitude, time, and location (Tellez, 2019). Yet, highly responsive emergency services are crucial for reducing “the morbidity and mortality associated with sudden medical and traumatic emergencies” (Sikka and Margolis, 2005).

The organization of emergency medical services highly depends on the country’s socio-economic level, culture, politics, geography, and healthcare infrastructure. As a result, a large diversity of emergency medical services exists (Sikka and Margolis, 2005). Urgent patient transportation systems can range from having no specific organization to being highly structured with on-site intensive care units and skilled medical personnel. Nevertheless, two primary strategies can be identified: “scoop and run” and “stay and play” (Fassbender et al., 2013). The first one focuses on quickly transporting the patient to the appropriate hospital with minimal prehospital care. In contrast, the second strategy involves deploying skilled medical personnel to provide intensive prehospital care, thus reducing the urgency of reaching the hospital.

We refer the reader to Aringhieri et al. (2017) for a review of emergency medical services.

Non-urgent patient transportation. Non-emergency medical transportation refers to the transportation of patients who are not in a critical condition. Nevertheless, it differs from ordinary passenger transportation because it requires additional services according to the condition of the patient (e.g., medical equipment, lying position, ...). For this type of transportation, demand may be known in advance (e.g., transportation of patients between hospitals, transportation of children with disabilities to medico-social institutions, medical appointments of elderly, ...) or arise unexpectedly (e.g., emergency

calls for non-emergency situations). When the demand is known in advance, a more efficient planning is possible. It gives time to optimize aspects related to the travel time, the costs, ridesharing, and more. On the opposite, when demand is not known in advance, decision-makers must quickly decide whether to accept or reject the request. Despite being non-urgent, this type of transportation must still provide a high quality of service. Indeed, improper transportation can worsen a patient's health. For example, patients in serious but not critical condition require adequate medical equipment, staff, and support during transportation (Deasy and O'Sullivan, 2007).

Because the majority of sanitary transportation is non-urgent by nature, optimizing it offers valuable financial and organizational benefits for health care systems (Robinson et al., 2009; Hains et al., 2011). Indeed, efficient non-emergency medical transportation can free up critical resources for emergencies. For example, improving the efficiency of non-urgent transportation can free up ambulances for urgent transportation (Jagtenberg et al., 2017). At the same time, the transportation cost is reduced by using less expensive vehicles instead.

This thesis focuses on non-urgent patient transportation.

2.2.1.3 Regular and occasional non-urgent patient transportation.

Two types of non-urgent patient transportation can be identified: occasional and regular non-urgent patient transportation. The main differences between the two are summarized in Table 2.2 as presented in Tellez (2019).

| | occasional | regular |
|---------------------|--|--|
| frequent objectives | number of served patients, transportation costs, individual/collective preferences | transportation cost, consistency (time, staff, group) |
| planning horizon | short and medium-term (i.e., hours, days) | long-term (i.e., months, years) |
| operating cost | medium - high | low - medium |
| examples | punctual medical appointments, extra-curricular activities for disabled children | patients with long-term treatments, daily or weekly transportation for disabled children |

Table 2.2: comparison of occasional and regular non-urgent patient transportation (Tellez, 2019)

Occasional non-urgent patient transportation. Occasional non-urgent patient transportation refers to the transport of patients with non-recurring demands (Tellez, 2019). This transportation is a specific application of on-demand transportation in a medical context. The occasional non-urgent patient transportation is typically used for punctual medical appointments for patients or extracurricular activities for disabled children.

Regular non-urgent patient transportation. Regular non-urgent patient transportation corresponds to the transportation of patients with recurring needs over a planning horizon (Tellez, 2019). This concerns several situations. The transportation can occur daily in the case of disabled children going to socio-medical institutions, or twice a week (on Monday and Friday) for those going to boarding houses. Patients who need regular care such as dialysis, tracheotomy, or ventilator support may need to be transported several times a week based on the frequency of their appointments. The quality of service is especially important due to the recurring nature of the transportation. A high quality of service increases the comfort for users and provides a competitive advantage over the concurrence for service providers. Many factors contribute to the quality of service such as reliability, transfers, waiting time, consistency,

and more. A detailed overview of quality measures can be found in Paquette et al. (2009), Accorsi and Vigo (2021).

According to the presented taxonomy, this thesis work focuses on regular non-urgent sanitary transportation for disabled children.

2.2.1.4 The levels of decisions in transportation systems

Decisions can have a variety of impacts, affecting a single user up to the entire health system and lasting from one day to several years. They can be classified into four categories: strategic, tactical, operational, and real-time decisions.

Strategic decisions. Strategic decisions have a long-term impact on healthcare structures, usually lasting for years. Examples include fleet sizing, determining the opening hours of medical-social institutions, and selecting depot locations.

Tactical decisions. Tactical decisions typically have an impact that lasts for several months. They usually address multi-period problems such as regular non-urgent patient transportation. Decisions regarding consistency (e.g., such as driver or time consistency) are considered tactical decisions.

Operational decisions. Operational decisions are made in the context of regular and occasional transportation. It requires that the demand is known in advance. Since these decisions typically last for a single day, they are more suited toward occasional transportation. However, it is possible to manage multi-period problems by repeating single-day decisions. While this approach may lack the broader perspective of the tactical level, it can still produce effective decisions. As a consequence, a significant portion of the literature has focused on this type of decision-making.

Real-time decisions. Real-time decisions involve managing requests that arise after the transportation planning has already been completed and vehicles are in service. These new requests must be either accepted or rejected, and the transportation plan must be dynamically adjusted to accommodate them. This requires a high degree of flexibility and responsiveness from the transportation system.

2.3 The regular non-urgent transportation of disabled children in France

2.3.1 Handicap in France

In France, handicap is defined according to two criteria: severe physical, sensory, or cognitive limitations; severe restrictions in daily activities due to health problems (Bellamy et al., 2023). For example, physical, sensory, or cognitive limitations refers to situations where a person has difficulties to walk more than 500 meters on a flat terrain, a visual impairment despite medical correction, or difficulties to understand and to be understood by others, among others. As a result, disabled persons often need assistance such as care from medical staff, daily help from a family member, an adaptation of their home layout, and so forth.

According to a report of the french directorate for research, studies and statistics (*Direction de la Recherche, des Études, de l'Évaluation et des Statistiques - DREES*) (Bellamy et al., 2023), 14% of people over 15 years old were living at home while being disabled (7.6 million persons) in 2021. For children, this number represented 4.8% of the population between 5 and 14 years old (approximately 400000 children). Providing care to disabled persons living at home requires an adequate organization. Two main options are available: home care or care at Medico-Social Institutions (MSI). In 2018, 58280 disabled persons under 20 years old received care at home and 110920 went to MSIs. Overall, this represent approximately 1% of the population under 20 years old in France. This demand is increasing. Indeed, the number of disabled children receiving care has grown by 17.9% between 2006 and 2018 (+69.6% for home care, +3.1% for MSIs).

Each children go to MSIs according to their medical situations. As a result, the transportation frequency varies from child to child. In 2018, approximately half of the children were commuting daily to MSIs on week days and one out of five children were going to their MSIs on Mondays and were returning home on Fridays. Few children were continuously staying at MSIs or going to MSIs on certain days of the week. In essence, this corresponds to a regular non-urgent sanitary transportation system. These differences in needs and transportation frequency require a particular attention from transporters in order to provide a good quality of service to the children and their families. The end goal is to provide equity and inclusion for all the users and help them to accomplish their personal life project. In order to organize an adequate transportation systems, multiple actors are involved.

2.3.2 Stakeholders

The regular non-urgent sanitary transportation of disabled children involves multiple stakeholders: the funder, the MSIs, the transporters, and the users (children and their families). Their interactions are detailed in the following paragraphs and illustrated in Figure 2.2.

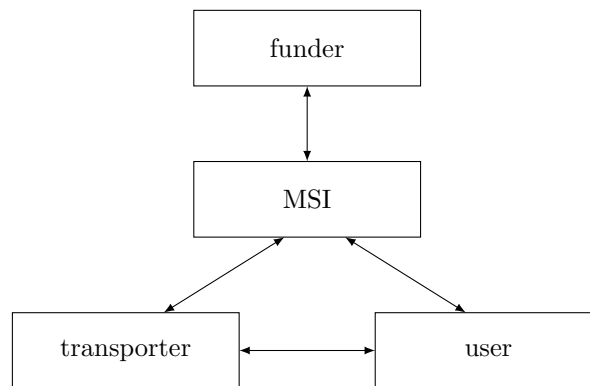


Figure 2.2: stakeholders of the non-urgent sanitary transportation in France (Tellez, 2019)

Funder. The sanitary transportation of people with disabilities in France is financed by the French public health insurance system (*Sécurité Sociale*) which grants an annual subsidy to MSIs through national, regional (*Agences Régionales de Santé*), and departmental (*Maisons Départementales pour les Personnes Handicapées*) socio-medical structures. With this grant, MSIs must manage all their activities for the coming year.

Medico-social institutions. MSIs provide care for individuals in vulnerable situations, such as the elderly, disabled individuals, those with serious diseases, early childhood, and so forth (Marraud et al., 2023). Their main goal is to help people suffering from disabilities. This help can take several forms: assessment and prevention of social and medico-social risks; administrative or judicial protection; medical care; educational, social, and professional integration; and daily life assistance⁴. Several types of MSIs carry out these missions: structures for prevention, screening, and early support; structures providing support in a daily-life environment; and support structures in institutions⁵. Our work focuses on MSIs involved in educational and medical care in institutions.

Transporter. In the sanitary transportation system in France, the transporter is in charge of the organization, planning, and execution of the transportation. MSIs are usually in charge of the transportation (vehicles and drivers) but some choose to outsource the transportation. As vulnerable people are

⁴Article L311-1 - Code de l'action sociale et des familles - Légifrance, https://www.legifrance.gouv.fr/codes/article_1c/LEGIARTI000045293678 (visited on 31/08/2023)

⁵ESMS: Établissement ou service social ou médico-social - <https://www.monparcourshandicap.gouv.fr/glossaire/esms> (visited 31/08/2023)

transported, transporters are required to adhere to strict regulations regarding safety measures, driving competencies, and vehicle specifications (Tellez et al., 2020).

Users. Sanitary transportation involves both the users and their families (Tellez et al., 2020). Patients using this system have varying levels of autonomy, such as being able to walk or requiring a wheelchair. Due to the situation of disabled children, it is usually a family member who manages the relationship with the MSI and the transporter. As such, the quality of the transportation is, to some extent, accessed by the families.

The regular non-urgent transportation of disabled children concerns thousands of children and their families every day in France. Every day, users are picked up at their homes and taken to their assigned MSI where they receive personalized care throughout the day. At the end of the day, users are taken from their respective MSI to their homes (Chane-Haï et al., 2023a). The goal of this sanitary transportation is to provide equity and inclusion for the users. Thanks to a law promoting the equality of people with disabilities in French society, there has been a positive evolution in transportation options for disabled children (*Loi n° 2005-102 du 11 février 2005 relative à l'égalité des droits et des chances, la participation, et la citoyenneté des personnes handicapées*).

2.3.3 Challenges faced by the regular non-urgent transportation of disabled children in France

The regular non-urgent patient transportation of disabled children in France faces the same challenges as other transportation systems but also specific challenges related to its application. The research questions identified in this thesis derives from these challenges.

2.3.3.1 Current situation

For all the stakeholders of the regular non-urgent transportation of disabled children in France, an ideal transportation system would have a low cost and a high quality of service. In addition, this ideal transportation system would also consider related aspects such as the environmental impact, the fleet management, the staff planning, the fairness among users, and so forth.

Thus, progress can be made in regard to the actual transportation system. A study conducted in the Auvergne-Rhône-Alpes region of France revealed that the average trip is 74 minutes and 48 kilometers. While half of the users were satisfied with the service, the other half were not entirely satisfied. In addition, approximately one out of ten users expressed their total dissatisfaction regarding the ride times (Tellez et al., 2020). From a financial perspective, there is also room for improvement. According to a 2012 report (Vachey et al., 2012), the expenditure associated with the sanitary transportation of disabled individuals ranges from 350 to 400 million euros. This represents approximately 10% of the total cost of sanitary transportation in France. Furthermore, when patients are minors, the MSIs cover the entire cost with public funding contributions from the French public health insurance system (*Sécurité sociale*). For MSIs, transportation expenses represent the second largest expense after wages. Between 2006 and 2014, these costs have increased by 40%. The high financial cost due to non optimal routes also translates to a high environmental cost. In 2016, approximately 250 diesel vehicles transported 1150 users, for a total of 4 million kilometers in the city of Lyon and its surrounding areas. This resulted in the emission of 1.42 million kg of CO₂ equivalent. In the context of global warming, it is necessary to reduce this carbon footprint.

2.3.3.2 Improving the regular non-urgent transportation of disabled children in France

The current transportation is not optimal for at least four main reasons: 1) the tools used for building the routing are not appropriate, 2) certain aspects of the transportation are not properly considered, 3) the dependency across a user's requests is not fully exploited, and 4) the current system is not able to adapt in real-time to unexpected events.

1) Building better decision-making tools. First, using better decision-making tools would greatly enhance the transportation of disabled children. Currently, routings are either built by hand or by using optimization solvers. First, hand-made routes are far from being optimal. Indeed, it is impossible to draw by hand the optimal routes for such a complex transportation. In addition, manually building the route takes much more time than using a solver. Thus, all this time allocated to planning means less time allocated to care. Moreover, this situation requires considerable efforts to incorporate new users into the planning, inducing long response times. Second, despite providing better results, the solvers used by transporters can still be improved. In practice, transporters do not have access to the latest optimization algorithms. However, the real challenge is that no solver is able to manage the large numbers of users involved in the regular non-urgent sanitary transportation in France. As a consequence, the generated routes are not optimal.

This lack of appropriate solvers means that this transportation system is not optimal. The greatest improvements will probably come from having better solvers. The first characteristic of an ideal solver would be its ability to generate optimal routes, with a controlled balance between cost and quality of service. However, this is not sufficient to be implemented for real applications. Indeed, it would also require a high computation speed. This second characteristic would allow to generate routes in a short amount of time, making it practical for real-life applications. From these elements, we can derive our first research question:

RQ1: How can we build an algorithm that computes fast and optimal solutions for the regular non-urgent sanitary transportation?

2) Integrating all aspects related to the transportation. The regular transportation of disabled children goes beyond providing on-demand transportation. Indeed, multiple aspects must be considered to reach an optimal system (e.g., fairness among users, staff scheduling, ...). While the addition of new components may benefit the system as a whole, it requires stakeholders to evaluate their impacts on the balance between the cost and the quality of service. Some aspects are occasionally considered such as consistency or resource allocation. However, their incorporation is not consistent among transporters. This can be explained by a lack of available information or/and the difficulty to use it. Indeed, in the face of day to day operations, collecting data is not the priority of the actors involved in this transportation system. In addition, the solver used by decision-makers may not be able to use the collected data in their models. We identified several elements to improve the transportation system.

From an optimization perspective, several aspects may improve the quality of the transportation. First, the sharing of vehicles is not implemented by all MSIs. Sharing may take multiple forms. For instance, a vehicle can be parked at any MSI, children going to different MSIs can be in the same vehicle, a vehicle can be used by one MSI on a day and by another MSI on a different day, and so forth. Second, a better fleet management would also benefit to the transportation system. Indeed, MSIs have the possibility to adjust their fleet once a year. Sensibility analysis by adding, removing or exchanging vehicles would give insights on the ideal fleet of vehicles at the scope of one or multiple MSIs. Third, meeting points could also provide benefits by reducing the number of locations to visit. However, this practice raises additional challenges in terms of assignment of users to meeting points, grouping locations, management of delays, and so on.

Aspects from a user perspective can also improve the quality of service. First, the assignment of users mainly depends on the availability at MSIs. As a result, some users do not go to the closest MSIs, inducing long ride times. A better assignment process or the possibility to change the current assignment would certainly reduce these ride times. Second, unfairness among users comes from the difference between their ride times. One user may have trips without much detours compared to its direct itinerary, and another one may have long trips with lots of detours. These unfair situations can emerge from poor assignments or simply because this criteria was not considered by the decision-makers. Third, users and their family appreciate consistency in the transportation service. It can take multiple forms: staff consistency, user consistency, time consistency.

Finally, aspects from the staff perspective can improve either the quality of service for users or the working condition of staff members. First, each staff member has specific competences. As a result, their allocation must consider their competencies in order to have the right practitioner for the right care.

Indeed, allocating an overqualified practitioner at one spot might result in a shortage at another one. Furthermore, it could be possible to measure the impact of recruiting new staff members on the overall system. Second, similarly to the users, staff may greatly benefit from fair situations. Indeed, unfair working conditions across staff members emerge as the priority is put on the users' needs. This results in unbalanced workloads or ride times to the workplaces. Third, competence management and fairness could be addressed by building the staff schedule. With the scheduling incorporated in the decision process, it could be possible to have a broader view and more precise control over staff-related aspects.

In regard to all these elements, we can define the second research question:

RQ2: How can we extend the scope of the optimization and consider the different aspects related to all the stakeholders?

3) Exploiting the regularity of the requests. In regular non-urgent sanitary transportation, a user has multiple requests over a time horizon. However, many transporters take decisions assuming that these requests are independent. While independent decisions are sufficient, this practice does not take advantage of the regularity of the requests to improve the transportation. This can result in unfair situations. For example, two users might only have a ride time difference of 5 minutes when going to their MSI. Nevertheless, this is repeated every day of the week, all year-round. Thus, the cumulative difference becomes significant. Avoiding such an unfair situation is only possible by processing a user's requests as a set of dependent requests.

Saying that a user has dependent requests means that the decision for serving one request may impact the decision for serving another one. Depending on the scope studied, this impact may last for one to multiple days.

In regard to this potential improvement, our third research question is:

RQ3: How the dependence between requests can be used to improve the transportation?

4) Building a dynamic system. In the context of regular non-urgent sanitary transportation of disabled children in France, requests are known in advance. This gives enough time to prepare and optimize the routes. As a result, when a vehicle leaves its depot, its entire route is already planned. Even if the majority of the planning can be performed statically, this transportation system is still subject to unexpected events. For instance, a user might show late or not show at all, a vehicle might break down or be stuck in a traffic jam. These events may result in large differences between the planned and the real routings. As a consequence, the ability to respond to such events would be beneficial for this transportation system.

Two elements are required to implement dynamic adaptation. First, it must be possible to change the routes after the beginning of service. Second, real-time communication must be possible between the users and the drivers. Indeed, dynamic adaptation requires the transporter to be able to inform the users of any change and reverse. With these two elements, the transportation can further be improved. For example, knowing in advance that a user will not show and being able to change the route accordingly will prevent unnecessary detours.

As the majority of the transportation is performed statically, the fourth research question is:

RQ4: How dynamic information can be collected from the stakeholders, conveyed to the other stakeholders, and used to respond to unexpected events?

While this research question certainly has its interest, we did not study it during this thesis. Due to a restricted time to perform this research, we decided to focus on the first three research questions.

2.4 Conclusion

Managing the regular non-urgent transportation of disabled children in France is challenging and requires a proper organization. Any improvements made to the system can benefit not only the users but also

other stakeholders and society as a whole, both current and future. In this chapter, we identified four research questions related to potential improvements, three of which are studied in this thesis:

- RQ1: How can we build an algorithm that computes fast and optimal solutions for the regular non-urgent sanitary transportation?
- RQ2: How can we extend the scope of the optimization and consider the different aspects related to all the stakeholders?
- RQ3: How the dependence between requests can be used to improve the transportation?
- RQ4: How dynamic information can be collected from the stakeholders, conveyed to the other stakeholders, and used to respond to unexpected events?

The next chapter presents a literature review regarding these research questions. The corresponding research gaps for the first three questions are identified.

Chapter 3

Literature review

3.1 Operations research

Operations Research (OR) was developed during the Second World War to assist military planners in making decisions using mathematics (Fortun and Schweber, 1993). Due to its practical applications, OR has been adopted in various industries such as transportation, supply chain management, energy, and finance. The goal of OR is to use mathematical modeling, statistical analysis, and mathematical optimization to achieve optimal or near-optimal solutions to complex decision-making problems.

OR problems are often challenging due to the combinatorial explosion of potential decisions. Many OR problems are classified as NP-hard. This means that the existence of algorithms that can solve them in polynomial time relative to the size of the input is considered unlikely (Garey and Johnson, 1979). As a result, finding good solutions to real-life problems, characterized by a multitude of possibilities, presents significant challenges. Advances in OR are driven by improvements in algorithmic efficiency and increases in computational power offered by computer science.

3.2 Routing problems

Routing problems are classic problems that have been studied since the early days of OR. The essence of any routing problem is to find the optimal route(s) between a set of nodes on a graph. Routing problems cover a wide range of optimization challenges from the shortest path problem up to our problem: the Dial-A-Ride Problem (DARP).

Shortest Path Problem (SPP). The simplest form of routing problem is the Shortest Path Problem (SPP). Solving the shortest path problem means finding the shortest route from an origin to a final location (Kairanbay and Mat Jani, 2013). Despite its simplicity, the shortest path problem remains an active area of research due to its practical applications for network routing protocols, route planning, traffic control, path finding in social networks, computer games, transportation systems, and more (Madkour et al., 2017).

Traveling Salesman Problem (TSP). Expanding upon the SPP, the Traveling Salesman Problem (TSP) adds a constraint: all the nodes of the graph must be visited. Applied to the real-life situation of a traveling salesman visiting customers, the goal is to find the shortest possible route to visit a set of cities exactly once and return to the starting point. The earliest works on the TSP go back to the *Icosian game* (Jaques, 1859), the *Botenproblem* (messenger problem) (Menger, 1932), and the Chinese postman problem (Mei-Ko, 1962). As the TSP is NP-hard (Laporte, 1992), all problems derived from it are also NP-hard.

Vehicle Routing Problem (VRP). The Vehicle Routing Problem (VRP) generalizes the TSP by incorporating more than one vehicle into the problem. The goal is to design the optimal routing for a fleet of vehicles from a depot to a set of nodes (Clarke and Wright, 1964). Introduced as the “Truck Dispatching Problem” in Dantzig and Ramser (1959) and formulated as a linear optimization problem in Clarke and Wright (1964), the VRP has given birth to many variants to incorporate real-life characteristics such as time windows, heterogeneous fleets of vehicles, multiple depots, and more. Current research on the VRP is driven by the emergence of new technologies and challenges such as the green vehicle routing problem which addresses environmental concerns (Moghdani et al., 2021).

Pickup and Delivery Problem (PDP). The Pickup and Delivery Problem (PDP) builds upon the VRP by pairing the nodes. The PDP is characterized by a set of transportation requests that must be satisfied by a fleet of capacitated vehicles. Each request corresponds to a pickup and a delivery location that must be served in that order within their respective time windows. The PDP is particularly used in the context of urban logistics.

Dial-a-ride problem (DARP). Finally, the Dial-A-Ride Problem (DARP) generalizes the PDP by adding a maximum ride time for each pair of pickup and delivery. As an extension of the TSP, the DARP is NP-hard (Baugh Jr et al., 1998). Among the routing problems mentioned in this section, it is the most challenging to solve. Among the many applications of the DARP, patient transportation is one of the most studied contexts.

3.3 The Dial-A-Ride Problem (DARP)

In a Dial-A-Ride (DAR) system, a fleet of vehicles transports a set of users from their individual pickup location to their individual drop-off location. The vehicles are shared by the users. To solve a DARP, a routing plan is created. This routing plan must satisfy all users’ demands, respect the constraints related to the quality of service, and minimize the defined objective (e.g., routing cost, pollution, ...).

DAR systems are commonly used for on-demand transportation. On the one hand, they offer a more affordable alternative to taxi services by allowing vehicle sharing among users. On the other hand, DAR systems provide a faster, more flexible, and less crowded transportation option compared to conventional public transportation systems (buses, tramways, and subways). Therefore, they are better suited for individuals in vulnerable situations. As a result, DAR systems occupy a middle ground between mass public transit and taxi services, providing a balance between quality of service and cost. A detailed comparison from Ho et al. (2018) is given in Table 3.1.

The first DAR system was implemented in 1970 in the city of Mansfield, Ohio, USA. It was called dial-a-ride because users would book their requests via a phone call to the service provider. As similar services emerged, their potential for serving individuals in vulnerable situations became apparent. Consequently, DAR services gradually shifted their focus away from general public transportation and turned their attention toward the transportation of vulnerable users (Ho et al., 2018). Today, with the aging of the population, more and more people can benefit from such systems. DAR systems prevent the isolation of vulnerable groups by allowing their users to travel within their communities and access health services more easily and more comfortably.

Even if DAR systems mainly apply to the transportation of persons in vulnerable situations, such transportation services are relevant for other applications. On-demand transportation is a good alternative to taxis and mass transit systems when the demand is scarce such as at night time or in rural areas

| | public transportation | on-demand | taxi |
|-------------|-----------------------|--------------|------------|
| route | fixed | flexible | customized |
| schedule | fixed | by request | by request |
| speed | slow | medium | fast |
| cost | low | medium | high |
| mode | shared | shared | non-shared |
| capacity | high | medium | low |
| reservation | not needed | often needed | not needed |

Table 3.1: comparison of public transportation services (Ho et al., 2018)

(Parragh et al., 2015). In addition, DAR systems can complement other transportation systems providing an efficient solution for the first and last miles (Häll et al., 2009; Posada et al., 2017). In any application, DAR systems deal with two contradictory objectives: minimizing the cost and maximizing the quality of service (Cordeau and Laporte, 2003). While the cost is relatively easy to measure, evaluating the quality of service is more difficult. In addition, some aspects of the quality of service may be relevant in some situations, but irrelevant in others. For example, patient transportation might limit transfers between vehicles while night-time transportation might focus on reducing waiting times.

On-demand transportation can take many forms depending on its application. As a result, researchers have taken into consideration different features relevant to their case studies. Building upon the classic DARP formulation, additional features can be classified into three categories: advanced service design, alternative objectives, and the nature of decisions and information. In the following sections, we present the classic DARP formulation, followed by a taxonomy of DARP variants.

3.3.1 The classic DARP

The formulation proposed in Cordeau and Laporte (2003) serves as a reference for the DARP. It corresponds to a DARP with a single depot, homogeneous vehicles, and homogeneous demands.

Mathematical notations. The DARP is modeled on a complete directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$. Given N requests, $\mathcal{N} = \{n_0, n_1, \dots, n_{2N}\}$ is the set of nodes and $\mathcal{A} = \{(n_i, n_j) : n_i, n_j \in \mathcal{N}, i \neq j\}$ is the set of arcs. n_0 is the depot node, $\{n_1, \dots, n_N\}$ is the set of pickup nodes, $\{n_{N+1}, \dots, n_{2N}\}$ is the set of drop-off nodes, \mathcal{R} is the set of requests, and \mathcal{V} is the set of vehicles. Each request r_i is represented by a pair of nodes (n_i, n_{i+N}) , corresponding to a transportation from pickup node n_i to drop-off node n_{i+N} . Each arc (i, j) has a routing cost duration $C_{ij} \geq 0$ and a travel time $T_{ij} \geq 0$. All travel times and distances are non-negative. Each request r_i is bounded by a maximum ride time \bar{L}_i and each route served by a vehicle v is bounded by a maximum route duration \bar{L}_v . All requests are served and all routes end before the time horizon H . Each node n_i is served within its time window $[A_i, B_i]$ with $B_i \geq A_i \geq 0$. At the depot node n_0 , $A_0 = 0$ and $B_0 = H$. Each node n_i has a service duration $S_i \geq 0$. At depot node n_0 , $S_0 = 0$. The load variation at a pickup node n_i is $Q_i \geq 0$, and the load variation at the corresponding drop-off node n_{i+N} is $Q_{i+N} = -Q_i$. The load variation at the depot node n_0 is $Q_0 = 0$. Each vehicle v has a maximal capacity $Q_v \geq 0$.

Times computation. The arrival time at a node a_i can be before its time window. However, the service must start within the time window ($A_i \leq s_i \leq B_i$). In these situations, the vehicle waits at the node and the waiting time w_i corresponds to the elapsed time between the arrival and the beginning of service ($w_i = s_i - a_i$). The arrival time is always before the beginning of service ($a_i \leq s_i$). The departure time d_i corresponds to the end of service ($d_i = s_i + S_i$). The ride time r_i of a request corresponds to the elapsed time between the end of service at its pickup node and the beginning of service at its drop-off node ($r_i = s_{i+N} - d_i$).

Constraints. Solving a DARP consists in building a solution where all requests are served and all the following constraints are respected:

- a vehicle starts and ends its route at the depot,
- a pair of pickup and drop-off nodes are in the same route,
- a pickup node is served before its drop-off node,
- the maximal capacity of a vehicle is not exceeded,
- the maximal route duration is not exceeded,
- the maximal ride time of a user is not exceeded,
- the service at a node begins in its time window,
- a vehicle leaves from the depot and returns to the depot in its time windows.

Objective. The quality of a solution is accessed by the objective function. In Cordeau and Laporte (2003), the objective function corresponds to the minimization of the transportation cost. This is the most common objective in the DARP literature. However, other objectives have been studied by researchers in order to capture other aspects of dial-a-ride systems.

3.3.2 Taxonomy of DARPs

The classic DARP formulation can be extended to take into consideration real-life characteristics. In the following, we summarize the main characteristics using the taxonomy used in Molenbruch et al. (2017b). While we include recent advances and references from the past five years, we refer the reader to Molenbruch et al. (2017b) and Ho et al. (2018) for an in-depth survey of the literature up to 2017 and 2018. Figure 3.1 illustrates this taxonomy.

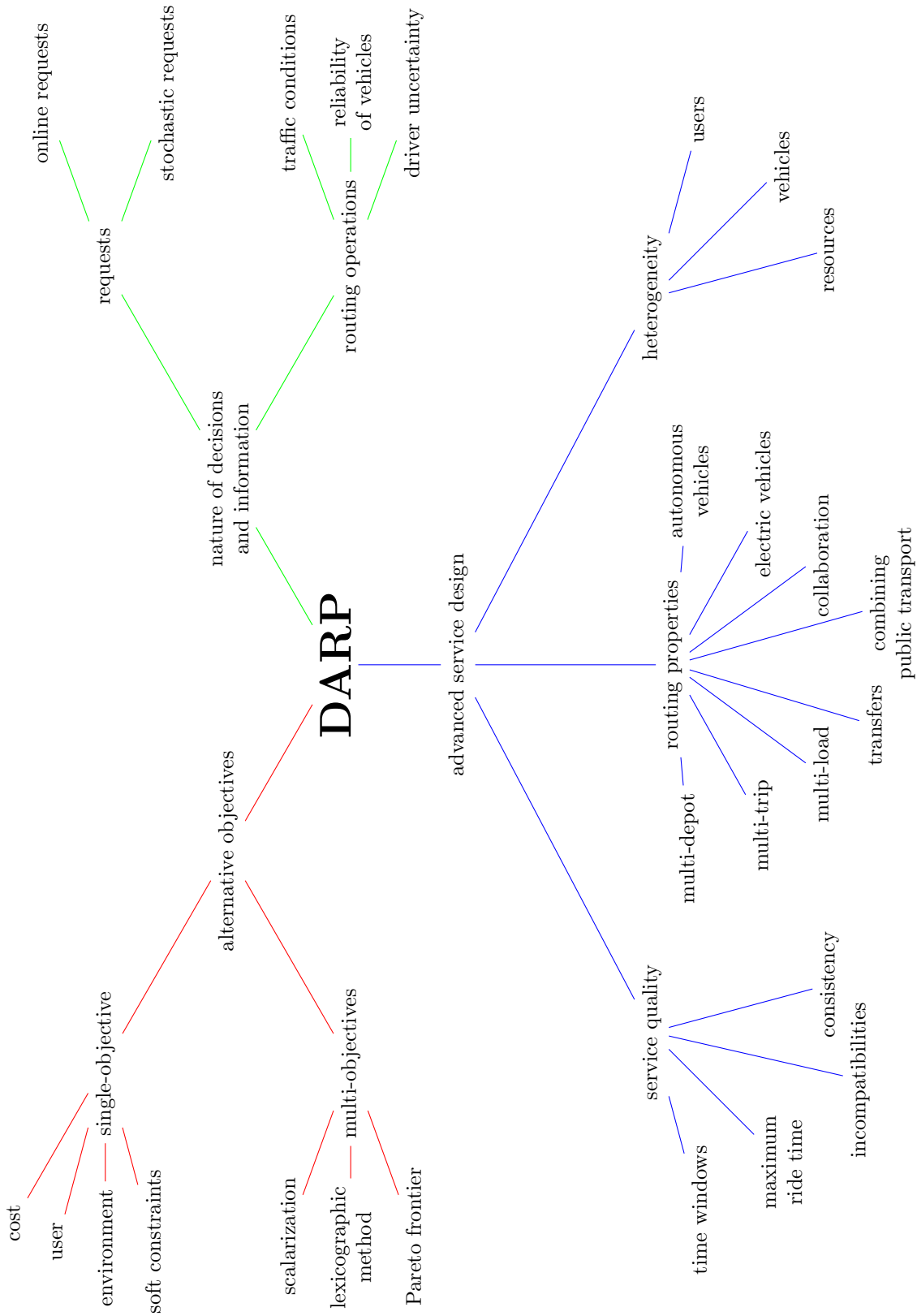


Figure 3.1: taxonomy of the DARP

3.3.2.1 Advanced service design

Heterogeneity. The classic DARP assumes that all users have the same needs (e.g., load, maximum ride time). However, real-life demand is more diverse. As each user has specific needs, heterogeneity is introduced into the problem.

Users have heterogeneous demand due to their specific physical needs or medical conditions. While some users are able to walk, some are in wheelchairs and some are on a stretcher (Parragh, 2011; Braekers et al., 2014; Malheiros et al., 2021). Furthermore, some need one or multiple accompanying persons while others do not (Liu et al., 2015; Zhang et al., 2015). The urgency of the demand relative to the medical condition of a user may also be considered (Hanne et al., 2009; Beaudry et al., 2010; Kergosien et al., 2011; Souza et al., 2022). This introduces a priority between urgent and non-urgent requests. For example, urgent demands can have a more restricted time frame than non-urgent ones. Another source of heterogeneity arises when transporting contagious patients (Zhang et al., 2015; Lim et al., 2017). In this case, the risk of contamination is considered to prevent the spread of diseases. Thus, isolated transportation is used instead of shared transportation.

Heterogeneous **vehicles** are required to satisfy the particular need of each request. As a consequence, configurable and reconfigurable vehicles can be considered as their layouts can be adjusted according to the needs of the transported users. For instance, a vehicle can be configured with a seating arrangement for users who are able to walk, or its seats can be folded to provide adequate space for users in wheelchairs. While the layout of configurable vehicles is set for the entire duration of the route (Liu et al., 2015), reconfigurable vehicles can change their interior layout during the route to accommodate incoming users (Tellez et al., 2018).

Heterogeneous **resources** (i.e., equipment and staff) are required as each user has specific needs. Some users might require medical equipment from specially-equipped vehicles and highly-qualified staff while others only require a driver (Xiang et al., 2006; Parragh, 2011; Detti et al., 2017; Cappart et al., 2018). As a result, the appropriate resources and staff must be assigned simultaneously according to the needs of each patient.

Routing properties. In a traditional DARP, vehicles depart from the depot, serve a set of requests, and then return to the depot. For a request, both the pickup and the drop-off must be served by the same vehicle. Finally, even if not explicitly mentioned in the classic definition by Cordeau and Laporte (2003), vehicles usually leave and return to the depot only once.

In a **multi-depot** DARP, there is more than one depot. A vehicle can either depart and return to the same depot (Braekers et al., 2014; Detti et al., 2017) or it can depart from one depot and end its route at another depot (Melachrinoudis et al., 2007). The second option encompasses multiple scenarios. For instance, drivers can return home with the service vehicle or service providers can anticipate a high demand around other depots on the next day.

In **multi-trip** DARP, vehicles can leave and return to a depot more than once. Multi-trip formulations are used to integrate personnel and sanitary constraints. As some users request specific qualifications, a vehicle can return to the depot to change its team members in order to meet the next patients' requirements (Lim et al., 2017). Multi-trip is also used to integrate constraints related to the staff schedule such as coffee breaks (Masmoudi et al., 2016), lunch breaks (Parragh et al., 2012; Liu et al., 2015; Zhang et al., 2015), and shifts between morning and afternoon teams (Parragh et al., 2012). For sanitary safety, multi-trip may be required to prevent the spread of diseases. When transporting a sick passenger, the vehicle must return to its depot for disinfection before picking up the next user (Carnes et al., 2013; Zhang et al., 2015). Finally, multi-trip naturally appears when studying consistency in multi-period DARP (Braekers and Kovacs, 2016; Tellez et al., 2022; Lindstrøm and Røpke, 2022).

Some problems allow **multi-load** requests. On the one hand, the load from a single request can be split into different vehicles (Parragh et al., 2015). This practice results in a more efficient use of vehicles' capacity. On the other hand, multiple requests can be grouped as a single request (Qu and Bard, 2015; Liu et al., 2015). This allows to decrease the size of the DARP.

In DARP with **transfers**, the pickup and drop-off of a user can be performed by different vehicles (Masson et al., 2014; Reinhardt et al., 2013). This implementation requires a transfer between the vehicles involved in the transportation. As transfers deteriorate the quality of service (i.e., transfers

themselves and waiting times at transfers' locations), synchronization between vehicles is required to limit users' inconvenience. However, this routing property cannot be applied for some users such as users in wheelchairs.

Similar to the previous variant with transfers, DAR systems can be **combined with regular public transportation**. In other words, users can take the tramway, bus, or subway for a portion of their trip. This practice reduces the costs as regular public transportation is cheaper. However, this also reduces the quality of service. Indeed, the waiting time can be significant when regular transportation is scarce (e.g., nighttime, rural areas) or public transportation can be crowded during peak hours. The waiting time can be reduced by synchronizing DAR systems and the regular public transportation systems (Posada et al., 2017). Nevertheless, the synchronization may be ignored if high-frequency regular public transportation is assumed (Häll et al., 2009).

Collaboration between DAR service providers have been investigated by few studies. Nevertheless, their findings offer valuable insights (Molenbruch et al., 2017a). In a situation where each depot belongs to a service provider, assigning users to service providers based on the proximity between pickup/drop-off locations and service providers' depots is not the best strategy to minimize the cost of serving all the demands. In Molenbruch et al. (2017a), the authors show that it is more efficient to consider all service providers and all requests as a whole.

The **electric vehicle** is a new technology that results in the emergence of new routing properties (Masmoudi et al., 2018; Molenbruch et al., 2023). Although electric vehicles are an interesting alternative to internal combustion vehicles to reduce the environmental footprint of DAR systems, their limited autonomy poses organizational challenges (i.e., battery management strategies, detours to charging stations, and recharge times).

DARP with **autonomous vehicles** is another variant that emerged recently (Pimenta et al., 2017; Bongiovanni et al., 2019; Bongiovanni et al., 2022; Su et al., 2023). In this variant, the use of autonomous vehicles removes the constraints related to the drivers' shifts and breaks. As a consequence, vehicles operate on non-stop schedules, increasing their level of service.

Specifications of service quality. The classic objective of DARP is the minimization of the transportation cost. In this case, the service quality is ensured by constraints on time windows and maximum ride times. However, other constraints related to service quality have been studied in variants of the DARP.

Time windows at both pickup and drop-off are required to ensure that users don't arrive too early and wait for too long at their drop-off location. Usually, users specify only one time window for the pickup or the drop-off. For example, in the case of an outbound trip from home to a medical facility, the user specifies the time windows at the drop-off location in order to arrive on time for the medical consultation. On the opposite, in the case of a disabled child returning home, the time window is set at the pickup location to match the closing time of the medico-social institution. Once the time window of the first location (i.e., pickup or drop-off) has been set, the time window of the second location is computed based on the user's maximum ride time. A procedure to compute the time windows has been introduced in Cordeau (2006).

The **maximum ride time** is the same for all users in the classic formulation of the DARP. However, this is inappropriate if some users have short trips and others have long trips. There are other ways to define the maximum ride time. First, the maximum ride time can be proportional to the direct ride time (Jaw et al., 1986). Second, if the time windows are specified at both pickup and drop-off locations, the maximum ride time may not be required (Wolfler Calvo and Touati-Moungla, 2011; Parragh et al., 2015). In this case, the ride time of the user is directly restricted by the time windows.

Incompatibilities may be added to the models to improve the quality of service (Detti et al., 2017; Molenbruch et al., 2017c). There are two types of incompatibilities: user-user and user-staff. These incompatibilities can result from medical requirements (e.g., qualification levels of the staff members) or from preferences mentioned by users.

Consistency has also been studied in the context of multi-period DARP. Indeed, users of regular on-demand transportation services are sensible to changes in their daily routine (Paquette et al., 2012; Tellez et al., 2020). It ensures that some aspects of transportation don't change too much from one day to the next. As shown in Paquette et al. (2012), staff consistency is the second most important

aspect of transportation. This ensures that users are already familiar with staff members. This aspect of transportation is especially important for some users (e.g., patients with autism) who only accept a few staff members known beforehand. Different levels of driver consistency are studied in Braekers and Kovacs (2016) by pairing users with a limited number of drivers over a time horizon. Time consistency is also important as some users and families are not always ready to accept a high variability in the transportation schedule (Feillet et al., 2014). In the context of on-demand transportation, the time-consistent DARP has been introduced in Tellez et al. (2022). Finally, group consistency has been studied in Lindstrøm and Røpke (2022). In the case of school children transportation, frequent group changes can be stressful. To solve this issue, each user can travel with a limited number of different users.

3.3.2.2 Alternative objectives

Single-objective. First, reducing the **cost** is the most studied objective. Costs can be explicitly expressed with the arc cost (Cordeau and Laporte, 2003) or with the vehicle costs (e.g., cost per kilometer, cost per minute, fixed cost) (Tellez et al., 2018). Costs can also be considered through distance minimization (Parragh et al., 2009; Diana and Dessouky, 2004; Wong et al., 2014) or duration minimization (Madsen et al., 1995; Chevrier et al., 2012; Ritzinger et al., 2016). A cost-related objective is the maximization the service providers' profit, which is equal to the revenue minus the transportation cost (Parragh et al., 2015; Molenbruch et al., 2017a). Other objectives are related to the transporter. For example, minimizing the number of vehicles is particularly helpful when vehicles have a high fixed cost (Madsen et al., 1995; Chevrier et al., 2012; Wong et al., 2014).

Second, **user-related** objective have also been studied to improve the comfort of users. Maximizing the quality of service is often measured using a metric related to ride times such as the minimization of total ride times (Paquette et al., 2013), average ride time (Parragh et al., 2009), maximum user ride time, or excess ride times (Melachrinoudis et al., 2007; Jorgensen et al., 2007; Diana and Dessouky, 2004). Other objectives consider waiting times (Madsen et al., 1995; Jorgensen et al., 2007; Paquette et al., 2013; Guerriero et al., 2014), transit times (Guerriero et al., 2014), deviations from requested pickup/drop-off time windows (Madsen et al., 1995; Melachrinoudis et al., 2007; Carotenuto and Martis, 2017), delays (Chevrier et al., 2012), or waiting time before the start of service (Krumke et al., 2006). An original objective is studied in Garaix et al. (2011). As DARP allows the sharing of vehicles, the authors try to maximize the occupancy rate of vehicles to promote social interactions between users.

Third, **environment-related** objective emerged with the need to develop green transportation. As such, objectives like the minimization of CO2 emissions (Atahran et al., 2014; Chen et al., 2022) or fuel consumption (Hu et al., 2019) have been studied.

Finally, **soft constraints** may be applied to penalize solutions without making them infeasible. Depending on the situation, soft constraints allow the violation of time windows (Jorgensen et al., 2007; Melachrinoudis et al., 2007; Melachrinoudis and Min, 2011; Chlebíková et al., 2023), maximum ride times (Chlebíková et al., 2023), or even rejection of some demands in dynamic DARP (Liang et al., 2020; Dong et al., 2020; Gaul et al., 2021). While soft constraints make it easier to reach feasible solutions, they can also lead to undesirable situations. For example, users might arrive late at their appointment or vehicles might leave before users are ready. In addition, setting the appropriate weight for the penalty requires good insights from the decision-makers.

Multi-objective. In cost-related mono-objective optimization, the quality of service is expressed through constraints. In this case, there is no incentive to improve it. In order to obtain solutions with more balanced trade-offs between cost and quality of service, researchers have used multi-objective strategies. To combine the different and often antagonist objectives mentioned in the previous section, multiple strategies have been developed.

First, **scalarization** is the simplest strategy. It combines as many objectives as desired using a weighted sum objective (Madsen et al., 1995; Jorgensen et al., 2007; Melachrinoudis et al., 2007; Mauri et al., 2009; Kirchler and Wolfler Calvo, 2013; Diana and Dessouky, 2004; Bongiovanni et al., 2019; Souza et al., 2022). In this strategy, each objective has a weight in the global objective function that expresses its importance. While this scalarization method is easy to implement, a good understanding of the tradeoffs between the objectives is needed to properly set the different weights.

Second, the **lexicographic method** optimizes the objective in decreasing order of importance (Garaix et al., 2011; Schilde et al., 2011; Schilde et al., 2014; Tellez et al., 2022; Ritzinger et al., 2022). As a result, the first objective optimized has more influence on the solution than the others. This method is inappropriate when searching for a fine trade-off between the objectives.

Third, the **Pareto frontier** reveals the best trade-offs between multiple objectives (Guerriero et al., 2014; Atahran et al., 2014; Hu et al., 2019). By definition, for any solution on the Pareto frontier, it is not possible to improve one objective without sacrificing another. Another way to describe a Pareto frontier is by using the notion of dominance. A solution is strongly dominated if it exists another solution that is better in all objectives, and weakly dominated if it exists another solution that is better in at least one objective and equal in the other objectives. Using dominance, the Pareto frontier of a problem contains the set of non-weakly dominated solutions. This method gives a very good insight into the trade-offs between the objectives. However, it is time-consuming to compute the solutions of the Pareto frontier.

3.3.2.3 Nature of decisions and information

The classic DARP assumes that all information is known in advance without uncertainty. In other words, the decision-maker has perfect information about the number of requests, the exact needs of each request, and the duration of each operation (routes and service times). Furthermore, no additional request is added after the start of the operations. Given the nature of the information, the classic DARP is static and deterministic. However, real-life on-demand transportation can be unpredictable and may need online adjustments. As a result, other variants of the DARP have been developed to tackle unpredictability and offer the possibility to make online adjustments.

First, a DARP can be static or dynamic. It is static if the routing plan cannot be changed after the start of the operations. On the opposite, a DARP is dynamic if adjustments can be made. Second, a DARP can be deterministic or stochastic. It is deterministic if the information is certain. On the opposite, a DARP is stochastic if the information is unknown or partially known (e.g., range of values, probabilistic distributions). Combining the static/dynamic and the deterministic/stochastic properties gives four types of DARP: static and deterministic, static and stochastic, dynamic and deterministic, and dynamic and stochastic. This taxonomy is summarized in Table 3.2 and explained in detail in Ho et al. (2018).

| decisions can be modified | information is known with certainty | designation |
|---------------------------|-------------------------------------|---------------------------|
| yes | yes | static and deterministic |
| yes | no | static and stochastic |
| no | yes | dynamic and deterministic |
| no | no | dynamic and stochastic |

Table 3.2: classification of DARP relative to the nature of the information

Requests. In real-life situations, requests can be unpredictable.

First, **online requests** may appear after the start of the operations. In this situation, the decision-maker must quickly respond by either accepting or rejecting the requests (Xiang et al., 2008; Berbeglia et al., 2012; Gaul et al., 2021). In case of acceptance, the routing plan is modified to accommodate the new request. If the new request has perfect information, then this corresponds to a dynamic and deterministic DARP.

Second, **stochastic requests** may be used for non-deterministic situations. For example, the exact end time of a medical appointment may be difficult to predict (Schildt et al., 2011). Nevertheless, some information can be used to make the routing plan such as the probability to end on time. This situation corresponds to a stochastic DARP. A similar source of uncertainty is user behavior. Indeed, users may be late or not show up at pickup locations (Xiang et al., 2008; Ho and Haugland, 2011; Heilporn et al., 2011). This also corresponds to stochastic DARPs.

Routing operations. There are also sources of uncertainty on the service provider side.

First, **traffic conditions** are influenced by many factors such as rush hour, traffic accidents, weather conditions, construction sites, road closures, and so on. This can lead to divergences between planned and real routing schedules. For example, traffic congestion can cause important delays. To a lesser extent, even traffic lights have an impact on the real schedule. These uncertainties can be modeled using stochastic travel times (Xiang et al., 2008; Schilde et al., 2014; Lu et al., 2022) or time-dependent travel times (Hu and Chang, 2015). Stochastic travel times correspond to stochastic DARPs, and time-dependent travel times correspond to deterministic DARPs. In the case of stochastic travel times, the problem is dynamic if it is possible to adjust the routing plan in response to the stochastic events.

Second, the **reliability of vehicles** is also a source of uncertainty. Stochastic vehicles' breakdown is studied in Xiang et al. (2008) and Beaudry et al. (2010) using a dynamic and stochastic DARP.

Third, **driver uncertainty** may also be introduced. For example, drivers may need unexpected breaks requiring online adjustments of the routing plan (Xiang et al., 2008; Beaudry et al., 2010). Service time may also be uncertain (Xiang et al., 2008). Indeed, staff members may take longer to serve a user compared to the theoretical service time. Stochastic service time can be introduced to take that into account.

3.3.3 Solution methods

Many solution methods have been used to solve the DARP and its variants. With slight modifications, some methods can solve multiple variants at the same time. In the following, we provide an overview of the classic solution methods (i.e., exact methods, heuristics, hybrid methods) used to solve the DARP. We also provide some insights into more recent approaches that use Machine Learning (ML) techniques. Again, we refer the reader to Molenbruch et al. (2017b) and Ho et al. (2018) for a more in-depth review of the classic solution methods.

3.3.3.1 Exact methods

Few articles use exact methods to solve the DARP. The main advantage of exact methods is their ability to find the optimal solution to a problem and guarantee its optimality. However, exact methods are computationally expensive. As a result, they are impractical for solving large DARPs, feature-rich DARPs, and DARPs with wide time windows. The largest classic DARP instance ever solved to optimality had 96 users and 8 vehicles, and the optimal solution was found in approximately 15 minutes (Gschwind and Irnich, 2015). For DARP variants, the computation time can take up to hours. As exact methods do not focus on speed but on quality, they have mostly been used for static and deterministic DARPs. Dynamic DARPs require fast decisions that are incompatible with exact methods. On the opposite, static DARPs may allow for enough computation time to reach optimality. For example, exact methods can be used when decisions are taken once a day and all requests are already known the day before. Three types of exact methods based on the branch-and-bound principle have been used to solve the DARP: branch-and-cut, branch-and-price, and branch-and-price-and-cut.

Branch-and-cut. The branch-and-cut runs a branch-and-bound algorithm in combination with the cutting plane method to tighten the linear programming relaxations. This allows faster convergence to integer solutions and gives better bounds for checking the optimality. Branch-and-cut was first applied to the DARP in Cordeau (2006). The authors adapted valid inequalities from the TSP and the VRP to a 3-index mixed-integer programming model. Later in Ropke et al. (2007), additional cuts (strengthened capacity, strengthened infeasible path, and fork constraints) were added and previous ones were adapted to a 2-index formulation.

In order to tackle variants of the DARP, other authors developed branch-and-cut algorithms with new valid inequalities related to their problems such as heterogeneous vehicles and users (Parragh, 2011), multi-depot (Braekers et al., 2014), configurable vehicles (Liu et al., 2015), trip number (Liu et al., 2015), lunch breaks (Liu et al., 2015), symmetry breaking (Braekers and Kovacs, 2016), and driver consistency (Braekers and Kovacs, 2016). Recently, Rist and Forbes (2021) proposed a new formulation for the DARP and new valid inequalities based on “restricted fragments” (segments of routes that can be present

in any route). Their method solved 9 instances of the literature for the first time and accelerated the computation time by one to two orders of magnitude for large instances.

Branch-and-price. Branch-and-price is based on the branch-and-bound algorithm combined with column generation. Compared to the branch-and-cut, this exact method can handle larger problems by decomposing the problem into a master and a subproblem. The branch-and-price was first used on the DARP in Garaix et al. (2011). In this work, the subproblem is solved using an exact method. However, the subproblem can be solved with any type of method: exact methods (Garaix et al., 2011; Parragh et al., 2012; Feng et al., 2014; Riley et al., 2019), heuristics (Coppi et al., 2013; Hu and Chang, 2015; Chen et al., 2022), metaheuristic (Parragh and Schmid, 2013), or hybrid methods (Parragh et al., 2015). On the one hand, using exact methods to solve the subproblem may be too computationally expensive but the global convergence of the branch-and-price is guaranteed. On the other hand, heuristic approaches are faster but the global convergence is not guaranteed. Hybrid methods propose a trade-off between the advantages and disadvantages of exact and heuristic approaches.

Branch-and-price-and-cut. The branch-and-price-and-cut is another variant of the branch-and-bound that combines the branch-and-price and the branch-and-cut approaches. This approach provides two benefits: first, the reduced problem is made smaller thanks to the column generation; second, the bounds of the problem relaxation are tightened using the cutting plane method. The method was first used on the DARP by Gschwind and Irnich (2015). They extended the work of Ropke and Cordeau (2009) on the pickup and delivery with time windows by adding the maximum ride time constraint. This method has been used by a few researchers such as Gschwind and Irnich (2015), Qu and Bard (2015), Luo et al. (2019), and Guo et al. (2023).

3.3.3.2 Heuristics

As DARP are NP-hard, it is difficult for exact methods to solve large or feature-rich DARP instances. As a result, research has revolved around the development of heuristics.

A wide variety of heuristics have been used to solve the DARP. Following the taxonomy used in Gendreau and Potvin (2005), some heuristics are trajectory-based and others are population-based. A trajectory-based heuristic operates on one solution at a time. It starts with an initial solution which is going to be modified at each iteration. Each time a new best solution is reached, this new best solution is saved. A population-based heuristic operates on multiple solutions simultaneously. At each iteration, the current population of solutions (current generation) is used to generate a new population of solutions (next generation). The principle is to give birth to the best solution from the selection of good characteristics across generations. In general, trajectory-based heuristics focus more on exploitation while population-based heuristics focus more on exploration.

Simulated annealing. Simulated annealing (SA) was introduced in Metropolis et al. (1953) and first applied to optimization problems by Kirkpatrick et al. (1983). It is based on the annealing process. Annealing is a heat treatment used in metallurgy in which a material is heated to a specified temperature and then cooled at a slow and controlled rate. Applied to optimization problems, simulated annealing works with a single solution. At each iteration, a new solution is generated. If it is better, the new solution is accepted. If not better, the new solution is accepted with a probability that depends on the temperature. High temperatures give high acceptance probabilities and low temperatures give low acceptance probabilities. At the beginning of the search, the temperature is high to stimulate exploration. Then, the temperature gradually decreases to stimulate exploitation.

The standard simulated annealing with simple neighborhood operators has been implemented with reasonable results (Baugh Jr et al., 1998; Mauri et al., 2009; Zidi et al., 2012; Reinhardt et al., 2013). In Braekers et al. (2014), the authors use a deterministic annealing algorithm with more advanced neighborhood operators and a restart procedure to avoid getting trapped in a local optimum. In this variant of the simulated annealing, the acceptance is deterministic and not based on probabilities. This work achieves very good results in comparison to other methods used in the literature.

Tabu search. Tabu search was introduced in Glover and McMillan (1986) and first used for the DARP in Cordeau and Laporte (2003). This trajectory-based metaheuristic excludes already visited search areas by temporarily placing them in a tabu list. The entire neighborhood of the current solution is generated and the best non-tabu solution is accepted. Thus, the tabu list avoids being trapped in cycles. In addition, the worst solutions can be accepted to avoid being stuck in local optima.

In Cordeau and Laporte (2003), the basic tabu search algorithm is enhanced by two mechanisms to speed up computation and escape from local optima. The first is the penalization of frequent moves, and the second is the possibility to accept worst solutions. Good results have been obtained with this algorithm. This encouraged other researchers to experiment with this metaheuristic and adapt it to variants of the DARP (Melachrinoudis et al., 2007; Beaudry et al., 2010; Ho and Haugland, 2011; Guerriero et al., 2014; Paquette et al., 2013; Kirchler and Wolfler Calvo, 2013; Detti et al., 2017).

Evaluating the complete neighborhood of a solution is computationally expensive. In order to overcome this, variants of the tabu search partially evaluate the neighborhood of the current solution. In Detti et al. (2017), a limited number of moves are randomly selected. In Kirchler and Wolfler Calvo (2013), a granular tabu search is used. Only moves with a reduced cost (i.e., cost estimation) inferior to a granular threshold are computed.

Variable neighborhood search (VNS). The variable neighborhood search was introduced in Mladenović and Hansen (1997). This metaheuristic is based on three actions: improve, shake, and change the neighborhood. The improvement phase (also called descent phase) is a variable neighborhood descent that aims to find the local optima. On the opposite, the shaking phase (also called perturbation phase) is used to escape local optima. Finally, the neighborhood change corresponds to the systematic change of operators used during the improvement and shaking phases. One advantage of the VNS is its simplicity as there are no (or few) parameters to set. Another one is the simplicity of creating variants. Indeed, operators can be specific to the problem, multiple cycling variants can be used in the neighborhood change (e.g., sequential, cyclic, pipe, skewed, ...), and multiple tactics can be used in the descent phase (e.g., steepest-descent, first-descent, stochastic-descent, ...).

Parragh et al. (2010) obtained great results on the DARP and has laid the foundation for other research on DARP variants (Parragh et al., 2009; Parragh, 2011; Parragh et al., 2012; Parragh et al., 2015; Schilde et al., 2011; Schilde et al., 2014; Muelas et al., 2013; Detti et al., 2017).

Large neighborhood search (LNS). The large neighborhood search was first introduced coupled with an exact method in Shaw (1998), and coupled with a heuristic under the name of ruin and recreate in Schrimpf et al. (2000). It has a very straightforward principle: it improves the current solution by ruining it and repairing it. At each iteration of the algorithm, the destroy operator degrades the solution, then, the repair operator rebuilds a new complete solution from the partial solution.

A very successful and widespread variant of the LNS is the Adaptive Large Neighborhood Search (ALNS) introduced in Ropke and Pisinger (2006). The main difference is that the ALNS selects the operators based on their weighted probabilities. The weight of each operator is dynamically adjusted according to its past performance. However, Turkeš et al. (2021) proved that adjusting the weights for the operator selection does not significantly improve the performance of the ALNS.

The LNS and its variants have been extensively studied and have often shown very good results on routing problems (Häll and Peterson, 2013; Qu and Bard, 2013; Lehuédé et al., 2014; Masson et al., 2014; Braekers and Kovacs, 2016; Gschwind and Drexler, 2019; Masmoudi et al., 2016; Ritzinger et al., 2016; Molenbruch et al., 2017a; Bongiovanni et al., 2020; Armbrust et al., 2022; Lindstrøm and Ropke, 2022). To our knowledge, the best results on classic DARP benchmarks have been obtained with a variant of the LNS by Gschwind and Drexler (2019).

Genetic algorithm. The genetic algorithm (GA) was first introduced in Holland (1975) and first used to solve the DARP in Rekiek et al. (2006). It is a population-based metaheuristic inspired by Darwin's theory of evolution. Several parents are selected from an initial population. Then, children are generated from the parents using crossover and mutation operators. Finally, some individuals (from both the parents and children) are selected to form the next generation.

The first genetic algorithms developed to solve the DARP used a cluster-first route-second principle (Rekiek et al., 2006; Jorgensen et al., 2007; Hanne et al., 2009; Cubillos et al., 2009). The genetic algorithm was used to build the clusters of requests, then, an additional heuristic was used to build the routes from those clusters. Later, researchers developed genetic algorithms that were able to entirely build the routes (Cubillos et al., 2007; Cubillos et al., 2009; Chevrier et al., 2012; Atahran et al., 2014). This approach requires managing the infeasibilities introduced by the genetic algorithm such as double requests, missing requests, and infeasible routes. Multiple crossover operators, mutation operators, and individual selection criteria exist, allowing researchers to explore many variants of the genetic algorithm.

3.3.3.3 Hybrid methods

Hybrid methods consist of any combination of a heuristic, a metaheuristic, or an exact method. A hybrid method combining exact and approximate methods is called *matheuristic*. While hybrid methods are more laborious to implement, they combine the advantages of multiple methods. They can be classified into two groups: hybrids of metaheuristics, and hybrids of metaheuristics and exact approach (i.e., *matheuristic*). For both groups, multiple structures may be used to combine the solution methods. When methods are on the same level, they may be run sequentially (Molenbruch et al., 2017a; Malheiros et al., 2021) or in parallel (Berbeglia et al., 2012). When methods are not on the same level, a primary method embeds other secondary methods (Jain and Van Hentenryck, 2011; Masmoudi et al., 2017).

Hybrids of trajectory and evolutionary methods. One popular combination is to integrate trajectory-based with population-based methods (Chassaing et al., 2016; Chevrier et al., 2012; Zhang et al., 2015; Masmoudi et al., 2016; Belhaiza, 2019; Fahmy, 2023). This allows to combine the exploitation qualities of the former with the exploration qualities of the latter. The trajectory-based method is applied to each individual in the population of solutions to mimic the biological phenomena of learning (Gogna and Tayal, 2013).

Matheuristics. Regarding *matheuristics*, two approaches have been studied: embedding a metaheuristic within an exact method or the reverse. The first is particularly used in the case of column generation where the pricing is performed by a metaheuristic (Parragh et al., 2012). In the second approach, the exact method is used within the metaheuristic to obtain a deeper level of intensification (Ritzinger et al., 2016; Gschwind and Drexl, 2019). To our knowledge, the hybrid method developed in Gschwind and Drexl (2019) is currently the best method for solving the classic DARP. The framework uses dynamic programming embedded in an adaptive large neighborhood search.

3.3.3.4 Machine Learning (ML)

This section briefly presents the literature on Machine Learning (ML) applied to Operations Research (OR). A more detailed review is included in Chapter 4.

In recent years, interactions between the artificial intelligence and operations research communities have grown rapidly. However, researchers have used machine learning to solve OR problems since the 1980s and 1990s (Smith, 1999). For example, Hopfield and Tank (1985) was the first to use a neural network to solve the TSP. Today, it is the recent increase in computational power that is currently driving the interest of the OR community toward ML techniques. There are two main uses of machine learning to improve OR's solution methods. The first is to replace time-consuming tasks with an approximation made using an ML method. The second is to explore the space of decisions (hyperparameters, parameters, or decision variables) using machine learning to learn the best behaviors.

Replacing time-consuming tasks. Usually, when machine learning is used to replace time-consuming tasks, the ML method assists a main OR method by making lower-level decisions. In particular, this has been used in combination with branch-and-bound. Since branch-and-bound is an enumerating method, it is very time-consuming. Several tactics are used to speed up the computation and avoid unnecessary enumerations. Machine learning can help to further improve these tactics. For example, it can be used to select nodes and variables in the branch-and-bound tree (Lodi and Zarpellon,

2017), to decide when to run primal heuristics to tighten the bound of the problem (Khalil et al., 2017), and to select promising cutting planes (Baltean-Lugojan et al., 2018).

Exploring the space of decisions. When machine learning is used to explore the space of decisions, it can be integrated at multiple levels.

At the highest level, end-to-end ML models can solve entire OR problems (Vinyals et al., 2015; Larsen et al., 2022; Bengio et al., 2021). However, end-to-end learning does not benefit from the decades of experience learned by the OR community. While promising, results from end-to-end models are still lacking compared to state-of-the-art pure OR solution methods.

Another possibility is to use machine learning for parameter tuning. Indeed, the performance of many heuristics depends on the proper adjustment of their hyperparameters (Bengio et al., 2021). However, there is no established standard practice for parameter tuning in the OR community. Many parameters are tuned based on trial and error, and on the expertise of researchers. Machine learning could be the appropriate tool to automate this process (Hoos, 2011; Bischl et al., 2016).

Similarly, machine learning can be used for heuristic selection (Malitsky et al., 2013; Xu et al., 2008; Gomes, 2001; Chen and Tian, 2019). Many metaheuristics use a portfolio of lower-level heuristics, each of which has its strengths and weaknesses. A good heuristic selection mechanism exploits the complementarity between heuristics to achieve the best performance (Kerschke et al., 2018). Again, instead of using handcrafted algorithms, machine learning could automate this process.

3.4 Research gaps relative to research questions

With respect to the literature, we identified several gaps that are relevant to the research questions related to the regular non-urgent transportation of disabled children in France.

3.4.1 RQ1: How can we build an algorithm that computes fast and optimal solutions for the regular non-urgent sanitary transportation?

There is a large variety of methods to solve the DARP: exact methods, metaheuristics, hybrid methods. In addition, the advances of machine learning has given access to a broad new set of tools for solving this routing problem. Nevertheless, the computation speed and the quality of the results are still not sufficient to solve problems with as many users as in real-life situations. Indeed, it takes almost ten minutes for the best solution method (Gschwind and Drexler, 2019) to solve a problem with 144 users (i.e., instance pr20 from Cordeau and Laporte (2003)). As the sanitary transportation of disabled children involves thousands of users and as the computation time increases exponentially with the size of the problem, even the best methods are still not able to manage real-life instances as they are. For example, they require slicing large instances into smaller sub-instances. Hence, our first research gap:

RG1: The current state-of-the-art solution methods for the DARP are not able to tackle real-life sized instances.

In order to build better solution methods, it would be beneficial to have a better understanding of the characteristics of DARP solutions. While some good characteristic can be identified intuitively, there may be others that cannot. In Arnold and Sörensen (2019), the authors successfully extracted the characteristics of good VRP solutions and used them to significantly speed up the resolution. To our knowledge, no similar study exist for the DARP. Hence, our second research gap:

RG2: The characteristics of good DARP solutions have yet to be identified through a rigorous process and used to solve the DARP.

The last research gap that could help us solve our first research question is related to the new methods brought by machine learning. As the ML and OR communities are distinct, new ML methods are not systematically tested on OR problems and on the DARP in particular. Moreover, ML algorithms, when

applied to DARP, are not always tested on classic benchmark from the literature. As a result, establishing a proper comparison with classic OR solution methods is difficult. Hence, our third research question:

RG3: Many machine learning methods have yet to be tested on the DARP and compared to state-of-the-art solution methods from the operations research literature.

3.4.2 RQ2: How can we extend the scope of the optimization and consider the different aspects related to all the stakeholders?

Many articles focus solely on solving the DARP or its variants with the objective of minimizing the transportation costs. However, the regular non-urgent sanitary transportation of disabled children in France goes beyond this sole objective. Indeed, the quality of service is at least as important as the costs. Fortunately, many articles have brought ingenious ways to consider the quality of service, whether in the context of single-objective or multi-objective optimization. But unfortunately, there are only few models that extend the scope of the resolution to other elements revolving around the transportation (e.g., staff scheduling, user assignment, meeting points, ...). The ultimate goal would be to develop a holistic model and the corresponding solution method that integrates all aspects related to the sanitary transportation. While this goal may be unreachable, it is possible to partially bridge the following research gap by integrating only some aspects.

RG4: There is no model that encompasses the entire scope of the regular non-urgent sanitary transportation of disabled children in France and considers all the aspects related to the different stakeholders.

3.4.3 RQ3: How the dependence between requests can be used to improve the transportation?

The classic DARP formulation consider all requests as independent. However, in the context of regular sanitary transportation, a user often have multiple requests. On one hand, the requests can still be considered as independent. On the other hand, the requests can be considered as dependent. Having depend requests means that the decision on one requests has an influence on the decision on another one. Binding multiple requests together can be obtained through constraints shared by the requests. For instance, this is the case of DARP variants that consider consistency. Shared constraints could take multiple forms. Indeed, they could apply to only the user's requests or to any set of requests regardless of the user. For instance, an individual maximum ride time constraint could apply to the requests of a user and a global maximum ride time constraint could apply to all the users. In regard to constraint sharing, we identified the following gap in the literature:

RG5: There are few works on shared constraints for the DARP. In particular, there is no study on a shared maximum ride time constraint.

3.5 Contributions

The research done during this thesis is built based on the gaps identified in the literature:

- RG1: The current state-of-the-art solution methods for the DARP are not able to tackle real-life sized instances.
- RG2: The characteristics of good DARP solutions have yet to be identified through a rigorous process and used to solve the DARP.
- RG3: Many machine learning methods have yet to be tested on the DARP and compared to state-of-the-art solution methods from the operations research literature.

- RG4: There is no model that encompasses the entire scope of the regular non-urgent sanitary transportation of disabled children in France and considers all the aspects related to the different stakeholders.
- RG5: There are few works on shared constraints for the DARP. In particular, there is no study on a shared maximum ride time constraint.

This thesis is divided into three chapters to address these issues. Each chapter attempts to fill some of the identified gaps.

- The first part of Chapter 4 explores the use of machine learning to build a knowledge-guided search method for the DARP. (RG1, RG2, RG3)
- The second part of Chapter 4 adapts an online ML method called Nested Rollout Policy Adaptation (NRPA) to the DARP. (RG1, RG3)
- Chapter 5 introduces the Commuting Dial-A-Ride Problem (Com-DARP) as a variant of the Dial-A-Ride Problem (DARP) where each user has a morning request, an afternoon request, and a daily maximum ride time. We accelerate the resolution by using a novel speed-up mechanism that exploits the dynamic precedence relations between users. (RG1, RG5)
- Chapter 6 introduces the Assignment Dial-A-Ride Problem (ADARP). This variant of the DARP combines the routing, user assignment, and staff sizing problems in the context of transportation of people with disabilities. (RG4)

Chapter 4

Machine learning applied to the dial-a-ride problem

4.1 Introduction

Machine Learning (ML) is a branch of artificial intelligence and computer science that imitates the way humans learn by using data and algorithms. ML algorithms do not need to be explicitly programmed to perform a specific task. This characteristic is a fundamental difference from classic algorithms. It makes machine learning well suited for natural signals (Bengio et al., 2021). Indeed, they are perfect candidates for solving problems with no clear mathematical formulation or where the data distribution cannot be described analytically. The first work on intelligent machines can be traced back to the 1950s (Solomonoff, 1957). In 1987, the Traveling Salesman Problem (TSP) was first solved using a neural network (Hopfield and Tank, 1985). However, the recent progress has been driven by the explosion of available data and the increase in inexpensive computational power. Today, ML models show good results across a variety of tasks such as data mining, image processing, predictive analytics, natural language processing, and so on. As a consequence, it is becoming increasingly popular in sectors such as industry, transport, energy, health care, services, and so forth.

Based on the way humans learn, ML models need to go through a learning process illustrated in Figure 4.1 (Wang et al., 2009). First, the model receives information from its environment. Second, it extracts knowledge from this information. Thirdly, it saves the acquired knowledge. Finally, the stored knowledge is used to execute new tasks. Additionally, a feedback loop can be used to adjust the model iteratively.

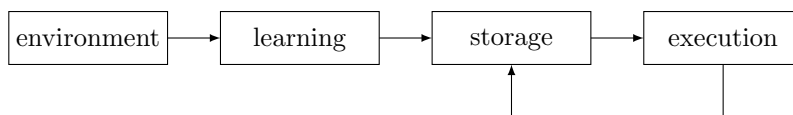


Figure 4.1: learning process (Wang et al., 2009)

In this chapter, we present two ML-based approaches to solve the DARP. The first solution method is based on an offline learning method. In this method, the characteristics of good and bad solutions are expressed using multiple metrics. Then, several classifiers are trained to discriminate good and bad solutions based on these metrics. Next, new ruin and recreate operators are developed based on the knowledge gained from this learning. Finally, the new operators are integrated into a Large Neighborhood Search (LNS) metaheuristic. The second solution method is based on an online learning method. It uses a variant of the Monte Carlo Tree Search (MCTS) called Nested Rollout Policy Adaptation (NRPA). Across multiple iterations, the algorithm learns the best way (i.e., policy) to build a complete solution. Starting from an empty solution, the NRPA iteratively selects one request to insert using a greedy insertion heuristic. After a solution is complete, the policy is adapted according to the best solution found so far.

The rest of this chapter is organized as follows. A literature review is presented in Section 4.2. The first work using offline learning is detailed in Section 4.3. Then, the second work using online learning is detailed in Section 4.4. For both works, we give a complete description of the solution methods and the numerical experiments. We also provide the conclusions and perspectives from these studies.

4.2 Literature review

4.2.1 A brief overview of machine learning

An ML problem is defined by a task, a learning process, and a performance measure (Alzubi et al., 2018). For example, distinguishing several species of iris flower is a problem where the task is to classify all the individual iris, the learning process is supervised learning using a set of labeled iris, and the performance measure is the number of iris correctly classified.

ML models essentially deal with three tasks: classification, regression, and clustering (Ray, 2019). A classification model tries to find the only correct output from a list of output classes. A regression model tries to find the correct output value from a continuous numeric interval. A clustering model seeks to explain a dataset by identifying a finite number of clusters based on the similarities between its items. In Alzubi et al. (2018), generic questions are associated with these tasks: classification answers the question “is it A or B?”, regression answers the question “how much?” or “how many?”, and clustering answers the question “how is it organized?”.

Several types of learning processes exist to train ML models. The three major paradigms are supervised, unsupervised, and reinforcement learning. In supervised learning, for any input, its corresponding output is available. The model builds a mapping structure to produce an output (or a probability distribution) for any given input based on input-output examples. In that sense, supervised learning is also called imitation or demonstration setting. In unsupervised learning, the goal is to understand the underlying characteristics of the data. On the opposite of supervised learning, the data is unlabeled. In reinforcement learning, the model learns by interacting with its environment. It receives positive or negative feedback for each action (or sequence of actions). The input corresponds to the interaction with the environment and the output corresponds to the feedback.

The performance measure in machine learning is essential. During the training phase, it allows to track the learning progress of a model. During the test phase, it allows to evaluate the quality of the model. As many performance metrics exist, they must be chosen according to the problem. For classification tasks, the most common metrics are accuracy, precision and recall, and f1-score. For regression tasks, these are mean absolute error, mean squared error, and root mean squared error for regression tasks. Finally, for reinforcement learning tasks, these are cumulative reward and time to reach a certain performance. In addition, task-specific metrics are also used in reinforcement learning (e.g., the number of wins in games or the number of positive feedback in natural language processing).

Machine learning and Operations Research (OR) are two specific fields with two specific communities. Despite this distinction, limitations from one domain can be tackled using techniques from the other domain. This has encouraged interactions between the two research fields. Thanks to the recent progress in machine learning, the interactions between machine learning and operations research are increasing. Four types of interactions can be identified (Song et al., 2019; Gambella et al., 2021). First, machine

learning can be used to solve OR problems, and more generally optimization problems. Second, the reverse is also possible: operations research can be used to solve ML problems or to enhance ML techniques. Third, data predictions using ML techniques can be used in decision-making processes. Fourth, machine learning can be treated as an optimization problem where the objective function is the performance metric.

4.2.2 Machine learning applied to operations research

Many articles review the interplay between the two fields (Corne et al., 2012; Curtis and Scheinberg, 2017; Song et al., 2019; Kraus et al., 2020; Gambella et al., 2021; Boggyrbayeva et al., 2022). In this literature review, we focus on the first type of interactions: machine learning for solving OR problems.

Many effective solution methods have been developed by the OR community over decades of research. Currently, the state-of-the-art algorithms are often composed of hand-crafted heuristics. Designing solution methods often relies on the researcher’s experience as it is difficult to find optimal configurations. Machine learning could provide a more systematic approach to the building and tuning process. The underlying goals behind using machine learning for operations research are to speed up computations and to improve the quality of the solutions. Machine learning has been used for operations research since the 1990s. However, due to the lack of computational power, it was not practical compared to classic OR methods such as exact approaches and metaheuristics (Smith, 1999). Today’s renewed interest in this research topic is fueled by an unprecedented amount of computational resources brought by the technological advances of the past decade.

The use of an ML technique for solving an OR problem can be described by answering three questions: “what is the goal?”, “what kind of knowledge is used?”, “how is it used?”. First, as explained previously, the goal is often to speed up computations, improve the quality of the solutions, or a combination of the two goals. Second, the knowledge extracted by ML models can be used either online or offline. Online models learn continuously during the resolution of the OR problem (Zhang and Lu, 2008). On the opposite, offline models learn during a training phase. Then, the extracted knowledge is used on unseen instances (Yalcinoz and Altun, 2001; Arnold and Sörensen, 2019). Third, there are many ways to use machine learning to solve OR problems. Three main uses can be identified: generate instances, approximate time-consuming tasks, and learn the best policies (i.e., strategies to solve a problem). We detail the different uses below.

4.2.2.1 Uses of machine learning for operations research

Instance generation. The first use is to generate instances (Song et al., 2019). Many applications of operations research aim to solve real-life problems. However, the number of real-life instances is limited in many cases. This lack of available instances is also apparent for some synthetic instance benchmarks. Furthermore, even when the number of instances is quite high, the benchmark may lack diversity. The lack of diversity and quantity raises several questions regarding the evaluation of solution methods: “is the performance of the solution methods specific to the benchmark used?”, “does the performance of the solution method translate from synthetic to real-life instances?”. As a result, generating synthetic instances could consolidate the results obtained by solution methods and prevent a decrease in performance during the deployment on real-life instances (Lopes and Smith-Miles, 2013). To ensure this, the synthetic instances should be numerous, close to real-life instances, and should cover as many significant regions of the instance space as possible. Machine learning helps to achieve this goal by analyzing the differences between instances. Ideally, the instances should form a large number of clusters mixing real-life and synthetic instances.

Time-consuming tasks approximation. The second use of machine learning in operations research is to substitute time-consuming tasks with fast approximations. In this architecture, the OR primary method calls an ML secondary method to assist in lower-level decisions (Bengio et al., 2021). This use is mostly applied to exact methods based on branch-and-bound.

ML models can select variables and nodes for branching (Lodi and Zarpellon, 2017). Here, the goal is to replace handcrafted heuristics with a more systematic approach and speed up the selection process.

Machine learning can also decide when to run primal heuristics (Khalil et al., 2017). On the one hand, primal heuristics help pruning the branching tree when they find tighter bounds. But on the other hand, they can slow down the resolution if launched too frequently without reaching tighter bounds. As a result, ML models could learn the best moments to launch them. Finally, machine learning can also select promising cutting planes (Baltean-Lugojan et al., 2018).

Outside the context of branch and bound, machine learning is also used to evaluate solutions (Song et al., 2019). Indeed, ML can be used to approximate the objective function (Li et al., 2011; Li et al., 2012; Václavík et al., 2016) or reduce the number of solutions to evaluate (Hunger and Huttner, 1999; Hee-Su Kim and Sung-Bae Cho, 2001; Park and Kim, 2017). This practice speeds up the computation when evaluating the solution objective is time-consuming or when there is a large number of solutions to evaluate.

Policy learning. The third use of machine learning in operations research is policy learning. In other words, ML models learn the best strategies to solve the OR problem. Models can be integrated into multiple algorithmic structures.

1) At the highest level, machine learning can heuristically solve problems by itself. This is called end-to-end learning (Vinyals et al., 2015; Bengio et al., 2021; Larsen et al., 2022).

2) Still at a high level, hyperparameters and parameters can be tuned using machine learning (Zhang et al., 2007; Hoos, 2011; Bischl et al., 2016). To our knowledge, there is no tuning method widely used by the OR community. These parameters are often set manually through trial and error during the development phase. Nevertheless, these parameters have an impact on the performance of solution methods (Bengio et al., 2021). As a result, machine learning could provide a more systematic approach and better tuning.

3) Machine learning can also be used to help the selection of decision variables in OR problems (Song et al., 2019). In the context of evolutionary multi-objective optimization, clustering has been used to classify decision variables between convergence-related and diversity-related variables (Zhang et al., 2018). This practice aims to improve the scalability relative to the number of decision variables.

4) A well-studied approach in using machine learning for solving OR problems tackles the selection process of competing heuristics (Gomes, 2001; Burke et al., 2003; Xu et al., 2008; Zhang and Lu, 2008; Di Gaspero and Urli, 2012; Malitsky et al., 2013; Degroote, 2018; Kerschke et al., 2018; Choong et al., 2018; Chen and Tian, 2019). This is also known as meta-learning for algorithm portfolios (Song et al., 2019; Bongiovanni et al., 2022). For many optimization problems, no heuristic dominates the others across all instances. Indeed, heuristics have complementary strengths that can be leveraged by using the right heuristic at the right moment (Kerschke et al., 2018).

5) The selection of solutions in the context of population-based metaheuristics is another use of machine learning related to a selection process. As maintaining an entire population can be computationally expensive, machine learning can be used to select only the most important individuals. Some studies aim to maintain population diversity (Streichert et al., 2003). Others aim to select the most promising regions by applying intensification only on the most promising solutions (Martinez-Estudillo et al., 2006; Wong and Choong, 2015).

6) Machine learning can be used to generate solutions. For example, it can be used at the beginning of the resolution to generate better initial solutions (Deng et al., 2015) or during the search as a part of the solution method (Montoya et al., 2016).

7) Finally, machine learning can be used for neighbor generation by exploiting acquired knowledge. Instead of enumerating all neighbors or randomly generating neighbors, ML models can limit and guide the search toward interesting neighbors (Arnold and Sörensen, 2019; Thevenin and Zufferey, 2019).

4.2.3 Machine learning for routing problems

Solving routing problems using machine learning perfectly illustrates the interactions between the OR field and the ML field.

While routing problems have been tackled since the early days of operations research, the first use of machine learning for solving a routing problem only appeared more than 40 years later. Hopfield and Tank (1985) was the firsts to solve the TSP using a Hopfield neural network. Until the mid-2010s,

similar works were scarce. However, recent years have seen a growing number of studies using machine learning for solving routing problems. Figure 4.2 illustrates this trend. It represents the number of results per year on the Science Direct database for a research query with keywords related to machine learning and routing problem¹. While this search is conducted on a single database and is not fine-tuned, it is still representative of the global trend.

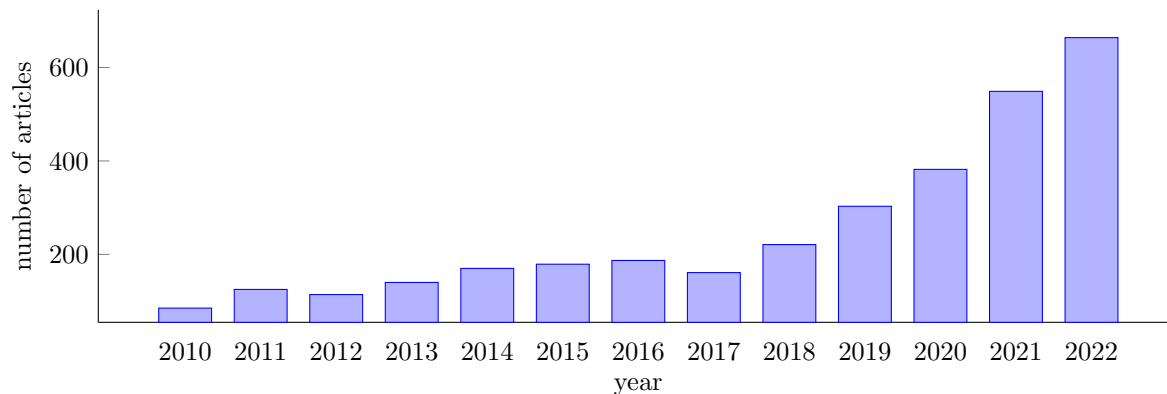


Figure 4.2: number of articles related to machine learning and vehicle routing problems per year

While this research topic lies at the intersection of two research fields, the two communities use different approaches. On the one hand, articles from the machine learning community mostly focus on end-to-end resolution entirely based on ML techniques (Vinyals et al., 2015; Bello et al., 2017; Khalil et al., 2017; Gao et al., 2020; Vera and Abad, 2019; Yu et al., 2019; Basso et al., 2022; Baty et al., 2023; Ma et al., 2022; Li et al., 2022; Kool et al., 2019; Nazari et al., 2018; Joshi et al., 2019; Kim et al., 2021; Chen and Tian, 2019; Wu et al., 2020). On the other hand, articles from the OR community often use machine learning as a complementary tool in an attempt to enhance the performance of metaheuristics and exact methods (Bongiovanni et al., 2020; Bongiovanni et al., 2022; Lu et al., 2019; Li et al., 2021; Arnold and Sørensen, 2019; Accorsi and Vigo, 2021).

This difference can be explained by the difficulty to tackle a multidisciplinary problem such as solving routing problems using machine learning. Multiple difficulties can be identified (Accorsi et al., 2022). First, learning and applying techniques from the other research field requires considerable effort. Second, researchers from one field may not be familiar with the literature and practices of the other field. As a result, the best solution methods are not always considered. For example, Nazari et al. (2018) use Google’s OR-tools to compare their results on the VRP. However, superior solution methods are available for the VRP. Third, comparing results within one research field is not always easy, but it is even more difficult across two fields. Indeed, the ML and OR communities tend to use different benchmarks, programming languages (e.g., Python vs. C++), and running environments (parallel GPUs vs. single-thread CPU). Finally, despite the growing efforts on this topic, results obtained using machine learning are still not on par with state-of-the-art OR solution methods. Unfortunately, this hinders the adoption of ML techniques.

Among routing problems, the Dial-A-Ride Problem (DARP) is one of the most advanced variants. In comparison to the classic TSP, there are significantly more constraints. The most challenging constraints are the time windows, the node pairing (i.e., pickup and drop-off), and the maximal ride time constraints. In a TSP, any node permutation results in a feasible solution. However, in a DARP, only a few node permutations are feasible due to the previously mentioned constraints. Thus, the space of the feasible solutions is often extremely small compared to the rest of the research space. This feasibility issue poses such challenges that some articles are almost entirely dedicated to it (Hunsaker and Savelsbergh, 2002; Tang et al., 2010; Firat and Woeginger, 2011; Braekers et al., 2014; Gschwind and Drexler, 2019). This additional hurdle adds to the previously mentioned difficulties in solving routing problems using ML techniques. As a result, research using machine learning to solve the DARP and its variants is scarce

¹The complete query is: (“machine learning” OR “reinforcement learning” OR “neural network”) AND (“traveling salesman” OR “vehicle routing” OR “dial-a-ride”)

(Moll et al., 1998; Bongiovanni et al., 2020; Bongiovanni et al., 2022; Ackermann and Rieck, 2022; Dong, 2022; Mortazavi et al., 2022). To our knowledge, only Dong (2022) and Mortazavi et al. (2022) have recently addressed the classic DARP problem using machine learning.

4.3 Knowledge-guided search for the DARP using offline machine learning

Highlights:

- The Dial-A-Ride Problem (DARP) is solved using an offline machine learning based method. The solution method extracts the characteristics of good DARP solutions, and then builds knowledge-guided heuristics targeting these characteristics.
- Two scopes are studied to understand what are the characteristics of good solutions (1st scope) and good arcs (2nd scope).
- The learning process is performed on 1000 synthetic instances, and then the new knowledge-guided heuristics are tested on classic instances from the literature.
- The classifiers are able to differentiate good from bad solutions/arcs with a 67/66% accuracy. However, the implementation of the new knowledge-guided heuristics does not improve the resolution.

4.3.1 Introduction

This section presents an offline learning method for solving the DARP. The goal is to extract the characteristics of good DARP solutions or arcs using ML techniques, and then use these characteristics to create new heuristics. Due to the complexity of routing problems, building a solution by hand is far from being the optimal method for any large problem. Yet, to a certain extent, it is possible to intuitively know if a solution is good or not for some simple routing problems such as the TSP and the VRP. However, the DARP is much more sophisticated, making this task impossible. This is mainly due to time-related constraints. For example, two nodes geographically close might not be in the same route because they are temporally far away. Hopefully, machine learning could be the perfect tool to make sense of the intricacy of DARP solutions. Indeed, machine learning is capable of identifying underlying patterns in databases with large and nontrivial structures. Using these patterns to guide the resolution would speed up the computation. As a result, it would be possible to solve larger DARP instances.

Many solution methods are available to solve the DARP. Exact methods guarantee the optimality of solutions, but their effectiveness is limited by instance size (Ho et al., 2018). For large instances or restricted computation time, metaheuristics present a more practical approach. Among these metaheuristics, the Large Neighborhood Search (LNS) and its variants have consistently shown good performances across a variety of DARPs (Pisinger and Ropke, 2019). Based on the principle of ruin and repair, the metaheuristic iteratively calls for lower-level heuristics (called operators) for inserting or removing requests (Shaw, 1998). Over the years, researchers have developed many operators. Some of these (e.g., greedy insertion, random removal) are problem agnostics and could be used on other optimization problems. Others are DARP-specific (e.g., related removal) and try to take advantage of the DARP's structure. Nevertheless, these knowledge-based operators were built based on the intuition and experience of the researchers. A data mining approach could be a systematic method to provide new insights, and thus build new knowledge-based operators for the DARP.

In Arnold and Sörensen (2019), this solution method has been successfully implemented on the VRP. The authors discovered that near-optimal and non-optimal VRP solutions can be distinguished by the compactness of routes, the width of routes, the span in radians of routes, the number of intersecting edges, and the distances of connecting edges to the depot. Then, they used the extracted knowledge to penalize edges based on these characteristics. As a result, the convergence speed of their metaheuristics has greatly improved. In Arnold and Sörensen (2019), the learning process operates at solution level. In other words, the metrics used as inputs for the ML process describe entire solutions. In our case, we study both the learning at solution level and at arc level. At arc level, the goal is to identify the characteristics of good arcs (i.e., arcs present in good solutions). Then, we use this knowledge to guide the resolution toward the insertion of good arcs.

In this study, we try to bridge research gaps RG1², RG2³, RG3⁴ by answering two questions. First, is it possible to identify the characteristics of good DARP solutions/arcs? Second, does the introduction of knowledge-based operators in an LNS speed up the resolution of a DARP?

The rest of this section is organized as follows. The solution method is presented in Section 4.3.2. Then, the experimental application of each of its steps is detailed in Sections 4.3.4, 4.3.5, and 4.3.6. Finally, conclusions and research perspectives are drawn in Section 4.3.7.

4.3.2 Solution method

The solution method has three steps: 1) generate the data, 2) extract interesting characteristics, and 3) create new operators based on the acquired knowledge.

Data generation. The first step is to generate enough data to learn from.

A large number of artificial instances are created and solved to generate solutions. This step is required as the number of instances from benchmarks of the literature is limited. Then, one good and one bad solution is computed for each instance. At solution level, the good and bad solutions directly serve as inputs for the ML process. At arc level, it is the arcs from these solutions that serve as inputs. In this case, arcs present in the good (respectively bad) solution are labeled as good arcs (respectively bad arcs). Finally, metrics that characterize instances, solutions, and arcs are extracted to build the database. While there is no guideline on choosing the metrics (Arnold and Sørensen, 2019), we rely on our experience and intuition to create them. As a result, we use an exhaustive list of metrics to increase the likelihood of discovering underlying patterns.

Once the metrics are computed, they are placed into a dataset. At solution level, each data point contains one solution's metrics, the corresponding instance's metrics, and the label (i.e., good or bad). At arc level, a solution's metrics are replaced by an arc's metrics. Instances' metrics are included because a solution's structure depends on the instance's structure. For instance, the average route load in a given solution is certainly influenced by the number of requests in the instance.

Characteristic extraction. The second step is to analyze the data using machine learning to extract interesting characteristics. Two questions are raised here.

The first question is "is it possible to differentiate good and bad solutions/arcs using the extracted metrics?". To answer this question, classifiers are trained using all the available metrics. A prediction accuracy of 50% is equivalent to randomly guessing the label. Thus, if the accuracy of the classifiers is greater than 50%, then the classifiers are able to differentiate good from bad solutions/arcs based on their metrics.

If the first question is positively answered, then the second question can be raised: "what are the characteristics of good solutions/arcs?". This question concerns the interpretability of the predictions which depends on the classifiers internal mechanism and the number of metrics used for making the predictions. Unfortunately, accuracy and interpretability are often two opposite objectives. While using all the metrics and advanced classifiers (e.g., deep neural networks) often leads to superior performance, the results are difficult to interpret. On the contrary, relying solely on one solution metric and basic classifiers (e.g., decision trees) can result in easier interpretation of predictions but often leads to lower accuracy. As a consequence, we must find a trade-off to extract usable knowledge from the data. Nevertheless, it is important to keep in mind that the objective of this study is to extract guidelines for solving the DARP, not to have the highest accuracy score.

Building knowledge-guided operators. The last step is to incorporate the acquired knowledge into the metaheuristics used for solving the DARP. In Arnold and Sørensen (2019), the authors used the extracted knowledge to penalize the edges. In our case, we decided to create new knowledge-based

²RG1: RG1: The current state-of-the-art solution methods for the DARP are not able to tackle real-life sized instances.

³RG2: The characteristics of good DARP solutions have yet to be identified through a rigorous process and used to solve the DARP.

⁴RG3: Many machine learning methods have yet to be tested on the DARP and compared to state-of-the-art solution methods from the operations research literature.

heuristics. Incorporated in a metaheuristic framework as low-level heuristics, the new heuristics insert and remove users based on the characteristics used for the predictions.

4.3.3 Presentation of the numerical experiments

Solutions are generated using a solver coded in C++ and compiled into a 64-bit single-thread code using g++ 9.4.0. Then, the data is analyzed with classifiers from the scikit-learn library. Finally, the C++ solver is modified according to the extracted knowledge and tested to see if any improvement has been made. The experiments are performed on a 2.0 GHz AMD EPYC 7702 CPU under a Linux Ubuntu 20.04.4 LTS operating system.

In the following numerical experiments, we first describe the solver used to generate the solutions in Section 4.3.4.1. Then, Sections 4.3.4.2 and 4.3.4.3 detail the instance and solution generation processes. It is directly followed by the presentation of the metrics used during the learning in Section 4.3.4.4. Next, the machine learning part of the experiment is detailed in Sections 4.3.5.1 and 4.3.5.2. Finally, the new knowledge-guided heuristics are tested in Section 4.3.6.

4.3.4 Data generation

4.3.4.1 The solver: Large Neighborhood Search (LNS)

As the DARP is NP-hard (Baugh Jr et al., 1998), solving the generated instances to optimality is not an easy task. While using an exact method would ensure optimality, it would require too much computation time. As a consequence, we use a metaheuristic. Among the variety of existing metaheuristics, we implemented a Large Neighborhood Search (LNS). The LNS is simple to implement and has shown great results over a variety of DARP benchmarks (Pisinger and Ropke, 2019). The adaptive large neighborhood search is a more widely used variant of the LNS which has an additional adaptive layer. However, this adaptive layer shows little improvement in practice (Turkeš et al., 2021). As a consequence, we decided to use the simpler LNS as our solver.

A wide variety of ruin and recreate operators have been implemented (Pisinger and Ropke, 2007). However, Christiaens and Vanden Berghe (2020) showed that good results can be obtained with a few operators. Thus, the LNS is implemented with the same operators as in Tellez et al. (2018) as they showed the best results on a similar metaheuristic implementation. Our LNS uses two ruin operators (random removal, historical node-pair removal) and 4 recreate operators (best insertion, k-regret insertion with $k = 2, 3, 4$). The solution acceptance uses a simple record-to-record criterion (Dueck, 1993) as its performance has been proven for the adaptive large neighborhood search (Santini et al., 2018). Finally, the feasibility of the solution is validated using a set of preliminary tests (Braekers et al., 2014) followed (if successful) by a scheduling evaluation (Tellez et al., 2018). The LNS implementation is summarized in Table 4.1.

| value | description |
|---------------------------|---|
| 5% | record-to-record acceptance percentage |
| [0.1, 0.45] | destruction size for the LNS |
| random, history node-pair | ruin operators |
| best, k-regret | recreate operators |
| 6 | roulette wheel parameter for the historical node-pair ruin operator |
| 2, 3, 4 | k parameter for the k-regret ruin operator |

Table 4.1: LNS parameter setting

4.3.4.2 Instance generation

In our study, 1000 instances are generated using the same protocol as in Cordeau (2006) (benchmark A). Here, we summarize the procedure and refer the reader to Cordeau (2006) for more details. The

instances are generated on a 10x10 grid. Each request has a pickup and a drop-off, both located at random positions on the map. The depot is located at the center. Half of the requests are outbound requests (i.e., from home to an outside location), and the other half are inbound requests (i.e., from an outside location back home). The time window is set at the outside location first, then the time window at home is computed using the time windows tightening procedure described in Cordeau (2006). The requests are served by a fleet of homogeneous vehicles.

In order to have good solutions for training the classifiers, we need to solve the instances close to optimality. However, generating near-optimal solutions using the LNS raises two difficulties. First, the optimality of a solution cannot be proven by the LNS. Second, the computation times required to reach optimal or near-optimal solutions are unknown. To tackle these difficulties, we can take advantage of years of experimentation on benchmark instances by using the best-known solutions (BKS) as references. As our instances are generated using the procedure in Cordeau (2006), the computation times required to reach near-optimal solutions will be similar for the benchmark instances and our generated instances. For example, if our method takes around 5 minutes to reach the best-known solutions for benchmark instances of size 50, then it will likely take around the same duration to reach good solutions for our generated instances of the same size. Thus, we solve each instance of benchmark A from Cordeau (2006) and save the convergence. An instance is named after its number of vehicles and users. For example, instance a2-20 has 2 vehicles and 20 users. For each instance, 5 runs are performed with a time limit of 8 hours. Table 4.2 shows the average and maximum computation time required to reach solutions with gaps lower than 1% ($gap = \frac{score - BKS}{BKS}$). When no value is provided, it means that not every run was able to reach a gap smaller than 1%.

| instance | average time to 1% | max time to 1% |
|----------|--------------------|----------------|
| a2-16 | 0 | 0 |
| a4-16 | 0 | 0 |
| a3-18 | 0 | 0 |
| a2-20 | 0 | 0 |
| a2-24 | 0 | 0 |
| a3-24 | 0 | 0 |
| a4-24 | 0 | 0 |
| a3-30 | 0 | 0 |
| a4-32 | 0 | 0 |
| a3-36 | 0 | 0 |
| a4-40 | 0 | 2 |
| a5-40 | 0 | 2 |
| a4-48 | 80 | 165 |
| a6-48 | 195 | 335 |
| a5-50 | 31 | 58 |
| a7-56 | 1250 | 2702 |
| a5-60 | 1034 | 3763 |
| a6-60 | 11078 | 18326 |
| a8-64 | - | - |
| a7-70 | - | - |
| a6-72 | - | - |
| a8-80 | - | - |
| a7-84 | - | - |
| a8-96 | - | - |

Table 4.2: computation time (in s) to reach a 1% gap

First, the LNS is not able to find near-optimal solutions for instances with 64 users and more. Second, the computation time required to reach good solutions for instances with 60 users is variable and can be long. Indeed, despite having the same number of users, the computation time is more than 10 times bigger for instance a6-60 than for instance a5-60 (11078 s vs. 1034 s on average). For these reasons, we

exclude instances with 60 users and more. Third, while the LNS is able to quickly find near-optimal solutions for small instances, we also exclude them from our study. Indeed, as they have a low number of requests, it is more likely that a small variation between two small instances results in a large difference in their solutions' structure. It can be an issue for a later generalization to larger instances. Having more requests limits this problem. Thus, we also exclude instances with 36 users or less. Finally, the computation time is approximately 8 times bigger for instance a7-56 than for instance a6-48 which has the second worst computation time (2702 s vs. 335 s at worst). Thus, we also exclude instances with 56 users due to their greater computation time relative to the other instances and with respect to our computational resources. To conclude, we generate instances between 40 and 50 users. We set the computation time to 600 seconds in order to have a reasonable margin with respect to the worst computation time found while solving the benchmark from the literature (i.e., 335 s for instance a6-48). This setting aims to balance optimality and generalization capabilities.

4.3.4.3 Solution generation

We define 3 levels of optimality: 0% gap (good solution), 2% gap (bad solution), and 4% gap (very bad solution). The 0% gap corresponds to the best solutions found after 600 seconds of computation. The optimality cannot be guaranteed with our solver as it is not an exact method. However, the good quality of our solutions is supported by the preliminary work on the benchmark instances from Cordeau (2006) (see Section 4.3.4.2). It is impossible to obtain a precise gap for all solutions. Thus, 2% and 4% gap solutions have in practice a gap in a $[1.5, 2.5]$ interval and a $[3.5, 4.5]$ interval.

4.3.4.4 Metric extraction

The knowledge extraction process starts by extracting metrics according to the level of the elements studied (i.e., solutions or arcs).

Instance metrics. First, we need to characterize the instances. The metrics used to describe an instance must give a good sense of its structure. These metrics can include the number of users, the geographical distribution of the requests, the location of the depot, etc. Our generated instances are basic DARP instances. Thus, they share many characteristics such as the depot location (i.e., at the center of the map), vehicles' characteristics (e.g., same capacity), the maximal ride time of users, and so on. As a result, we exclude such metrics from our database as they would be identical for every data point. However, these metrics are relevant in the case of instances with more features (e.g., heterogeneous fleet, multiple depots, clustered demand, ...). The list of extracted metrics is presented in Table 4.3.

| characteristic | metric |
|-------------------------------|---|
| demand | number of users |
| resources | number of vehicles |
| availability of the resources | capacity utilization |
| feasibility | percentage of feasible arcs |
| density | distance between nodes |
| temporal tightness | users' time windows |
| remoteness | distance between the requests and the depot |

Table 4.3: list of instance metrics

The demand and resources are measured using the number of users, the number of vehicles, and the capacity utilization. The capacity utilization corresponds to the ratio between the total load (i.e., the number of users in our case) and the total capacity of the fleet of vehicles. We characterize the difficulty to solve an instance by its global feasibility. It corresponds to the percentage of feasible arcs. Here, we define a feasible arc as an arc that passes all the preliminary feasibility tests related to the temporal constraints (Braekers et al., 2014). The density is measured using the pair-wise distance between nodes. This metric is split into multiples metrics (i.e., minimum, maximum, median, first quartile, third

quartile, average, standard deviation) to have a more detailed picture of the distribution of the density. In our instances, the locations of the pickups and depots are chosen randomly according to a uniform distribution. As a result, the more users, the more these metrics are similar across instances. However, we decided to include these metrics because the variability is probably not negligible for our medium-sized instances. The temporal tightness is measured using the duration of the users' time windows. Like the density, the temporal tightness distribution is studied. Finally, the remoteness of the users is measured using the distance between the requests and the depot. For each request, the distance of the direct itinerary "depot - pickup - drop-off - depot" is computed. Once again, the remoteness distribution is studied.

Solution metrics. At solution level, the next step is to characterize the solutions. The set of metrics used is presented in Table 4.4.

The first metric is the value of the objective function (i.e., the total distance). Then, we use a set of metrics that characterize the routes. We measure the number of routes, the routes' depth, the routes' angle, the number of requests per route, and the routes' average load. Except for the first metric which is straightforward, we analyze each metric using its distribution (i.e., minimum, maximum, median, first quartile, third quartile, average, standard deviation). The route's depth and angle are represented in Figure 4.3. Given a route r , its load is given by the following formula $\sum_{a \in r} L_a d_a / \sum_{a \in r} d_a$, with L_a being the cumulative load when traveling arc a , and d_a being the distance of the arc a . It corresponds to the weighted average cumulative load relative to the arcs' distance.

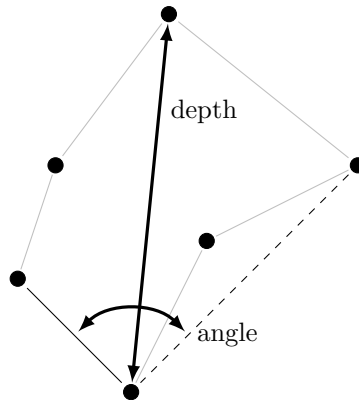


Figure 4.3: route metrics

The next set of metrics gives a global overview of the arcs in the solution. We measure the distribution of the arcs' length relative to the overall solution, to the users, and to the depot. The arcs' length relative to a user corresponds to the length of the arcs incident to its pickup and drop-off node (i.e., incoming and outgoing pickup and drop-off arcs). The arcs' length relative to the depot corresponds to the length of the first and last arc of a route. We measured another metric related to the arcs: the connection rank. We define this metric as follows: the connection rank between a node and its first (respectively second, third, ..., n-th) closest reachable neighbor is a 1st-rank (respectively 2nd-rank, 3rd-rank, ..., n-th-rank) connection. A neighbor is reachable if the arc passes the preliminary tests related to the time windows, the service time at origin, and the arc's duration (Braekers et al., 2014). It is important to note that the connection ranks are not symmetrical. For example, the arc from node A to node B can have a 1st-rank connection, but the opposite arc from B to A can have a higher-rank connection or even be infeasible. Thus, the connection rank of a node is the average between its incoming and outgoing connection rank. Once again, we measure the distribution of the connection ranks to characterize solutions.

Finally, we measure the distribution of the waiting time of users. This metric is related to the quality of service. We suppose that this metric may not be as relevant as others because the routes' duration is not considered in our objective function. However, this metric is probably important when routes' duration impacts the objective function.

| solution characteristic | metric |
|-------------------------|---|
| objective | total distance |
| routes | number depth angle number of requests per route average load |
| arcs | distance incident distance for users incident distance for the depot connection rank |
| users | waiting time |

Table 4.4: list of solution metrics

Arc metrics. At arc level, solution metrics are replaced by arc metrics. For the sake of clarity, we devised two sets of metrics. The first one corresponds to the set of simple metrics and the second one corresponds to the set of composed metrics. A composed metric is a ratio between two simple metrics. For example, if the number of LNS iterations and the number of times the arc has been inserted are two simple metrics, then the ratio between the two is a composed metric. The set of simple metrics is presented in Table 4.5 and the set of composed metrics is presented in Table A.1 in Appendix A.1.

| arc characteristic | metric |
|---------------------|---|
| search | total number of feasibility tests |
| static information | cost, FTS, minimum waiting time |
| dynamic information | number of feasibility tests number of apparitions/insertion/removals |

Table 4.5: list of arc metrics

The first metric gives information on the overall search process using the total number of feasibility tests performed during the search. It corresponds to four metrics: the total number of preliminary tests computed, the total number of preliminary tests succeeded, the total number of preliminary tests computed, and the total number of preliminary tests succeeded.

The next set of metrics uses the arcs static information. These metrics stay unchanged during the resolution and can be computed during a preprocessing phase. The first metric of this set is the cost. In our context, it corresponds to its distance. In other DARP variants, the cost may include the duration-related cost and the fixed cost of the vehicles. The second metric of this set is the arc's Forward Time Slack (FTS). The FTS has been introduced in Savelsbergh (1992). It is the maximum value by which the start of service at a node can be delayed without making the route infeasible. We adapt this definition at arc level. In our case, the FTS is the maximum value by which the start of service at the origin node can be delayed without making the arc to a destination node infeasible. In other words, the arc's FTS is the maximum delay at the origin that still allows to start the service at destination within its time window. The last metric of this set of static metrics is the minimum wait of an arc. It equals the difference between the latest arrival time and the earliest service time at a destination node. The latest arrival time is equal to the sum of the time window's end at the departure node, the service time at the departure node, and the duration of the arc. The earliest service time at the destination node corresponds to the beginning of its time window.

The next set of metrics includes metrics related to dynamic information. In other words, their value changes during the resolution. The arc's number of apparitions corresponds to the number of times the arc has appeared in a newly built solution of the LNS (one new solution per iteration of the LNS). The number of insertions and removals corresponds to the number of times the arc has been inserted and removed by the LNS operators.

There is a fundamental difference between the solution and the arc metrics: solution metrics are computed at the end of the resolution and the arc metrics are computed during the early stages of the resolution. This is due to the dynamic nature of some arc metrics. Let us take the number of successful scheduling tests as an example. In the first scenario, the arc metrics are saved at the end of the computation. Here, the extracted knowledge "an arc is good if it has more than 1000 successful scheduling tests at the end of the resolution" does not indicate if an arc is good or bad during the resolution. Thus, it is difficult to use this knowledge to guide the resolution. In the second scenario, the arc metrics are saved at an early stage of the resolution. Here, the extracted knowledge "an arc is good if it has more than 50 successful scheduling tests after 1000 LNS iterations" can be used just after the first 1000 iterations. Thus, it is possible to build operators that target arcs with at least 50 successful scheduling tests after 1000 iterations. As a consequence, arc metrics are computed during the early stages of the resolution. We save the arc metrics at 1000, 2000, 3000, 4000, and 5000 LNS iterations.

4.3.5 Characteristic extraction

4.3.5.1 Is it possible to differentiate good and bad solutions/arcs using the extracted metrics?

As mentioned in the presentation of the solution method in Section 4.3.2, the first question we try to answer is: "is it possible to differentiate good and bad solutions/arcs using the extracted metrics?".

In order to extract useful knowledge, we avoid advanced classifiers such as deep neural networks. Indeed, our goal is not to have perfect accuracy on the predictions but rather to obtain guidelines for solving the DARP. As a result, we used less accurate but simpler classifiers. We used the following classifiers from the scikit-learn library: decision trees, random forests, linear SVMs, and SVMs with a Radial Basis Function kernel (RBF-SVM). For decision trees and random forests, the number of splits is set to the number of metrics. For linear SVMs and RBF-SVMs, the metrics are first normalized using the standard scaler from the scikit-learn library.

At solution level, two datasets are created. The first one contains the good and the bad solutions (i.e., 0% and 2% gap solutions), the second one contains the good and the very bad solutions (i.e., 0% and 4% gap solutions). At arc level, there are also two datasets. However, these two datasets exist for every iteration level at which the arc metrics are computed. In other words, there are two datasets corresponding to the arcs at 1000 iterations, two others at 2000 iterations, two others at 3000 iterations, etc. It is important to note that an arc can be present in a good and in a bad solution at the same time. Thus, this arc is not exclusive to good solutions. As a result, we only keep the arcs that appear in one gap category: arcs labeled as good arcs (respectively bad and very bad) only appear in 0% gap solutions (respectively 2% and 4% gap solutions). Then, the dataset is balanced by randomly removing data points from the excess class. The average accuracy of the classifiers is measured over 10 runs. At each run, every classifier is trained on a training set (75% of the data points) and its accuracy is measured on a test set (25% of the data points).

Tables 4.6 and 4.7 present the results from the prediction. In Table 4.7, sub-columns 1000, 2000, 3000, 4000, and 5000 correspond to the iteration at which the arc metrics were saved. First, every accuracy measured is greater than 50% for all classifiers, across all datasets, and at both solution and arc levels. On average, 67% of the solutions' and 66% arcs' labels are correctly predicted. Thus, it is possible to differentiate good and bad solutions/arcs based on the extracted metrics. At solution level, the accuracy goes up to 77% when using a random forest classifier on the 4% gap dataset. At arc level, the accuracy goes up to 73% when using a random forest classifier on the 4% gap 5000 iterations dataset. So at best, it is possible to correctly predict the quality of approximately 3 out of 4 solutions/arcs. Second, the accuracy score is almost always better on the 4% gap datasets. Thus, it is more difficult for classifiers to identify the difference between 0% gap and 2% gap solutions/arcs compared to 0% gap and

| | 2% gap | 4% gap | average |
|---------------|--------|--------|---------|
| decision tree | 58 | 63 | 61 |
| random forest | 69 | 77 | 73 |
| linear SVM | 62 | 71 | 66 |
| RBF-SVM | 64 | 75 | 70 |
| average | 63 | 71 | 67 |

Table 4.6: prediction accuracy (in %) when using all the metrics at solution level

| | 2% gap | | | | | 4% gap | | | | | average |
|---------------|--------|------|------|------|------|--------|------|------|------|------|---------|
| | 1000 | 2000 | 3000 | 4000 | 5000 | 1000 | 2000 | 3000 | 4000 | 5000 | |
| decision tree | 64 | 61 | 65 | 61 | 62 | 63 | 62 | 66 | 64 | 68 | 64 |
| random forest | 65 | 63 | 69 | 65 | 65 | 69 | 67 | 71 | 71 | 73 | 68 |
| linear SVM | 63 | 64 | 69 | 65 | 63 | 67 | 64 | 68 | 69 | 71 | 66 |
| RBF-SVM | 66 | 62 | 71 | 66 | 67 | 66 | 67 | 67 | 70 | 71 | 67 |
| average | 65 | 62 | 69 | 64 | 64 | 66 | 65 | 68 | 69 | 71 | 66 |

Table 4.7: prediction accuracy (in %) when using all the metrics at arc level

4% gap solutions/arcs. This confirms the intuition that 2% gap solutions/arcs have characteristics more similar to 0% gap solutions/arcs than 4% gap solutions/arcs. Third, the accuracy is better when using more advanced classifiers. For instance, there is a 12% ($= 73 - 61$) accuracy difference at solution level between the predictions of decision trees and random forests. At arc level, the difference only amounts to 4% ($= 68 - 64$). This result was predictable as the prediction capabilities of advanced classifiers are often better than simpler ones. Nevertheless, we judge that the simplest classifier (i.e., decision tree) showed good enough results for it to be used to extract good metrics. Finally, the accuracy at arc level is relatively stable across all iteration levels relative to a classifier. For instance, the 9% ($= 71 - 62$) accuracy difference measured for the RBF-SVM classifier on the 2% gap dataset is the biggest difference. It means that more iterations do not result in more accuracy.

4.3.5.2 What are the characteristics of good solutions/arcs?

At this point, we know that it is possible to use the extracted metrics to predict if a solution or an arc is good. The next step is to find which are the metrics or set of metrics that are responsible for such predictions.

Tuning the decision tree classifier. Despite having modest prediction performance, decision trees are the ideal classifiers for this task as they are easy to interpret and their parameters are easy to set. As a reminder, our goal is not focused on prediction accuracy but on finding guidelines to identify good solutions/arcs. On the one hand, it is difficult to interpret and derive simple guidelines from a deep decision tree. On the other hand, a shallow decision tree may have a low accuracy score, thus being unreliable for implementing guidelines. As a consequence, we need to find the ideal size for our decision trees. We test decision trees of different sizes by setting the number of leaf nodes between 2 (smallest tree) and 20. The results are presented in Tables 4.8 and 4.9.

First, the results are fairly stable across the number of leaves at both solution and arc levels. Indeed, the worst score is 62.2% (2 leaves) and the best is 65.0% (5 leaves) at solution level. At arc level, the worst score is 65.5% (17 leaves) and the best is 67.9% (4 leaves). Second, the best accuracies are found for a relatively low number of leaves (between 3 and 6 leaves). This shows that the decision trees are overfitting when the number of leaves is too large. As a confirmation, we can observe that the previous results obtained with bigger decision trees (Tables 4.6 and 4.7) are worse than the results obtained here.

| nb leaf nodes | 2% gap | 4% gap | average |
|---------------|--------|--------|---------|
| 2 | 60.5 | 64.0 | 62.2 |
| 3 | 62.2 | 65.2 | 63.7 |
| 4 | 62.6 | 66.6 | 64.6 |
| 5 | 62.9 | 67.0 | 65.0 |
| 6 | 61.8 | 67.1 | 64.5 |
| 7 | 61.3 | 66.9 | 64.1 |
| 8 | 61.3 | 66.9 | 64.1 |
| 9 | 60.9 | 66.2 | 63.5 |
| 10 | 60.7 | 66.5 | 63.6 |
| 11 | 60.6 | 66.7 | 63.6 |
| 12 | 60.7 | 66.5 | 63.6 |
| 13 | 60.7 | 66.4 | 63.5 |
| 14 | 60.6 | 66.7 | 63.7 |
| 15 | 60.4 | 67.3 | 63.8 |
| 16 | 60.5 | 67.2 | 63.8 |
| 17 | 60.7 | 67.6 | 64.1 |
| 18 | 60.2 | 67.5 | 63.8 |
| 19 | 60.1 | 67.5 | 63.8 |
| 20 | 60.1 | 67.3 | 63.7 |

Table 4.8: prediction accuracy (in %) of relative to the number of leaf nodes in the decision tree at solution level

| nb leaf nodes | 2% gap | | | | | 4% gap | | | | | average |
|---------------|--------|------|------|------|------|--------|------|------|------|------|---------|
| | 1000 | 2000 | 3000 | 4000 | 5000 | 1000 | 2000 | 3000 | 4000 | 5000 | |
| 2 | 64.1 | 62.0 | 61.8 | 64.9 | 66.2 | 67.2 | 67.9 | 72.4 | 74.5 | 71.3 | 67.2 |
| 3 | 63.4 | 61.4 | 62.4 | 64.9 | 67.1 | 68.9 | 69.7 | 72.0 | 73.9 | 70.8 | 67.5 |
| 4 | 64.1 | 64.6 | 64.7 | 64.3 | 68.1 | 71.4 | 67.9 | 70.6 | 72.6 | 70.3 | 67.9 |
| 5 | 64.6 | 65.6 | 65.5 | 63.6 | 66.6 | 70.8 | 67.4 | 70.7 | 71.7 | 70.4 | 67.7 |
| 6 | 64.1 | 65.0 | 64.7 | 62.7 | 68.2 | 70.2 | 66.6 | 69.5 | 71.6 | 70.2 | 67.3 |
| 7 | 65.2 | 64.3 | 64.2 | 60.4 | 67.0 | 70.8 | 66.8 | 67.2 | 71.4 | 70.9 | 66.8 |
| 8 | 65.0 | 64.5 | 64.4 | 61.6 | 66.3 | 70.6 | 66.3 | 67.2 | 72.0 | 69.7 | 66.8 |
| 9 | 65.1 | 64.7 | 65.3 | 63.2 | 67.8 | 69.0 | 66.2 | 66.3 | 71.7 | 70.6 | 67.0 |
| 10 | 64.0 | 64.6 | 65.1 | 62.6 | 66.6 | 69.2 | 66.5 | 66.5 | 71.4 | 70.7 | 66.7 |
| 11 | 64.1 | 64.2 | 64.9 | 62.6 | 66.9 | 69.5 | 67.1 | 66.3 | 71.2 | 70.5 | 66.7 |
| 12 | 64.5 | 64.3 | 64.0 | 63.5 | 65.6 | 69.4 | 67.1 | 66.7 | 71.2 | 70.8 | 66.7 |
| 13 | 64.0 | 64.0 | 64.3 | 62.9 | 65.4 | 68.2 | 66.4 | 66.8 | 70.5 | 69.6 | 66.2 |
| 14 | 63.8 | 64.8 | 63.5 | 63.2 | 65.6 | 68.8 | 66.6 | 66.3 | 70.4 | 70.3 | 66.3 |
| 15 | 64.2 | 64.5 | 62.9 | 63.5 | 64.8 | 69.5 | 66.7 | 66.0 | 69.8 | 68.9 | 66.1 |
| 16 | 64.1 | 64.1 | 63.1 | 63.7 | 64.9 | 69.0 | 66.6 | 65.8 | 69.7 | 69.1 | 66.0 |
| 17 | 63.3 | 64.7 | 62.4 | 63.5 | 64.6 | 66.8 | 66.2 | 65.5 | 69.3 | 68.7 | 65.5 |
| 18 | 64.7 | 64.5 | 62.7 | 62.9 | 64.0 | 69.5 | 66.1 | 65.8 | 69.0 | 68.9 | 65.8 |
| 19 | 64.7 | 63.9 | 61.7 | 63.2 | 64.8 | 68.9 | 65.9 | 65.5 | 69.5 | 69.6 | 65.8 |
| 20 | 64.7 | 63.6 | 63.0 | 62.7 | 64.2 | 67.8 | 65.3 | 64.9 | 69.7 | 70.6 | 65.7 |
| average | 64.3 | 64.2 | 63.7 | 63.2 | 66.0 | 69.3 | 66.8 | 67.5 | 71.1 | 70.1 | |

Table 4.9: prediction accuracy (in %) relative to the number of leaf nodes in the decision tree at arc level

Indeed, every decision tree with less than 20 leaf nodes has an accuracy greater than 61% at solution level, and greater than 64% at arc level. Another observation is that the accuracy of the simplest decision trees with 2 leaves is generally lower. Nevertheless, they still show good results compared to bigger decision trees (e.g., 60.5% accuracy for 2 leaves vs. 62.9% accuracy for 5 leaves at solution level for 2% gap dataset). Third, at arc level, the accuracy score is stable across the iteration levels. For example, for 4 leaves on the 2% gap dataset, the prediction accuracy is between 64% and 69% for any iteration level. Thus, after 1000 iterations, it is already possible to predict if an arc is good. These results suggest that it could be possible to guide the resolution after only 1000 iterations, which is a fairly early stage in an LNS resolution. However, the metrics used for the prediction at 1000 iterations might not be the same used at 5000 iterations. The same is true between the 2% gap and 4% gap datasets.

To summarize, the accuracy of the decision tree classifier is good and stable relative to the number of leaves. The best results are obtained for decision trees with 3 to 6 leaves. Nevertheless, decision trees with 2 leaves also achieve good results despite being the simplest decision trees. As a consequence, we decided to explore the decision trees with 2 to 6 leaves. At arc level, results are stable across iteration levels with good predictions as early as 1000 iterations. However, the metrics used in the 1000 iterations and 5000 iterations decision trees might not be the same. As a consequence, we explore the decision trees at 1000 and 5000 iterations.

Decision tree analysis at solution level. In this section, 2% and 4% gap datasets are studied using decision trees with 2 to 6 leaves. Among the many decision trees generated, the two decision trees shown in Figures 4.4 and 4.5 illustrate the following observations. First, the metrics used by the decision trees on the 2% and the 4% gap datasets are very similar. For both datasets, decisions are often made using the standard deviation of the number of requests per route, the median or average arc distance, the user incident arcs distance metrics, and the average route load. Second, the main characteristic to differentiate good and bad solutions is the standard deviation of the number of requests per route. Indeed, for almost every decision tree, the first split is made using this metric. Moreover, the splitting rule is always the same: “if the standard deviation of the number of requests per route is smaller than (approximately) 3.5, then the solution is good”. This rule indicates that the requests are better distributed between routes in good solutions than in bad solutions. As a result, our first guideline is to aim for a balanced distribution of requests between routes.

Third, a common splitting rule indicates that shorter arcs are more likely to correspond to good solutions. This corresponds to metrics like the average arc distance or the user incident arcs distance. However, this guideline is obvious as our objective function is to minimize the distance. Moreover, metrics related to the arcs’ distances might overshadow other metrics. Thus, we remove these distance-related metrics (i.e., arc distance and user incident distance) and build new decision trees. Figure 4.6 shows a decision tree that does not use any distance-related metric. Fourth, a similar accuracy is obtained without the distance-related metrics (i.e., 63% on average). Now, it is the metrics related to the connection ranks that prevail. While still being related to the arc distances, these metrics use more DARP-specific knowledge as they include only the arcs that are feasible according to the preliminary tests (Braekers et al., 2014). As suggested in Figure 4.6, our second guideline should aim to reduce the connection ranks.

Decision tree analysis at arc level. In this section, 2% and 4% gap datasets at 1000 and 5000 iterations are studied using decision trees with 2 to 6 leaves. Figures 4.7 and 4.8 illustrate the following observation. First, the metrics used at 1000 iterations are not the same as the metrics used at 5000 iterations. On the one hand, the decision trees at 1000 iterations rely more on static metrics. On the other hand, decision trees at 5000 iterations rely more on dynamic metrics. On the opposite, the metrics used for the 2% and 4% gap dataset are similar with respect to the iteration level and the number of leaves. Second, the most important criterion at 1000 iterations is the FTS. As shown in Figure 4.7, a lower FTS tends to correspond to good arcs. This suggests that arcs with a low FTS, while being more temporally constrained, are more exclusive to good solutions. We suppose that these arcs are more difficult to insert in solutions, but at the same time, they are key in reaching the regions of the research space that contain the best solutions. As a result, the third guideline is to target arcs with a low FTS.

Third, running the decision trees multiple times showed that some metrics lead to similar results. As a consequence, we decided to remove some metrics to simplify the interpretation of the decision trees. We

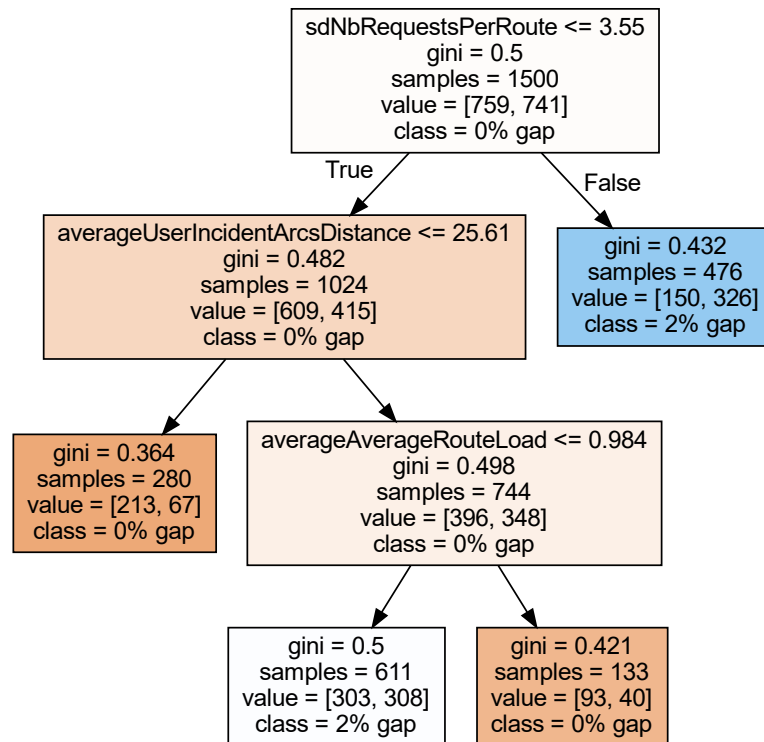


Figure 4.4: decision tree at solution level on the 2% gap dataset with 5 leaves

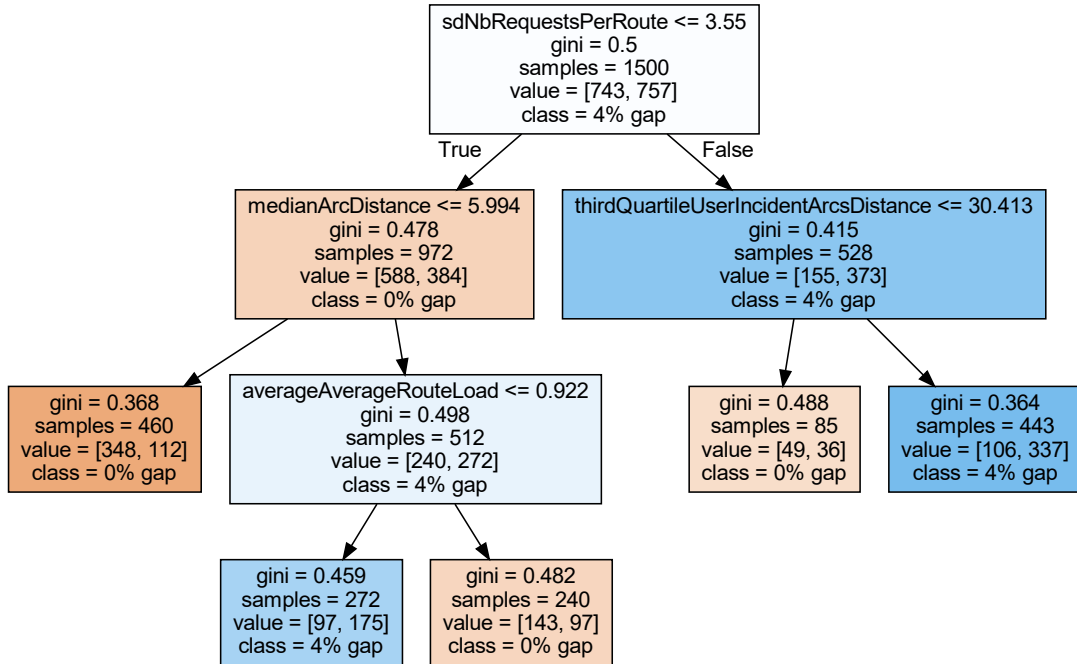


Figure 4.5: decision tree at solution level on the 4% gap dataset with 5 leaves

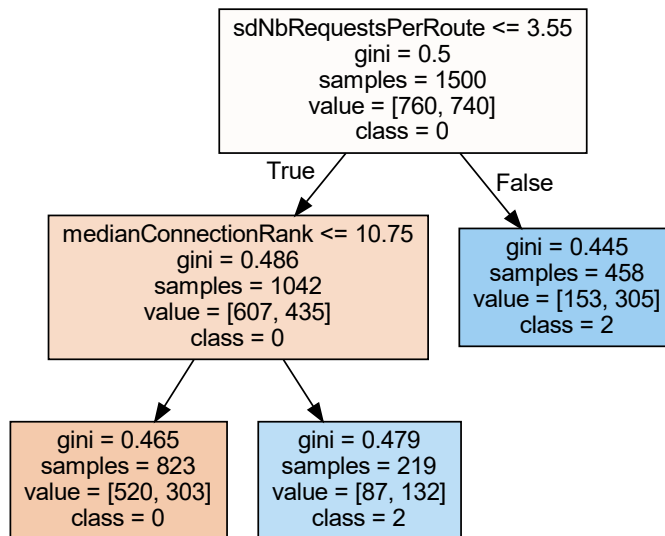


Figure 4.6: decision tree at solution level on the 2% gap dataset with 3 leaves without the distance-related metrics

remove metrics related to the number of preliminary tests, insertion, and removal. Indeed, the number of scheduling tests gives results similar to the number of preliminary tests (68.2% accuracy without the preliminary tests metrics vs. 67.0% accuracy without the scheduling tests metrics for 10 runs). Similarly, the number of apparitions gives slightly better results than the number of insertions or removals (68.5% accuracy with only the apparition metric vs. 67.5% accuracy with only the insertion metric vs. 67.3% accuracy with only the removal metric for 10 runs). Removing these metrics has a positive impact on the prediction accuracy with a 67.4% accuracy compared to a 66.7% accuracy with all the metrics (average for 10 runs). Thus, having fewer metrics reduces overfitting. Fourth, given the new decision trees, the second most frequent criterion is the ratio between the number of apparitions and the number of scheduling tests succeeded. Using only this criterion, the decision tree illustrated in Figure 4.9 obtains a prediction accuracy of 64.86%. It shows that a higher ratio tends to correspond to good arcs. This means that good arcs are already used in far-optimal solutions during the early stage of the resolution. However, it is difficult to identify a specific value. Indeed, depending on the decision tree parameters and the dataset, the value for the splitting criteria varies between 0.1 and 0.4. Nevertheless, we can set a new guideline that promotes arcs with a high apparition to successful scheduling ratio.

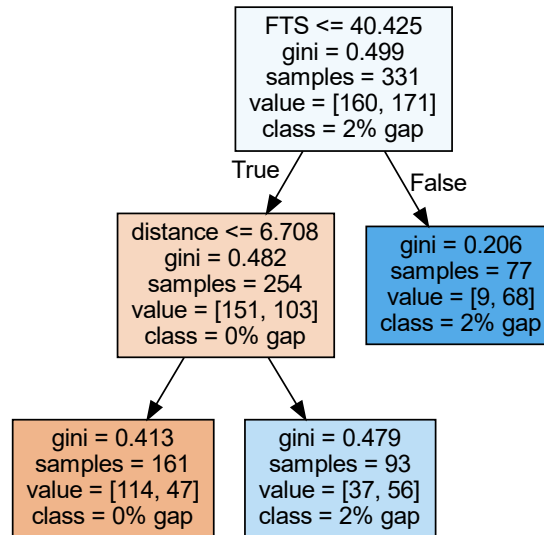


Figure 4.7: decision tree at arc level on the 2% gap 1000 iterations dataset with 3 leaves

4.3.6 Building knowledge-guided operators

Thanks to the previous section, 4 guidelines have been established: we should aim for a balanced distribution of requests between routes, low connection ranks, low FTSs, and high number of apparitions to number of successful scheduling tests ratios. As a result, we developed 4 new operators.

4.3.6.1 4 new operators.

The first operator aims to balance the distributions of requests between routes. First, the operator selects the route with the fewer users. Then, for each uninserted user, it computes the insertion cost for every position in the route and saves the cheapest insertion. Then, it inserts the user with the best insertion cost. This process is repeated until there is no more user to insert or no more feasible insertion.

The second operator aims to insert arcs with low FTS. For each uninserted user, it computes if the insertion is possible for every position in every route. If the insertion is possible, it computes the average

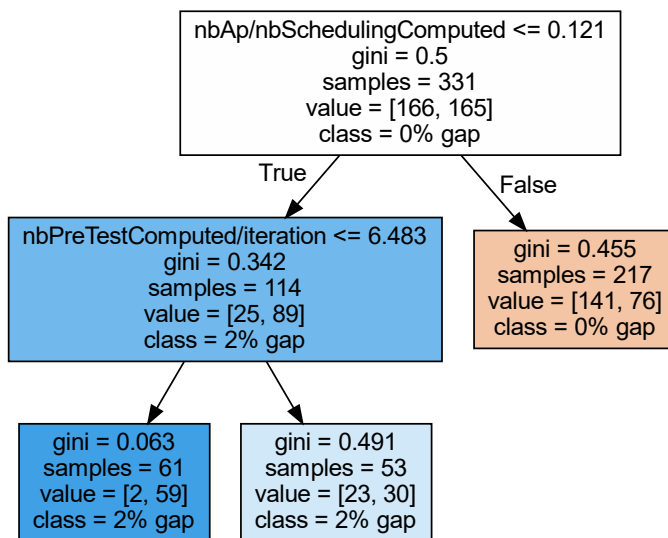


Figure 4.8: decision tree at arc level on the 2% gap 5000 iterations dataset with 3 leaves

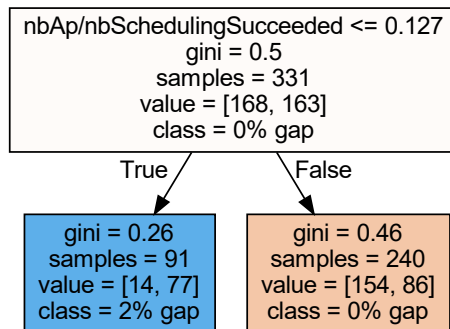


Figure 4.9: decision tree at arc level on the 2% gap 5000 iterations dataset with only one split on the ratio between the number of apparitions and the number of scheduling tests succeeded

FTS of the pickup and drop-off incident arcs. Then, the operator inserts the user with the lowest FTS insertion. Once again, this process is repeated until there is no more user to insert or no more feasible insertion.

The third operator aims to reduce the connection ranks in the solution. Here, the connection ranks are computed during the preprocessing step using the definition presented in Section 4.3.4.4. The ranks are stored in a matrix of connection ranks. The insertion process of this operator is the same as the previous one for low FTS.

The last operator aims to insert arcs with a good ratio between their numbers of apparitions and their numbers of scheduling tests succeeded. As shown during the knowledge extraction process, this metric cannot be used directly at the start of the resolution but after 5000 iterations. Thus, this operator is not used during the first 10% iterations of the resolution. During the first 5000 iterations, the number of apparitions and the number of scheduling tests are saved. At 5000 iterations, the target metric value is computed for every arc. Then, this metric is used by the fourth operator. It is important to note that the metric's value stays constant during the rest of the resolution. Indeed, when its value is computed at 5000 iterations, we know that it is a good metric for predicting if an arc is good. However, we don't know if this is the case when computed after 5000 iterations. Once again, the insertion process is the same as for the two previous operators.

4.3.6.2 Testing the new operators

The performance of the new operators is evaluated on the benchmark from Cordeau (2006) as this benchmark was used for the learning phase of our study. Table 4.10 presents the average gaps obtained without the new operators, for each operator added individually, and with all the new operators. The gaps correspond to the gap with the best-known solutions for the literature (Ho et al., 2018). Columns “no op” and “all” respectively correspond to the LNS without any new operator and with all new operators. In between, each column corresponds to the operator associated with the metric in the column label. Bold values correspond to the best gap for each instance.

First, the best gap (i.e., 0.40%) is obtained with the basic LNS without the knowledge-guided operators. Nevertheless, the results with only the “FTS” operator and “connection rank” operator are very close (0.41%). However, including all the new operators at the same time gives the worst results (0.60%). These observations show that the knowledge-guided operators do not improve the performance of the LNS.

Second, the knowledge extracted from instances with 40 to 50 users can be generalized to larger instances. Indeed, on instances with 50 users and more, the best gap is obtained 4 times with the basic LNS, 3 times with the “FTS” LNS, and 3 times with the “connection rank” LNS. When looking at the instances with 50 users and more, the gaps are relatively similar between the “no op”, “FTS”, and “connection rank” columns. The gap difference is at most 0.5% ($= 2.37 - 1.87$) for instance a8-80 between “no op” LNS and “connection rank” LNS. Thus, the new operator performs similarly on medium and large instances. Such small gap differences are probably caused by the variability between the different runs. For example, one lucky run can result in a very good solution, thus giving the edge to one LNS configuration. Thus, we can say that the performance between the “no op”, “FTS”, and “connection rank” LNS are very similar.

Finally, we also measured the computation duration required to reach a 1% gap. Over the 120 runs per LNS configuration, this gap was not reached 21 times for the “no op” LNS, 20 times for the “FTS” LNS, and 22 times for the “connection rank” LNS. In addition, when excluding the experiments where a 1% gap was not reached, the average computation duration required is 4.8 seconds for the “no op” LNS, 5.3 seconds for the “FTS” LNS, and 4 seconds for the “connection rank” LNS. Once again, this particular implementation of the knowledge-guided operators do not significantly improve the performance of the LNS but at the same time, they do not decrease it.

4.3.7 Conclusion and perspectives

In this study, we developed four new knowledge-guided operators for solving the Dial-A-Ride Problem (DARP). These operators were generated using a 3 steps offline learning method. During the first step,

| instance | no op | balanced route | FTS | connection rank | nbAp/nbSched | all |
|----------|-------------|----------------|-------------|-----------------|--------------|-------------|
| a2-16 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| a4-16 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| a3-18 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| a2-20 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| a2-24 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| a3-24 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| a4-24 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| a3-30 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| a4-32 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| a3-36 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| a4-40 | 0.02 | 0.02 | 0.00 | 0.00 | 0.00 | 0.01 |
| a5-40 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| a4-48 | 0.21 | 0.07 | 0.09 | 0.06 | 0.26 | 0.38 |
| a6-48 | 0.32 | 0.33 | 0.32 | 0.36 | 0.65 | 0.42 |
| a5-50 | 0.11 | 0.07 | 0.04 | 0.09 | 0.11 | 0.15 |
| a7-56 | 0.09 | 0.35 | 0.13 | 0.59 | 0.43 | 0.93 |
| a5-60 | 0.43 | 0.53 | 0.55 | 0.40 | 0.55 | 0.53 |
| a6-60 | 0.64 | 0.72 | 0.63 | 0.54 | 0.61 | 0.81 |
| a8-64 | 1.24 | 1.27 | 1.16 | 1.18 | 1.52 | 1.71 |
| a7-70 | 0.95 | 0.72 | 0.94 | 0.86 | 1.09 | 1.34 |
| a6-72 | 0.74 | 1.03 | 0.99 | 0.84 | 0.93 | 1.44 |
| a8-80 | 2.37 | 2.53 | 2.22 | 1.87 | 2.59 | 2.79 |
| a7-84 | 0.75 | 1.04 | 1.10 | 1.04 | 1.39 | 1.65 |
| a8-96 | 1.66 | 2.00 | 1.66 | 1.91 | 2.03 | 2.14 |
| average | 0.40 | 0.44 | 0.41 | 0.41 | 0.51 | 0.60 |

Table 4.10: average gaps (in %) on the Cordeau 2006 A benchmark

1000 instances based on the benchmark introduced in Cordeau (2006) are created and the corresponding good and bad solutions are computed using a Large Neighborhood Search (LNS) metaheuristic. Metrics describing the characteristics of the solutions and arcs are generated. During the second step, several classifiers are trained to predict the quality of solutions or arcs based on their metrics. The classifiers were able to correctly predict the quality of solutions and arcs with an average accuracy of respectively 67% and 66% (up to respectively 77% and 71%). Then, the most prevalent metrics are converted into guidelines for solving the DARP. During the last step, new LNS operators are designed based on these guidelines. The new operators are tested on the benchmark from Cordeau (2006). We identified four guidelines in our study and derived four repair operators. The first operator balances the requests between routes, the second targets insertions involving arcs with low Forward Time Slack (FTS), the third aims for low connection rank insertion, and the last one targets arcs with a high number of apparitions to number of scheduling tests succeeded ratio. Unfortunately, neither the individual nor the collective performance of the new operators improved the resolution. Indeed, the basic LNS implementation obtained a better average gap (0.40%) than any LNS implementation with the new operators (0.41% at best) over the benchmark from Cordeau (2006).

Several factors can explain the results and require to be investigated in future work. First, the accuracy of the classifiers might be too low to be correctly converted into new operators. More elaborated metrics or guidelines might be required. Second, we decided to translate the extracted knowledge to new repair operators. However, this knowledge could have been used differently (e.g., penalized arcs' cost, ruin operators). Third, the developed operators are very specific as they target particular metrics instead of trying to improve the overall objective function. Thus, they are probably more useful during the exploration phases of the resolution. A better use of the knowledge-guided operator needs to be investigated. Finally, we studied if a solution or an arc was good. Other scopes could be investigated, such as whether a route or group of users is good.

4.4 Nested rollout policy adaptation applied to the dial-a-ride problem

Highlights:

- An online machine learning method called Nested Rollout Policy Adaptation (NRPA) is adapted to the Dial-A-Ride Problem (DARP)
- The NRPA is integrated in a greedy insertion framework. It orders the list of requests to be greedily inserted in sequence.
- The solution method is tested on classic instances from the literature and compared to a Large Neighborhood Search (LNS) metaheuristic.
- The experiment shows that the NRPA is able to learn. It shows results similar to the LNS on instances with 20 to 50 users. However, the gap increases for larger instances. In addition, the computation time is much higher for the NRPA.

4.4.1 Introduction

The Nested Rollout Policy Adaptation (NRPA) is a reinforcement learning method based on the Monte Carlo Tree Search (MCTS) method. The MCTS was introduced in 2006 in Kocsis and Szepesvári (2006) and Coulom (2007). One of its major advantages is that it requires little to no domain-specific knowledge (Browne et al., 2012). Thus, it can be used on a wide variety of problems. In particular, the MCTS has shown outstanding results on games. For instance, it has greatly improved the performance of computers on the Go game, from amateur to professional level (Gelly et al., 2012). MCTS has also been used on multiple problems such as scheduling problems (Asta et al., 2016; Waledzik et al., 2014), routing problems (Juan et al., 2013; Mańdziuk and Świechowski, 2017), and mixed-integer programming (Sabharwal et al., 2012).

To find the optimal solution, the MCTS explores the research tree by taking random samples in the research space (Browne et al., 2012). The MCTS can be decomposed into 4 steps: selection, expansion, simulation, and backpropagation. Figure 4.10 illustrates these steps. First, starting from the root node, the algorithm successively selects nodes until a leaf node is reached. This selection is guided by a selection policy (e.g., upper confidence bound for trees). Second, the algorithm expands the tree by adding an unexplored node. Third, the algorithm performs a playout (also called rollout). It iteratively adds random child nodes to the explored path until a final state is reached. Finally, the backpropagation updates the scores of the path from the expanded node to the root node based on the score of the simulation.

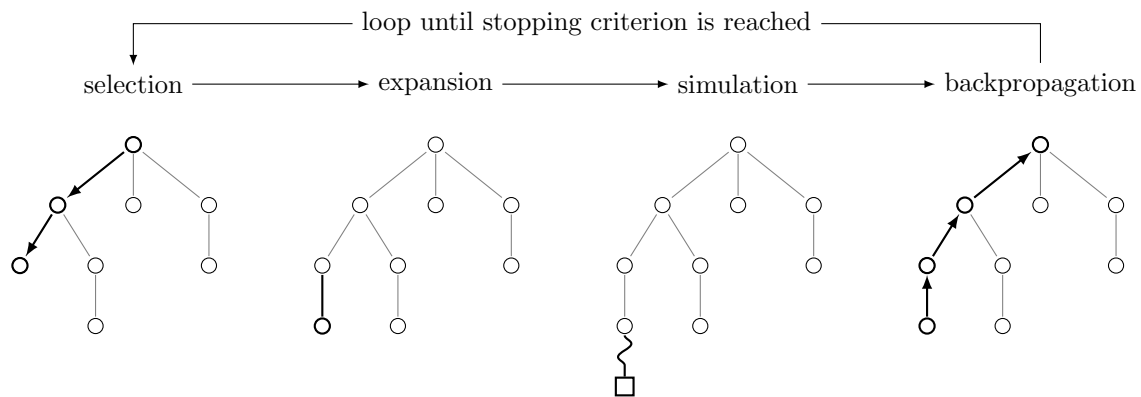


Figure 4.10: the four steps of the Monte Carlo Tree Search (MCTS)

The Nested Monte Carlo Tree Search (NMCS) is a variant of the MCTS introduced in Cazenave

(2009). Here, the algorithm calls itself recursively. As a result, higher-level playouts are biased by lower-level playouts. At level 1, the search is a standard playout. The NRPA is an even more advanced variant. Like the NMCS, the NRPA calls itself recursively. However, the performance of each move (i.e., action of going from one node to another in the research tree) is tracked using a policy. Then, the policy is used to build the solution.

In this study, we try to bridge research gaps RG1⁵ and RG3⁶ by extending the work done on the Vehicle Routing Problem (VRP) in Cazenave et al. (2021) to the Dial-A-Ride Problem (DARP). First, we present the solution method in Section 4.4.2. Then, we analyze the results in Section 4.4.3. Finally, we draw conclusions and perspectives in Section 4.4.4.

4.4.2 Solution method

4.4.2.1 Nested Rollout Policy Adaptation (NRPA)

The NRPA's structure is built around four elements: states, sequences, moves, and policies. A state describes the current status of a solution. A solution is built according to a sequence that describes the order of the nodes in the solution. A sequence is a list of moves, where each move corresponds to an instruction to build the solution (e.g., next node to add in the solution). Finally, the policy is a list of scores where each score corresponds to a move. The policy influences the probability for a move to be selected.

The NRPA can be decomposed into three functions: the playout function, the adapt function, and the main function. In order to facilitate the comprehension, $policy[m]$ designates the score corresponding to the move m stored in the policy table. The presentation of the functions is followed by Figure 4.11 which illustrates an NRPA with 2 levels and 2 iterations.

First, Algorithm 1 presents the playout function. This function is in charge of the simulation step. It builds a sequence from scratch (line 2) by adding one move at a time (lines 12-13) until a terminal state is reached (lines 4-6). The move selection (lines 7-11) is influenced by the policy. Moves with high scores have a higher probability of being selected. At each state, only the possible moves are considered (line 8). In Figure 4.11, playouts are drawn in black.

Algorithm 1 playout function

```

1: function PLAYOUT(state, policy)
2:   sequence  $\leftarrow$  [ ]
3:   while true do
4:     if state is terminal then
5:       return (score(state), sequence)
6:     end if
7:      $z \leftarrow 0$ 
8:     for move in possible moves of state do
9:        $z \leftarrow z + e^{policy[move]}$ 
10:    end for
11:    choose a move with probability  $\frac{e^{policy[move]}}{z}$ 
12:    state  $\leftarrow$  play(state, move)
13:    sequence  $\leftarrow$  sequence + move
14:  end while
15: end function

```

Then, Algorithm 2 presents the adapt function. This function updates the policy of the current level according to the best sequence found so far (i.e. drawn in red in Figure 4.11). For example, in Figure 4.11, the steps that share a common policy are 1)-2), 4)-5), and 3)-6). The function goes through all the states of the best sequence (line 4). At each state, it changes the policy by increasing the score of the move in

⁵RG1: The current state-of-the-art solution methods for the DARP are not able to tackle real-life sized instances.

⁶RG3: Many machine learning methods have yet to be tested on the DARP and compared to state-of-the-art solution methods from the operations research literature.

the sequence (line 5) and decreasing the scores of the other possible moves (lines 6-12). The magnitude of the modifications is set by a parameter: α . Larger value for α results in larger modifications of the scores in the policy.

Algorithm 2 adapt function

```

1: function ADAPT(policy, sequence)
2:   pol'  $\leftarrow$  policy
3:   state  $\leftarrow$  rootState
4:   for move in sequence do
5:     pol'[move]  $\leftarrow$  pol'[move] +  $\alpha$ 
6:     z  $\leftarrow$  0
7:     for m in possible moves of state do
8:       z  $\leftarrow$  z +  $e^{\text{policy}(m)}$ 
9:     end for
10:    for m in possible moves of state do
11:      pol'[m]  $\leftarrow$  pol'[m] -  $\alpha \times \frac{e^{\text{policy}[m]}}{z}$ 
12:    end for
13:    state  $\leftarrow$  play(state, move)
14:  end for
15:  return pol'
16: end function

```

Finally, Algorithm 3 presents the main function. This is the recursive component of the NRPA. The recursivity is defined by the number of levels. At each level, N iterations are performed (line 6). Each level l calls for the level $l - 1$ (line 7). At level 0, simple playouts are performed (lines 2-3). The result of each playout is compared to the best solution found so far. If the result is equal or better, then the new sequence become the current best sequence (lines 8-10). Accepting equally good sequences is used to promote exploration. At the end of each iteration, the main function calls the adapt function on the best sequence (line 12). This increases the scores related to the best sequence, thus encouraging playouts in its neighborhood.

Algorithm 3 main function of the NRPA

```

1: function NRPA(level, policy)
2:   if level = 0 then
3:     return PLAYOUT(rootState, policy)
4:   else
5:     bestScore  $\leftarrow$   $-\infty$ 
6:     for  $N$  iterations do
7:       (newScore, newSequence)  $\leftarrow$  NRPA(level - 1, policy)
8:       if newScore  $\geq$  bestScore then
9:         bestScore  $\leftarrow$  newScore
10:        bestSequence  $\leftarrow$  newSequence
11:      end if
12:      policy  $\leftarrow$  ADAPT(policy, bestSequence)
13:    end for
14:    return (bestScore, bestSequence)
15:  end if
16: end function

```

Figure 4.11 represents an NRPA with 1 level and 2 iterations. To facilitate the comprehension, each step is numbered chronologically. Each step is labeled using the following terminology “s) level l it I-i”. “s” corresponds to the step, “l” corresponds to the level, “it” means iteration, “I” corresponds to the iteration at the higher level, and “i” corresponds to the iteration at the current level.

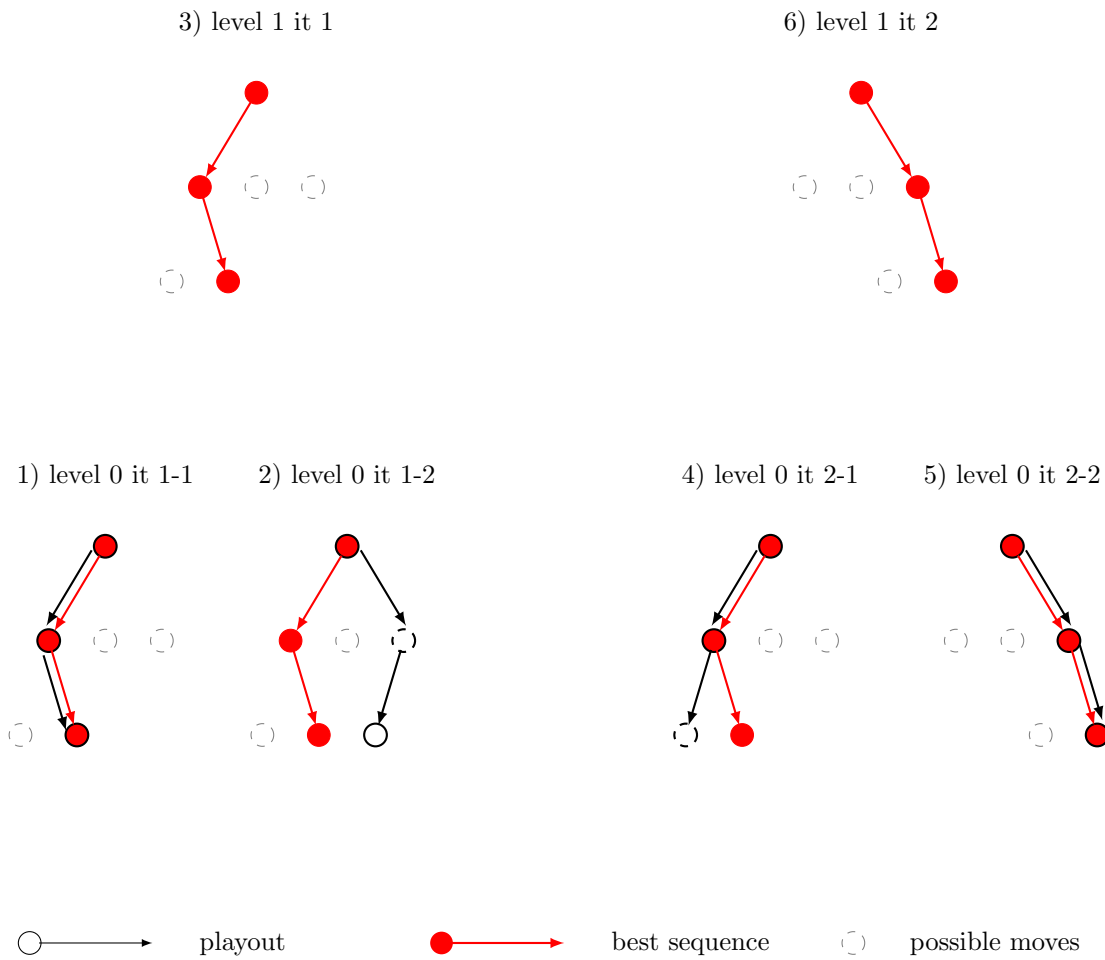


Figure 4.11: NRPA with 1 level and 2 iterations

Step 1) is called recursively from the first iteration at level 1. As it is at level 0, a playout is performed (drawn in black). The best solution is updated (drawn in red) and the policy at this level is updated. The scores of the moves in the best sequence are increased and the scores of the possible moves are decreased. Step 2) is the second iteration at level 0 called from the first iteration at level 1. A playout is performed but does not find a better solution. Thus, the adapt function does the same update as in step 1). In Step 3), the best solution found so far is passed to the first iteration at level 1. Then, the adapt function updates the policy at level 1. Step 4) is called recursively from the second iteration at level 1. The initial policy at Step 4 is a copy of the policy passed from Step 3) to Step 6). It is important to note that modifying the policy at level 0 does not modify the policy at level 1. At this step, the playout does not find a better solution. During step 5), the playout finds a new best solution. Thus, the best sequence is updated and the adapt function updates the policy according to this new best sequence. The possible moves in the adapt function also change. Finally, the best sequence at step 6) is updated according to the new best sequence found at level 0. The resolution ends and the best solution corresponds to the best sequence found so far.

4.4.2.2 NRPA applied to the DARP

In order to apply the NRPA to the DARP, we must define what is a move. Multiple definitions are possible.

Applied to the VRPTW for example, Cazenave et al. (2021) defines a move by a starting node, a destination node, and a vehicle. While it is possible to do the same for the DARP, it would raise several challenges. Indeed, the learning would be difficult because very few playouts would result in feasible solutions. This is due to the time windows and the maximal ride times constraints which are difficult to satisfy. Moreover, the route feasibility testing would require to have both pickup and drop-off in the route. Preliminary experiments confirmed these challenges. Indeed, using this move definition, feasible solutions are seldom reached.

In order to obtain more feasible solutions during the search, we implement the NRPA on a higher level of our solution method. Our method is based on a simple greedy insertion procedure. Starting from an empty solution, requests are inserted one at a time. At each step, the NRPA selects the next request to insert. Once a request is selected, every position for its pickup and its drop-off is tested in every route. Then, the request is inserted at its best position. A terminal state is reached when the selected request cannot be inserted in the solution or when all the requests are inserted. Finally, the policy is updated according to the score of the solution.

In the developed solution method, a move corresponds to two requests: the last inserted request and the candidate request for the next insertion. Scores stored in the policy correspond to the probability for one request to be inserted after another one. As a result, the goal of the NRPA is to find the sequence of requests that gives the best solution when greedily inserted in sequence. The main advantage of this solution method is that the feasibility testing is not managed by the NRPA but by the greedy insertion procedure. On one hand, the NRPA easily retrieves the possible moves. Indeed, they just correspond to the moves containing the last inserted request and any uninserted request. On the other hand, the greedy insertion procedure often results in a feasible insertion. Indeed, as one move gives multiple insertion positions to test, it is very likely for a move to produce a feasible solution. As a result, feasible solutions are obtained more often, facilitating the learning process.

4.4.3 Computational experiments

In the following experiments, we compare two versions of the NRPA with a well-known metaheuristic: the Large Neighborhood Search (LNS). The algorithms are coded in C++ and compiled into a 64-bit single-thread code using g++ 9.4.0. The experiments are performed on a 2.0 GHz AMD EPYC 7702 CPU under a Linux Ubuntu 20.04.4 LTS operating system. The comparison is performed on the benchmark from Cordeau (2006). An instance is named after its number of vehicles and users. For example, instance a2-20 has 2 vehicles and 20 users. For each instance, 5 independent runs are performed.

In the following section, we first describe the parameter tuning process for the NRPA. Then, we present the LNS used for the comparison. Finally, we analyze the results from the comparison.

4.4.3.1 Parameter tuning

There are three parameters to set in the NRPA: the policy adaptation coefficient α , the number of levels, and the number of iterations per level. In the initial work on the NRPA by Rosin (2011), α is set to 1. We try different values between 1 and 2 as our problem is fairly different from their. In the literature, results from the LNS and its variants are often provided for 50000 iterations (Ho et al., 2018). In order to have a similar total number of iterations, we test multiple combinations between the number of levels and iterations. We could have set the NRPA to match the runtime of the LNS. However, preliminary tests showed that it would have resulted in a very small learning time for the NRPA. Thus, an iteration-based comparison would give more learning time. Moreover, this practice prevents any performance gap due to a difference in code optimization between the NRPA and the LNS.

The tested values are presented in Table 4.11. Every setting combinations are tested on three instances: a2-20, a4-40, and a8-80. All results are expressed in terms of gap relative to the scores of the best-known solutions (bks) such as $gap = 100 * \frac{score - bks}{bks}$. The best-known solutions can be found in Ho et al. (2018). Table 4.12 presents the results relative to α , Table 4.13 presents the results relative to the number of levels and iterations, and Table 4.14 presents the detailed results.

| α | nb level - nb iteration | total number of iterations |
|----------|-------------------------|----------------------------|
| 1.0 | 1 - 50000 | 50000 |
| 1.25 | 2 - 224 | 50176 |
| 1.5 | 3 - 37 | 50653 |
| 1.75 | 4 - 15 | 50625 |
| 2.0 | 5 - 9 | 50049 |

Table 4.11: values for tuning the parameters of the NRPA

| α | a2-20 | a4-40 | a8-80 | average |
|----------|-------|-------|-------|---------|
| 1.0 | 0.0 | 1.76 | 7.50 | 3.09 |
| 1.25 | 0.0 | 2.47 | 9.64 | 4.04 |
| 1.5 | 0.0 | 1.10 | 8.32 | 3.14 |
| 1.75 | 0.0 | 0.74 | 9.31 | 3.35 |
| 2.0 | 0.0 | 0.94 | 9.55 | 3.49 |

Table 4.12: parameter tuning results relative to α (gaps in %)

| nb level - nb iterations | a2-20 | a4-40 | a8-80 | average |
|--------------------------|-------|-------|-------|---------|
| 1 - 50000 | 0.0 | 3.05 | 10.56 | 4.54 |
| 2 - 224 | 0.0 | 1.19 | 7.94 | 3.05 |
| 3 - 37 | 0.0 | 1.15 | 7.32 | 2.83 |
| 4 - 15 | 0.0 | 0.60 | 9.35 | 3.32 |
| 5 - 9 | 0.0 | 1.01 | 9.15 | 3.38 |

Table 4.13: parameters tuning results relative to the number of levels and iterations (gaps in %)

Several interpretations can be made from the parameter tuning results. First, the use of recursion provided by multiple levels is essential to ensure good results. Indeed, results for 1 level and 50000 iterations are always the worsts for any given value of α (Table 4.14). Second, on average the best solutions are found when the number of levels and the number of iterations are balanced (i.e., 3 levels and 37 iterations). We can assume that a low number of iterations does not provide enough intensification and a low number of levels does not provide enough diversification. As shown in Table 4.13, the worst average gap is found for 1 level - 50000 iterations (4.54% gap), followed by the ones found for 4 levels -

| α | 1.0 | | | | | 1.25 | | | | | 1.5 | | | | |
|---------------|-------|------|------|------|------|-------|------|------|-------|------|-------|------|------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 |
| nb levels | | | | | | | | | | | | | | | |
| nb iterations | 50000 | 224 | 37 | 15 | 9 | 50000 | 224 | 37 | 15 | 9 | 50000 | 224 | 37 | 15 | 9 |
| a2-20 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| a4-40 | 3.46 | 1.86 | 0.76 | 1.18 | 1.56 | 5.19 | 1.18 | 2.42 | 1.79 | 1.79 | 2.05 | 1.75 | 0.31 | 0.00 | 1.39 |
| a8-80 | 10.18 | 5.19 | 7.90 | 7.15 | 7.06 | 9.38 | 9.44 | 7.33 | 12.34 | 9.73 | 9.99 | 7.66 | 6.52 | 8.60 | 8.84 |
| average | 4.55 | 2.35 | 2.89 | 2.78 | 2.87 | 4.86 | 3.54 | 3.25 | 4.71 | 3.84 | 4.01 | 3.14 | 2.28 | 2.87 | 3.41 |

| α | 1.75 | | | | | 2.0 | | | | |
|---------------|-------|-----|------|------|-----|-------|------|------|-------|------|
| | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 |
| nb levels | | | | | | | | | | |
| nb iterations | 50000 | 224 | 37 | 15 | 9 | 50000 | 224 | 37 | 15 | 9 |
| a2-20 | 0.00 | 0.0 | 0.00 | 0.00 | 0.0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.0 |
| a4-40 | 1.71 | 0.0 | 1.97 | 0.00 | 0.0 | 2.86 | 1.18 | 0.31 | 0.04 | 0.3 |
| a8-80 | 12.13 | 8.7 | 8.31 | 7.53 | 9.9 | 11.11 | 8.72 | 6.56 | 11.14 | 10.2 |
| average | 4.61 | 2.9 | 3.43 | 2.51 | 3.3 | 4.66 | 3.30 | 2.29 | 3.73 | 3.5 |

Table 4.14: parameter tuning results (gaps in %)

15 iterations and 5 levels - 9 iterations (respectively 3.32% and 3.38% gap). Third, no obvious link can be made between α and the results (Table 4.12).

Finally, we choose $\alpha = 1.5$, $nbLevel = 3$, $nbIteration = 37$. This choice was made given the previous analysis and the fact that this configuration shows the best results on average. We call this NRPA with this particular setting NRPA-3-37. To put our choice into perspective, one must not forget that this configuration also provides the greatest total number of iterations (i.e., 50653 iterations). By modifying the NRPA algorithm, it could have been possible to precisely select the total number of iterations. This would have allowed a more precise comparison between the different configurations. However, the parameter tuning is not the focus of our study and the difference between the total numbers of iterations of the different settings tested is less than 2% (Table 4.11). A second NRPA is used in the experiments to compensate for the low total number of iterations of the NRPA-3-37. Indeed, it is approximately 20 times lower than in Rosin (2011) (i.e., 50653 vs. 1000000 iterations). Given the limited number of iterations, the learning process might be incomplete. As a result, we also provide the results from a second NRPA with the 3 levels and 100 iterations per level as in Rosin (2011). However, we keep $\alpha = 1.5$ as it has shown the best results in the parameter tuning. This NRPA is named NRPA-3-100.

4.4.3.2 Large Neighborhood Search (LNS)

Since its introduction in Shaw (1998), the LNS has been extensively studied by the OR community. Up to this day, it is one of the most effective methods for solving routing problems (Pisinger and Ropke, 2019). The LNS is based on a ruin and repair principle: it improves the current solution by ruining it and repairing it using low-level ruin and repair heuristics. The LNS stops when the stopping criterion is reached (e.g., time limit or iteration limit).

The LNS developed has 2 ruin operators and 4 repair operators: random removal, historical node-pair removal, greedy insertion, and k-regret insertion with $K = 2, 3, 4$. We refer the reader to Pisinger and Ropke (2007) for a more detailed description of these operators. The stopping criterion is a limit on the number of iterations. This limit is set to 50653 iterations. It corresponds to the total number of iterations of the NRPA-3-37. We use the same parameter setting as in Tellez et al. (2018) for the LNS. The implementation is summarized in Table 4.15.

4.4.3.3 Results analysis

The results presented in Table 4.16 compare the three algorithms (i.e., NRPA-3-37, NRPA-3-100, and LNS) on the benchmark A from Cordeau (2006).

First, both NRPAs perform relatively well compared to the LNS with an average gap of 2.02% for NRPA-3-37 and 0.97% for NRPA-3-100. However, the best results are almost exclusively obtained with

| value | description |
|---------------------------|---|
| 50653 | iteration limit |
| 5% | record-to-record acceptance percentage |
| [0.1, 0.45] | destruction size for the LNS |
| random, history node-pair | ruin operators |
| best, k-regret | recreate operators |
| 6 | roulette wheel parameter for the historical node-pair ruin operator |
| 2, 3, 4 | k parameter for the k-regret ruin operator |

Table 4.15: LNS parameter setting

| instance | NRPA-3-37 | | NRPA-3-100 | | LNS | | gap with LNS (%) | |
|----------|-----------|-----------|------------|-----------|---------|-----------|------------------|------------|
| | runtime | objective | runtime | objective | runtime | objective | NRPA-3-37 | NRPA-3-100 |
| a2-16 | 71 | 294.25 | 1428 | 294.25 | 2 | 294.25 | 0.00 | 0.00 |
| a4-16 | 50 | 282.68 | 1082 | 282.68 | 3 | 282.68 | 0.00 | 0.00 |
| a3-18 | 57 | 300.48 | 1379 | 300.48 | 3 | 300.48 | 0.00 | 0.00 |
| a2-20 | 145 | 344.83 | 3112 | 344.83 | 5 | 344.83 | 0.00 | 0.00 |
| a2-24 | 168 | 431.12 | 3615 | 431.12 | 8 | 431.12 | 0.00 | 0.00 |
| a3-24 | 149 | 344.83 | 3211 | 344.83 | 7 | 344.83 | 0.00 | 0.00 |
| a4-24 | 135 | 375.02 | 2604 | 375.02 | 6 | 375.02 | 0.00 | 0.00 |
| a3-30 | 299 | 494.85 | 6434 | 494.85 | 10 | 494.85 | 0.00 | 0.00 |
| a4-32 | 239 | 489.16 | 5273 | 485.50 | 9 | 485.50 | 0.75 | 0.00 |
| a3-36 | 388 | 583.30 | 7293 | 583.30 | 15 | 583.54 | -0.04 | -0.04 |
| a4-40 | 443 | 559.43 | 8917 | 557.69 | 17 | 558.45 | 0.18 | -0.14 |
| a5-40 | 494 | 513.58 | 9208 | 509.07 | 19 | 498.41 | 3.04 | 2.14 |
| a4-48 | 887 | 696.50 | 15793 | 672.37 | 29 | 675.92 | 3.04 | -0.53 |
| a6-48 | 681 | 642.66 | 13107 | 624.06 | 30 | 612.07 | 5.00 | 1.96 |
| a5-50 | 891 | 694.59 | 17982 | 714.35 | 33 | 693.13 | 0.21 | 3.06 |
| a7-56 | 760 | 772.73 | 15855 | 762.68 | 38 | 743.59 | 3.92 | 2.57 |
| a5-60 | 1237 | 845.26 | 21967 | 825.72 | 45 | 821.60 | 2.88 | 0.50 |
| a6-60 | 945 | 852.43 | 18772 | 847.49 | 43 | 843.60 | 1.05 | 0.46 |
| a8-64 | 1064 | 811.70 | 21068 | 788.33 | 57 | 776.81 | 4.49 | 1.48 |
| a7-70 | 1698 | 975.73 | 28135 | 954.90 | 63 | 920.89 | 5.96 | 3.69 |
| a6-72 | 1712 | 1009.12 | 31292 | 956.14 | 73 | 946.05 | 6.67 | 1.07 |
| a8-80 | 2238 | 1007.38 | 47210 | 984.62 | 97 | 983.58 | 2.42 | 0.11 |
| a7-84 | 2683 | 1133.74 | 53528 | 1118.64 | 109 | 1071.61 | 5.80 | 4.39 |
| a8-96 | 3000 | 1336.86 | 65114 | 1327.89 | 153 | 1295.25 | 3.21 | 2.52 |
| average | 851 | 658.01 | 16807 | 649.20 | 36 | 640.75 | 2.02 | 0.97 |

Table 4.16: comparison between NRPA-3-37, NRPA-3-100 and LNS

the LNS (21 out of 24). Second, the NRPA's performance on small instances is on par with the LNS for a similar total number of iterations. Indeed, up to 40 users, the gap between the NRPA-3-37 and the LNS is less than 1% (except for instance a5-40). Third, the good performance on small instances is not translated to larger instances. Indeed, the gaps for bigger instances between 48 and 96 users increase up to 6.67% for NRPA-3-37. In fact, the average gap for small instances between 16 and 40 users is 0.33%, while the average gap for large instances between 48 and 96 users is 3.72% for NRPA-3-37. This highlights the difficulty for the NRPA to solve large instances. NRPA-3-100 shows that more learning time allows to reach better results. Indeed, the average gap for large instances reaches 1.77% which represents a 1.95% ($= 3.72 - 1.77$) improvement over NRPA-3-37. However, it is still not on par with the LNS. Finally, the computation speed of the LNS is much greater than the NRPA. For this benchmark, the LNS takes on average 36 seconds to generate the 50653 solutions while NRPA-3-37 takes 852 seconds. This can be easily explained. At each iteration, the NRPA builds a solution from scratch while the LNS starts with a partial solution. While the NRPA-3-100 gives better results, it has an average computation time of 16807 seconds (i.e., approximately 4.5 hours).

Given the results of this comparison, we can conclude that the NRPA is not competitive with the LNS.

4.4.4 Conclusion and perspectives

In this study, we implemented the Nested Rollout Policy Adaptation (NRPA) algorithm for solving the Dial-A-Ride Problem (DARP). The NRPA is integrated into a greedy insertion framework that builds a solution from scratch by sequentially inserting the users. The NRPA is in charge of selecting the next user to insert according to the last inserted one. The solution method is tested on benchmark A from Cordeau (2006) and compared to a Large Neighborhood Search (LNS) metaheuristic. While the NRPA is able to find similar results on instances with up to 40 users, the performance gap increases for larger instances (3.37% gap for instances with 48 or more users). In addition, the computation duration is much slower for the NRPA. To summarize the findings, the NRPA is able to solve the DARP using an online learning process. However, it is not competitive with the LNS, especially for large instances.

We identified several perspectives with regard to this work. First, it could be possible to improve the computation speed. Currently, the NRPA builds the solution from scratch, but it could start from partial solutions instead. Second, the greedy insertion was the only insertion procedure in this study. But, it could be possible to add other insertion mechanisms such as the k-regret insertion (Braekers et al., 2014) to diversify the search. However, this significantly increases the size of the decision space of the NRPA. Indeed, instead of just choosing the next user to insert, it would select the next user and an insertion heuristic as well. As a consequence, the size of NRPA's search tree would significantly increase.

Chapter 5

The Commuting Dial-A-Ride Problem (Com-DARP)¹

Highlights:

- The Commuting Dial-A-Ride Problem (Com-DARP) is a variant of the Dial-A-Ride Problem (DARP) where each user has a morning request, an afternoon request, and a daily maximum ride time.
- The problem is solved with a small and large neighborhood search metaheuristic coupled with a speed-up mechanism that uses the dynamic precedence relations between users.
- The solution method is compared to two state-of-the-art metaheuristics on instances from a new benchmark generated for this problem.
- Managerial insights show that solving a commuting problem using the Com-DARP formulation provides better solutions than using the corresponding formulation with one morning and one afternoon DARP.

5.1 Introduction

In a dial-a-ride system, a fleet of vehicles transports a set of users from their individual pickup location to their individual drop-off location. The vehicles are shared by several users. This transportation system usually benefits schoolchildren, patients, the elderly, disabled people, commuting workers, and people in rural areas. It can be more practical than regular public transportation systems, especially for vulnerable groups and in regions with a low population density. With the aging population and the need to reduce carbon emissions, more and more people can benefit from such systems. Vulnerable users can travel within their community and access health services more easily and more comfortably, and commuters

¹This chapter is an updated version of the journal article Chane-Haï et al. (2023b) (submitted).

can reduce their carbon footprint and save money by sharing a ride instead of using their personal car.

To solve a Dial-A-Ride Problem (DARP), a routing plan that satisfies all users' demands, respects the constraints related to the quality of service, and minimizes the defined objective (e.g., routing cost, pollution) must be created. Unfortunately, as an extension of the Traveling Salesman Problem (TSP), the DARP is NP-hard. It is therefore difficult to solve real-life sized instances even for the simplest versions of the DARP. Today, exact methods can only tackle problems with up to a few dozens of requests, and metaheuristics push this limit from one to two hundred requests. However, real-life instances can contain thousands of requests and integrate particularities such as heterogeneous vehicles, different types of users, multiple-depot, etc. With these additional characteristics, the problem becomes even more difficult to solve.

In this chapter, we propose a generalization of the DARP: the Commuting Dial-A-Ride Problem (Com-DARP). In this variant, a user has two requests: one for the morning period and one for the afternoon period. The first request corresponds to the outgoing morning trip and the second request corresponds to the inbound afternoon trip. The novelty introduced here is that a user has both a maximum ride time for each request and a daily maximum ride time encompassing both requests. When the morning and afternoon problems are separated, the daily maximum ride time is distributed equally across the morning and the afternoon requests. On the other hand, when the morning and afternoon requests are linked by a daily maximum ride time, the ride time can be distributed unevenly. Thus, a user may have a long ride in the morning which is then compensated for by a short ride in the afternoon. This situation expands the number of feasible solutions in the research space. As illustrated by Figure 5.1, the Com-DARP allows for solutions with an unbalanced ride time distribution between the outgoing and inbound requests. As a result, new solutions are available for the same daily maximum ride time.

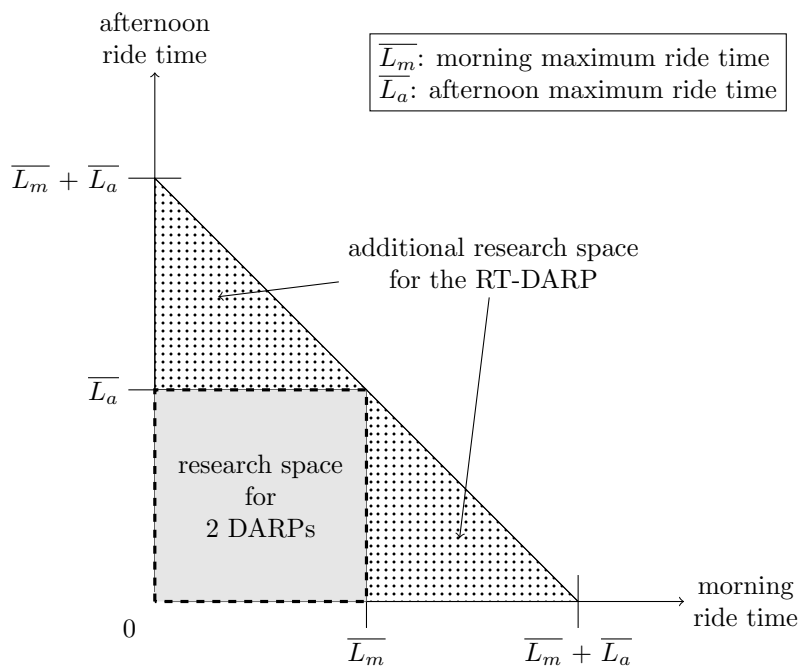


Figure 5.1: daily maximum ride time decomposition for the Com-DARP

This research is motivated by a real-life application. In the Auvergne-Rhône-Alpes region in France, the company GIHP is in charge of the daily transportation of children with disabilities. Every day, around 1800 children commute using this transportation service, from their homes to Medico-Social-Institutions (MSI) in the morning and from the MSIs to their homes in the afternoon. To simplify the generation of the routing plan, the daily maximum ride time of the users is broken down evenly between the morning trip and the afternoon trip. The morning and afternoon routes can thus be planned independently. This procedure halves the problem, making it easier to solve. However, it also introduces a bias in the

ride time distribution among the children, for some children will have both a long morning and a long afternoon trip, while others will have two short trips. This results in unfair situations. Our first goal is to integrate the morning and afternoon problems into one daily problem. By combining the morning and afternoon problems through the daily maximum ride time, we limit users' daily ride time. Children with a long morning trip must have a short afternoon trip and vice versa.

The rest of this chapter is organized as follows. A literature review is presented in Section 5.2. The mathematical formulation for the Com-DARP is introduced in Section 5.3. Our solution method is detailed in Section 5.4 and the subsequent computational results are detailed in Section 5.5. Final conclusions are drawn in Section 5.6.

5.2 Literature review

5.2.1 The Dial-A-Ride Problem

The DARP is one of the classic problems in the field of operations research. It was first introduced by Psaraftis (1980), but the formulation introduced in Cordeau and Laporte (2003) serves as the standard. In this section, we briefly present the main contributions related to the DARP. For a detailed literature review, we refer the reader to the following surveys: Parragh et al. (2008), Molenbruch et al. (2017b), and Ho et al. (2018). Similarly to Ho et al. (2018), we will prefer the term drop-off when writing about users and delivery when writing about goods.

Multi-trip problems The Com-DARP belongs to the multi-trip problem category. In a multi-trip Vehicle Routing Problem (VRP), a vehicle is allowed to leave and return to a depot several times during the planning period. This feature has been introduced in Fleischmann (1990) to model more realistic VRPs where vehicles return to the depot multiple times because of high demand and short travel times. Most of the research incorporating this multi-trip component concerns VRP variants. We refer the reader to Cattaruzza et al. (2018) for a literature review on the multi-trip VRP.

In the field of DARP, the multi-trip component has been integrated to take into account real constraints and to improve the quality of service, especially in the domain of sanitary transportation. Parragh et al. (2012) take into account drivers' lunch break that can happen at any node. In Zhang et al. (2015), Masmoudi et al. (2016), and Lim et al. (2017), coffee breaks are added. While the lunch break is taken at the depot, the coffee breaks can take place within a route. In addition, a vehicle must be disinfected each time it returns to the depot, for sanitary reasons. In the previously cited articles, each user has only one request. Thus, the multi-trip component is related only to the vehicles. Yet it can also be related to users, which is the case when users have multiple requests. In Braekers and Kovacs (2016), Tellez et al. (2022), and Lindstrøm and Røpke (2022), users have multiple requests over the planning horizon which can be several days. Respectively, drivers', service times', and groups' consistency are taken into account to ensure a good quality of service.

The novelty of our problem lies in the daily maximum ride time that creates a dependency between the users' requests.

Solution method Few articles use exact methods to solve the DARP. Branch and cut algorithms are used in Rist and Forbes (2021) and Cordeau (2006), and branch and price and cut algorithms are used in Luo et al. (2019), Qu and Bard (2015), and Ropke and Cordeau (2009). However, the performance of exact methods decreases rapidly as the problem size increases, because of the NP-hardness of the problem. To our knowledge, exact methods can solve classic DARP with up to 96 users and fewer for more elaborated DARP variants.

With regard to the limited performance of exact methods, the majority of the research has revolved around the development of heuristics to solve the DARP. One of these is the Large Neighborhood Search (LNS). This method has been extensively studied because it has consistently shown good results on routing problems. The LNS was first introduced in Shaw (1998), coupled with an exact method. Later, in Schrimpf et al. (2000), the LNS was coupled with a heuristic under the name of *ruin and recreate*. Its

principle is straightforward: it improves the current solution by ruining it and repairing it. The LNS ends when a stopping criterion is reached (e.g., time limit or iteration limit).

Many LNS variants have been developed, such as the Variable Neighborhood Search (VNS) or the Adaptive Large Neighborhood Search (ALNS). The VNS was first introduced by Mladenović and Hansen (1997) and first used on a DARP by Parragh et al. (2010). The ALNS was first used on a DARP by Ropke and Pisinger (2006). Both methods use insertion and removal operators. In a VNS, the operators are systematically changed while in an ALNS, the operators are chosen according to a weighted probability. In the ALNS, the weighted probabilities are dynamically adjusted during the resolution. However, Turkeš et al. (2021) proved that the adjustment of the adaptive weights for the selection of the operators does not significantly improve the performance of the ALNS.

A promising metaheuristic has been introduced in Dumez et al. (2021b) for the Generalized Vehicle Routing Problem with Time Windows (GVRPTW). The metaheuristic, called Small and Large Neighborhood Search (SLNS), is a variant of the LNS. In the SLNS, removals alternate between small and large proportions of the inserted requests. Most of the time, small-sized removals are performed. But once in a while, a large-sized removal is used to reach a different part of the research space. This algorithm is competitive with the state-of-the-art solution methods for VRP variants and yet is very simple. Laporte (2009) argues that simplicity is a strength compared to “over-engineered” metaheuristics that are more difficult to implement. To our knowledge, the SLNS has not yet been tested on the DARP despite its promising results on the GVRPTW.

In this work, we extend the SLNS approach to the Com-DARP.

Feasibility testing One of the most critical barriers to solving the DARP is the schedule evaluation. Testing the feasibility of a route is more difficult in DARP than in Pickup and Delivery Problems (PDP), due to the users’ maximum ride times Cordeau and Laporte (2007). To tackle this problem, many articles develop a testing procedure based on the forward time slack from Savelsbergh (1992). This procedure was introduced for the DARP by Cordeau and Laporte (2003). This algorithm has a $O(n^2)$ complexity and serves as a basis for many route-scheduling algorithms. In Hunsaker and Savelsbergh (2002), a three-pass algorithm with a $O(n)$ complexity is presented. However, it may incorrectly declare unfeasibility due to ride time constraints. A revised procedure of this algorithm is presented in Tang et al. (2010) with a $O(n^2)$ complexity. As demonstrated in Gschwind (2019), this procedure gives better results than the linear time algorithm of Firat and Woeginger (2011). More recently, a constant-time feasibility check with $O(n^3)$ preprocessing complexity has been introduced in Gschwind and Drexel (2019).

To further speed up the algorithms, other tests have been developed, which are well summarized in Braekers et al. (2014). Recently, a precedence test was introduced succinctly in Malheiros et al. (2021). In this test, called feasible search range, the algorithm saves the feasible positions for insertions for each node in each route. Thus, infeasible positions are avoided. The majority of the computation is done during a preprocessing step with a $O(n^2)$ complexity. The information is then updated in $O(n^2)$ and retrieved in $O(1)$.

In this work, we adapt this precedence test for the Com-DARP as the dependence between requests dynamically impacts the precedence constraints between nodes.

5.2.2 Contributions

With its contributions, this work tries to bridge research gaps RG1² and RG5³. First, the Commuting Dial-A-Ride Problem (Com-DARP) is presented with its mathematical model. In this problem, a user has multiple requests over the planning horizon and also a global maximum ride time for the set of requests. Second, we develop a new route feasibility check that exploits the precedence constraints between nodes and adapts dynamically during the computation. Third, we compare the performance of the new SLNS framework with the classic LNS framework. Finally, we derive managerial insights from the comparison between the Com-DARP and the 2-DARP, which is the equivalent problem where the morning and evening DARPs are solved independently.

²RG1: The current state-of-the-art solution methods for the DARP are not able to tackle real-life sized instances.

³RG5: There are few works on shared constraints for the DARP. In particular, there is no study on a shared maximum ride time constraint.

5.3 Problem description

This section introduces the mathematical formulation of the Com-DARP. To facilitate its comprehension, the complete list of all the notations used in this chapter is provided in Appendix A.2. In order to better fit the problem, The notation differs from the one used in Section 3.3.1 for the classic DARP.

5.3.1 General problem setting

The Com-DARP is modeled on a complete directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$. \mathcal{A} is the set of arcs and \mathcal{N} is the set of nodes. $\mathcal{P} \subset \mathcal{N}$ is the set of pickup nodes, $\mathcal{D} \subset \mathcal{N}$ is the set of drop-off nodes, $\mathcal{O}^+ \subset \mathcal{N}$ is the set of starting depot nodes, and $\mathcal{O}^- \subset \mathcal{N}$ is the set of ending depot nodes. In practice, a starting depot and an ending depot can correspond to the same location. Each starting depot has a corresponding ending depot such that $\mathcal{E} \subset \mathcal{O}^+ \times \mathcal{O}^-$ is the set of pairs starting and ending depots. Finally, \mathcal{U} is the set of users, \mathcal{R} is the set of requests, and \mathcal{K} is the set of vehicles.

Each arc $(i, j) \in \mathcal{A}$ has a travel duration $T_{ij} \geq 0$ and a travel distance $D_{ij} \geq 0$. All travel times and distances are non-negative and travel times satisfy the triangle inequality.

Each user $u \in \mathcal{U}$ has a set of requests $\mathcal{R}_u \subset \mathcal{R}$. Within this set, each request $r \in \mathcal{R}$ corresponds to a pickup node $P_r \in \mathcal{P}$ and a drop-off node $D_r \in \mathcal{D}$. The transportation respects the maximum ride time of the request $\overline{L}_r \geq 0$. Additionally, there is a maximum ride time associated with this set of requests, which is $\overline{L}_u \geq 0$.

At each node, the arrival time is in its time window $[A_i, B_i]$ with $A_i \geq 0, B_i \geq 0$. The service duration at the node is $S_n \geq 0$ and the load variation is Q_n . For a request $r \in \mathcal{R}$ with n^+ the pickup node and n^- the drop-off node, $Q_{n^+} = -Q_{n^-}$. For a depot node $o \in \mathcal{O}^+ \cup \mathcal{O}^-$, the service duration is null: $S_o = 0$ as is the demand: $Q_o = 0$.

Each vehicle $k \in \mathcal{K}$ has a capacity $\overline{Q}_k \geq 0$, a fixed cost $C_k^f \geq 0$, a cost related to the distance $C_k^d \geq 0$ and a cost related to the ride time $C_k^t \geq 0$. As a vehicle can do multiple trips, it has a set of starting depots $\mathcal{O}_k^+ \in \mathcal{O}^+$ and a set of ending depots $\mathcal{O}_k^- \in \mathcal{O}^-$. $\mathcal{E}_k \in \mathcal{E}$ is the set of pairs of starting and ending depots for this vehicle. Each individual trip is limited by a maximum trip ride time $\overline{L}_{o_k^+} \geq 0, o_k^+ \in \mathcal{O}_k^+$ and the vehicle's overall ride time is limited by the maximum total ride time $\overline{L}_k \geq 0$.

The mathematical model presented here is a generalization of the Com-DARP. This formulation corresponds exactly to a multi-request multi-trip multi-depot heterogeneous fleet dial-a-ride problem. In the case of the Com-DARP, the time horizon H can be used to define a morning period $[0, H/2]$ and an afternoon period $[H/2, H]$. As a result, a vehicle $k \in \mathcal{K}$ has only two pairs of starting and ending depot nodes and a user $u \in \mathcal{U}$ has only two requests. Time windows for the first pairs of depot nodes and the morning requests close during the morning period. Similarly, time windows for the second pairs of depot nodes and the afternoon requests open during the afternoon period.

5.3.2 Mathematical model

The following variables are used to solve the problem:

- x_{ij}^k : a binary variable which is equal to 1 if the vehicle $k \in \mathcal{K}$ uses the arc $(i, j) \in \mathcal{A}$, and 0 otherwise;
- l_i^k : a positive integer variable which corresponds to the number of users in the vehicle $k \in \mathcal{K}$ after leaving the node $i \in \mathcal{N}$;
- w_i^k : a positive continuous variable which corresponds to the service time of the vehicle $k \in \mathcal{K}$ at node $i \in \mathcal{N}$;
- w_i : a positive continuous variable which corresponds to the service time at node $i \in \mathcal{N}$;
- y^k : a binary variable which is equal to 1 if the vehicle $k \in \mathcal{K}$ is used;
- z_r : a positive continuous variable which corresponds to the ride time of request r .

The Com-DARP can be formulated with the following mixed integer program:

- the objective is to minimize the total transportation cost

$$\min \sum_{k \in \mathcal{K}} C_k^f y^k + \sum_{k \in \mathcal{K}} \left(\sum_{(o_k^-, o_k^+) \in \mathcal{E}_k} C_k^h (w_{o_k^-} - w_{o_k^+}) + \sum_{(i,j) \in \mathcal{A}} C_k^d D_{ij} x_{ij}^k \right); \quad (5.1)$$

- the flow is conserved

$$\sum_{(j,i) \in \mathcal{A}} x_{ji}^k - \sum_{(i,j) \in \mathcal{A}} x_{ij}^k = 0, \quad \forall i \in \mathcal{P} \cup \mathcal{D}, k \in \mathcal{K}; \quad (5.2)$$

- every pickup node is served once and only once

$$\sum_{k \in \mathcal{K}} \sum_{(i,P_r) \in \mathcal{A}} x_{iP_r}^k = 1, \quad \forall r \in \mathcal{R}; \quad (5.3)$$

- every pickup and drop-off pair is served by one and only one vehicle

$$\sum_{(P_r,j) \in \mathcal{A}} x_{P_r j}^k - \sum_{(i,D_r) \in \mathcal{A}} x_{i D_r}^k = 0, \quad \forall r \in \mathcal{R}, k \in \mathcal{K}; \quad (5.4)$$

- the service time at a node is set according to the vehicle service time at the same node

$$w_i = \sum_{k \in \mathcal{K}} w_i^k, \quad \forall i \in \mathcal{N}; \quad (5.5)$$

- the arrival time at a node matches its incoming arc duration and previous node service duration

$$w_j^k \geq w_i^k + S_i + T_{ij} - M_{ij}(1 - x_{ij}^k), \quad \forall (i,j) \in \mathcal{A}, k \in \mathcal{K} \quad (5.6)$$

$$M_{ij} = B_i + S_i + T_{ij};$$

- a pickup is served before its corresponding drop-off

$$w_{P_r} + S_{P_r} + T_{P_r D_r} \leq w_{D_r} + M_r \left(1 - \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N}} x_{i P_r}^k \right), \quad \forall r \in \mathcal{R} \quad (5.7)$$

$$M_r = B_{P_r} + S_{P_r} + T_{P_r D_r};$$

- the arrival time at a node is within its time windows

$$A_j \leq w_j + M_j \left(1 - \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N}} x_{ij}^k \right) \leq B_j, \quad \forall j \in \mathcal{N} \quad (5.8)$$

$$M_j = A_j;$$

- a vehicle that leaves also returns

$$\sum_{j \in \mathcal{P}} x_{o_k^+ j}^k - \sum_{i \in \mathcal{D}} x_{i o_k^-}^k = 0, \quad \forall k \in \mathcal{K}; \quad (5.9)$$

- a vehicle is used if it leaves the depot

$$M^k y^k \geq \sum_{(o_k^+, i) \in \mathcal{O}_k^+ \times \mathcal{P}} x_{o_k^+ i}^k, \quad \forall k \in \mathcal{K}; \quad (5.10)$$

$$M^k = |\mathcal{O}_k^+|;$$

- the load at a node depends on the cumulative load of the route and the load variation at this node

$$l_j^k \geq l_i^k + Q_j - M_j^k(1 - x_{ij}^k), \quad \forall (i, j) \in \mathcal{A}, k \in \mathcal{K} \quad (5.11)$$

$$M_j^k = \overline{Q}_k + Q_j;$$

- the load respects the capacity of the vehicle

$$l_i^k \leq \overline{Q}_k \sum_{j \in \mathcal{N}} x_{ij}^k, \quad \forall i \in \mathcal{P}, k \in \mathcal{K}; \quad (5.12)$$

- the ride time is defined by the service time at pickup and the arrival times at pickup and drop-off

$$w_{D_r} - (w_{P_r} + S_{P_r}) = z_r, \quad \forall r \in \mathcal{R}; \quad (5.13)$$

- the ride time for completing a request respects the request max ride time

$$z_r \leq \overline{L}_r, \quad \forall r \in \mathcal{R}; \quad (5.14)$$

- the ride time for completing the set of requests of a user respects the user max ride time

$$\sum_{r \in \overline{r}_u} z_r \leq \overline{L}_u, \quad \forall u \in \mathcal{U}; \quad (5.15)$$

- the route duration respects the route max duration

$$w_{o_k^+} - w_{o_k^-} \leq \overline{L}_{o_k^+}, \quad \forall k \in \mathcal{K}, (o_k^+, o_k^-) \in \mathcal{E}_k; \quad (5.16)$$

- the cumulative ride time of a vehicle over multiple routes respects the vehicle max ride time

$$\sum_{(o_k^+, o_k^-) \in \mathcal{E}_k} w_{o_k^+} - w_{o_k^-} \leq \overline{L}_k, \quad \forall k \in \mathcal{K}; \quad (5.17)$$

- each variable is defined in its respective set

$$x_{ij}^k \in \{0, 1\}, \quad \forall i \in \mathcal{N}, j \in \mathcal{N}, k \in \mathcal{K};$$

$$l_i^k \in \mathbb{Z}^+, \quad \forall i \in \mathcal{N}, k \in \mathcal{K};$$

$$w_i^k \in \mathbb{R}^+, \quad \forall i \in \mathcal{N}, k \in \mathcal{K};$$

$$w_i \in \mathbb{R}^+, \quad \forall i \in \mathcal{N};$$

$$y^k \in \{0, 1\}, \quad \forall k \in \mathcal{K}.$$

5.4 Solution method

In this section, we detail our solution method for solving the Com-DARP. This method is based on an SLNS metaheuristic associated with a speed-up mechanism using a precedence filter (PF). We name the complete metaheuristic SLNS-PF. The rest of this section is organized as follows: the SLNS framework is described in Section 5.4.1, the insertion and removal operators are presented in Section 5.4.2, the mechanisms behind the precedence filter are explained in detail in Section 5.4.3 and the feasibility tests are listed in Section 5.4.4.

Algorithm 4 SLNS framework

```

1:  $S$ : initial solution
2:  $iter \leftarrow 0$ 
3: while stopping criterion is not reached do
4:   if  $iter < \omega$  then
5:      $S' \leftarrow S$ 
6:      $iter \leftarrow iter + 1$ 
7:     randomly select a removal operator  $op^- \in \S^-$ 
8:     randomly select a removal size  $\phi \in [\delta_{small}, \Delta_{small}]$ 
9:   else
10:     $S' \leftarrow S^*$ 
11:     $iter \leftarrow 0$ 
12:    randomly select a removal operator  $op^- \in \S^-$ 
13:    randomly select a removal size  $\phi \in [\delta_{large}, \Delta_{large}]$ 
14:   end if
15:   remove a proportion  $\phi$  of requests from  $S'$  using  $op^-$ 
16:   randomly select an insertion operator  $op^+ \in \S^+$ 
17:   insert requests  $S'$  using  $op^+$ 
18:   if  $f(S') < f(S)$  or  $iter = 0$  then
19:      $S \leftarrow S'$ 
20:   end if
21:   if  $f(S') < f(S^*)$  then
22:      $S^* \leftarrow S'$ 
23:      $iter \leftarrow 0$ 
24:   end if
25: end while
26: return  $S^*$ 

```

5.4.1 Small and Large Neighborhood Search (SLNS) framework

To solve our problem, we implemented the SLNS as presented in Dumez et al. (2021b). We chose to compare our metaheuristic with the LNS as it performs well on routing problems and is simple to implement. The SLNS is presented in Algorithm 4. We refer the reader to Ropke and Pisinger (2006) for the LNS algorithm.

Three solutions are used in the SLNS. S^* is the best solution found during the resolution. S' is the solution on which the insertions and removals are performed. S is the current solution, used to reset S' if S' does not improve the best solution S^* . The algorithm is initialized at lines 1-2: the initial solution is loaded and $iter$ is set to 0. The core of the algorithm corresponds to the while loop that starts at line 3. Until a stopping criterion is reached, the SLNS will perform removals and insertions. The removal is done between the line 4 and 15. If $iter$ is smaller than the small iteration threshold ω , a small removal is selected (lines 4-9). Otherwise, if the number of small iterations since the last new best solution reaches the threshold ω , then a large removal is selected (lines 9-14). Small removals are performed from the current solution (line 5) and large removals are performed from the best solution (line 10). The solution is repaired at lines 16 and 17. Once a new solution is obtained, the acceptance criterion is applied at lines 18-24. The acceptance is performed in two steps. In the first step, the current solution is updated if the new solution is better or if a large removal has been performed. In the second step, the best solution is updated if the new solution is better. In addition, the reader can note that $iter$ is incremented for small removals (line 6), and reset when a large removal is performed or a new best solution has been found (lines 11 and 23).

There are two main differences between the SLNS and the LNS. First, the SLNS has small and large moves. In LNS, the size of each removal is randomly chosen within a rather large interval. On the other hand, the SLNS does a lot of small removals and, once in while, a large removal. Thus, many iterations are performed. These small removals really intensify the search. Second, the acceptance in the LNS is based on only one acceptance criterion. By contrast, the SLNS has two methods of acceptance: after a new best solution has been found or after a large removal. Hence, the SLNS leaves local optimums when doing a large move (exploration) but after a lot of small moves (intensification). To summarize, while the LNS might leave a region of the research space quickly, the SLNS leaves the same region only after an intensification phase.

5.4.2 Operators

The LNS operators are used to remove and insert users in the solution. They are key components of any LNS heuristic. A good set of operators results in a good search intensification while allowing sufficient diversification. Many operators have been developed but as demonstrated in Christiaens and Vanden Berghe (2020), it is possible to obtain very good results with only a few carefully selected operators. We use the same operators as in Tellez et al. (2018) (2 removals, 4 insertions). Following the recommendations in Cordeau et al. (2002), these operators keep the heuristics simple, yet fast and accurate. We refer the reader to Pisinger and Ropke (2007) for a more in-depth explanation of these operators.

5.4.2.1 Removal operators

The removal operators select a portion ϕ of the inserted requests according to a criterion and remove them from the solution. The two removal operators are used: the random removal and the historical node-pair removal.

The random removal randomly selects a proportion ϕ of inserted requests and removes them from the solution.

The historical node-pair removal selects a proportion ϕ of inserted requests and removes them based on their “historical” score. This operator favors the removal of requests that were inserted in better positions in previous solutions. A score h_{ij} is associated with each arc $(i, j) \in \mathcal{A}$ with an initial score set to infinity. At each new solution and for each inserted arc, this score is updated with the new objective value if it is better. Similarly, a score h_r is associated with each inserted request $r \in \mathcal{R}$. This score is the sum of the scores of the four incident arcs (incoming and outgoing arcs for pickup and drop-off). Then,

the list of inserted requests \mathcal{R}_{ins} is ordered by decreasing scores h_r . The operator removes the request at the position $\xi^\psi \times |\mathcal{R}_{ins}|$, ξ is a random number between $[0, 1[$, and $\psi \in \mathbb{Z}$ is a deterministic parameter. This rule used in Pisinger and Ropke (2007) gives higher probabilities of removing the requests with a higher score (i.e., worst score).

5.4.2.2 Insertion operators

The insertion operators try to place uninserted requests into the partial solution while respecting the constraints. The following insertion operators have been used: greedy insertion and k-regret insertion.

The greedy insertion places the uninserted requests sequentially according to their insertion cost. For each uninserted request, its best insertion is computed. A request's best insertion corresponds to the positions for its pickup and its drop-off with the lowest cost. Then, the request with the lowest insertion cost is inserted. The operator repeats this procedure until all the requests are inserted or until there is no more feasible insertion.

The k-regret insertion places the uninserted requests sequentially according to their k-regret cost. The change in the objective value when inserting the request r in its q th best position is represented by Δf_r^q . Therefore, Δf_r^1 corresponds to the insertion of the request r in its best position, Δf_r^2 corresponds to the insertion of the request r in its second best position, etc. The k-regret cost of a request is equal to $\sum_{q=2}^k \Delta f_r^q - \Delta f_r^1$. For each request, its k-regret cost is computed, and then the request with the lowest value is inserted. The operator repeats this procedure until all the requests are inserted or until there is no more feasible insertion. In this work, we used three different k-regret operators with $K = 2, 3, 4$.

During the resolution, these insertion operators test the feasibility of millions of insertions. As the feasibility testing consumes most of the computation time, operators must be carefully implemented for best performance.

5.4.2.3 K-regret insertion implementation

The k-regret operator can be associated with a matrix describing the insertion cost of every request in every route. We call this matrix the k-regret matrix. In this matrix, each row corresponds to a request and each column corresponds to a route. Thus, each cell contains the insertion cost of a request in a route at its best pickup and drop-off positions. As illustrated in Figure 5.2, at the intersection of the row corresponding to the request r_4 and the column corresponding to the route k_2 , the blue cell contains the insertion cost (Δ_{r_4, k_2}) of request r_4 in route k_2 . To facilitate the explanations, the row r corresponds to request $r \in \mathcal{R}$ and the column k corresponds to the route served by the vehicle $k \in \mathcal{K}$.

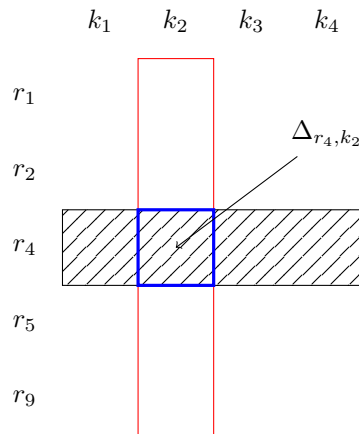


Figure 5.2: column to update after an insertion

After each insertion, the matrix must be updated. Optimizing the update procedure is paramount to speed up the research, for testing the insertions is the most time-consuming part of the algorithm.

Instead of recomputing the whole matrix, the matrix update optimization selects only the cells for which the insertion cost is not up to date.

In the case of single-period DARP, this selection is straightforward. This situation is illustrated in Figure 5.2. When request r is inserted into route k (blue cell), route k is modified. Thus, any insertion cost related to route k must be recomputed. In this optimization process, only the cells in the column corresponding to the route k are updated (red column). In addition, the row associated with the inserted request is deleted (hatched row).

In the case of multi-period DARP, this selection is more complicated. First, there is a k-regret matrix for the morning and afternoon routing. Second, a morning insertion can change the feasibility of an afternoon insertion and vice versa. To understand this mechanism, we present the following example illustrated in Figure 5.3. Let us assume that request $r_2 \in \mathcal{R}$ is inserted in route $k_2 \in \mathcal{K}$ (blue cell). Additionally, let us assume that this insertion changes the ride time of requests $r_3 \in \mathcal{R}$ and $r_4 \in \mathcal{R}$ also inserted in route k_2 . The selection of cells to update is done in three phases:

1. Update of column k_2 : as in single-period DARP, any insertion related to route k_2 must be recomputed. Thus, the column k_2 is recomputed (red column in the morning matrix).
2. Related rows update: with the previous insertion, the ride times of requests inserted in route k_2 may have been modified. Therefore, any insertion related to a ride time modification must be recomputed. For example, let us assume the pair request r'_3 is not inserted. If the ride time of request r_3 increases (respectively decreases), then the remaining ride time for request r'_3 might decrease (respectively increase). Any insertion of request r'_3 must therefore be recomputed to take into account the new remaining ride time. In this example, the row r'_3 must be recomputed (red row in the afternoon matrix).
3. Related columns update: with the previous insertion, the ride times of requests inserted in the route k_2 may have been modified. Hence, if a request is inserted in route k_2 and its pair request is inserted in another route, then the other route must be recomputed. For example, let us assume that the afternoon request r'_4 is inserted in route $k'_1 \in \mathcal{K}$. If the ride time of pair request r_4 increases (respectively decreases), then the remaining ride time for request r'_4 might decrease (respectively increase). As a consequence, any insertion in route k'_1 must be recomputed to take into account the new remaining ride time of request r'_4 . In this example, the column k'_1 must be recomputed (red column in the afternoon matrix).

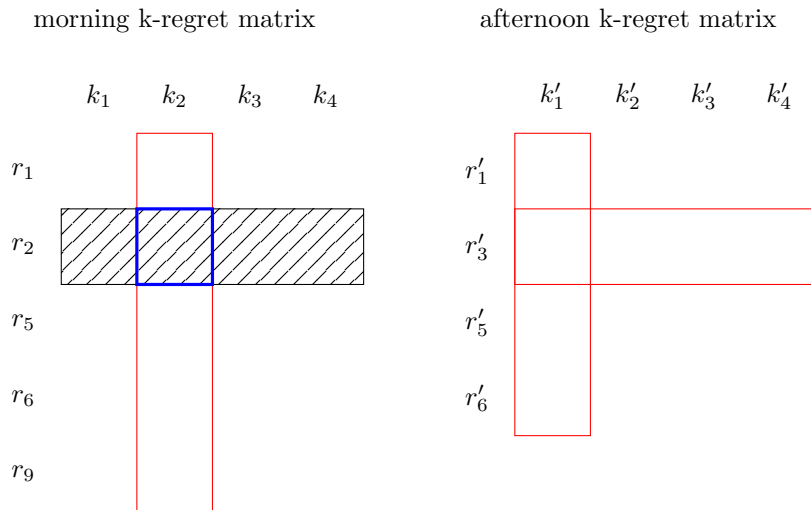


Figure 5.3: k-regret matrix

5.4.3 Precedence filter

In this section, a node n is represented by a tuple (uhs) , with a user $u \in \mathcal{U}$, a period (half-day) $h \in (AM, PM)$, and a site for pickup or drop-off $s \in (P, D)$. Similarly, a request $r \in \mathcal{R}$ is represented by a tuple (uh) . If h denotes the morning period, then h' denotes the afternoon period, and vice-versa. Additionally, a dynamic time window is defined as $[a_{uhs}, b_{uhs}]$ and a dynamic maximum ride time for a request as $\overline{l_{uh}}$.

In this section, several definitions are defined in Section 5.4.3.1. These definitions are then used in Section 5.4.3.2 to describe the precedence relations for the Com-DARP. Finally, the precedence filter implementation is detailed in Section 5.4.3.3.

5.4.3.1 Definitions for the precedence filter

The new notations allow us to formulate the precedence relation between two nodes, the dynamic maximum ride time, and the time windows tightening.

Precedence between two nodes Let us consider two users u and w at sites s and r respectively. For a given period h these users are represented by the nodes (uhs) and (whr) . The node (whr) is a predecessor of (uhs) if it is not possible to leave (uhs) at its earliest and arrive in time at (whr) . This property is used in Malheiros et al. (2021) to limit the number of tested insertions. Mathematically, the precedence between two nodes is defined by the following equation:

$$a_{uhs} + S_{uhs} + T_{uhs,whr} > b_{whr} \quad (5.18)$$

Maximum ride time of a request The maximum ride time of a request depends on its pair request ride time. It can take its initial value or the user's remaining maximum ride time. This is mathematically defined by the following inequality and illustrated in Figure 5.4:

$$\overline{l_{uh}} = \min\{\overline{L_{uh}}; \overline{L_u} - l_{uh'}\} \quad (5.19)$$

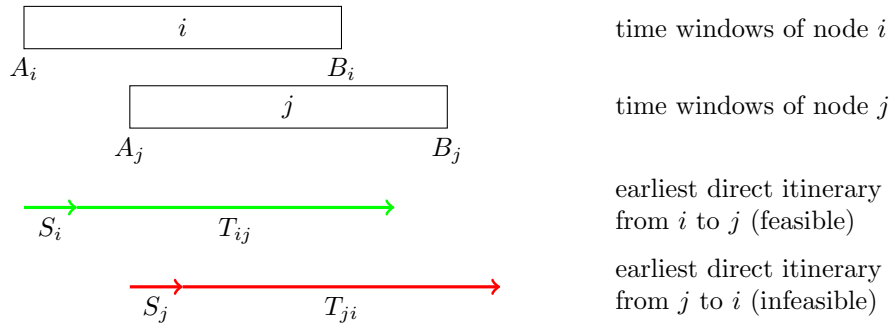


Figure 5.4: precedence relation

Time window tightening The time windows are tightened according to the tightening procedure introduced in Cordeau (2006). The time window tightening results in the following inequalities:

$$A_{uhP} \geq A_{uhD} - S_{uhP} - \overline{L_{uh}} \quad (5.20)$$

$$B_{uhP} \leq B_{uhD} - S_{uhP} - T_{uhP,uhD} \quad (5.21)$$

$$A_{uhD} \geq A_{uhP} + S_{uhP} + T_{uhP,uhD} \quad (5.22)$$

$$B_{uhD} \leq B_{uhP} + S_{uhP} + \overline{L_{uh}} \quad (5.23)$$

In the Com-DARP, another tightening can be performed according to the dynamic maximum ride time of a request. As defined by equation (5.19), $\overline{l_{uh}} = \min\{\overline{L_{uh}}; \overline{L_u} - l_{uh'}\}$. This dynamic tightening impacts the start of the time window at pickup and the end of the time window at drop-off. This is defined by the following equations:

$$a_{uhP} = \max\{A_{uhP}; A_{uhD} - S_{uhP} - \overline{l_{uh}}\}, \quad (5.24)$$

$$b_{uhP} = B_{uhP}, \quad (5.25)$$

$$a_{uhD} = A_{uhD}, \quad (5.26)$$

$$b_{uhD} = \min\{B_{uhD}; B_{uhP} + S_{uhP} + \overline{l_{uh}}\}, \quad (5.27)$$

5.4.3.2 Precedences for the Com-DARP

Given the four different types of time window tightening, the precedence relations $a_{uhs} + S_{uhs} + T_{uhs,whr} > b_{whr}$ can be developed for four pairs of nodes: pickup/pickup, drop-off/drop-off, pickup/drop-off, drop-off/pickup. The results are summarized here. The complete demonstration is detailed in Appendix A.3.

Pickup/pickup

- the node (whP) is always a predecessor of (uhP) if:

$$0 < A_{uhP} + S_{uhP} + T_{uhP,whP} - B_{whP}$$

- the node (whP) is never a predecessor of (uhP) if:

$$0 \geq A_{uhD} - T_{uhP,uhD} + T_{uhP,whP} - B_{whP}$$

- the node (whP) becomes a predecessor of (uhP) when:

$$l_{uh'} > B_{whP} - A_{uhD} + \overline{L_u} - T_{uhP,whP}$$

Drop-off/drop-off

- the node (whD) is always a predecessor of (uhD) if:

$$0 < A_{uhD} + S_{uhD} + T_{uhD,whD} - B_{whD}$$

- the node (whD) is never a predecessor of (uhD) if:

$$0 \geq A_{uhD} + S_{uhD} + T_{uhD,whD} - B_{whP} - S_{whP} - T_{whP,whD}$$

- the node (whD) becomes a predecessor of (uhD) if:

$$l_{wh'} > \overline{L_w} + B_{whP} + S_{whP} - A_{uhD} - S_{uhD} - T_{uhD,whD}$$

Pickup/drop-off

- the node (whD) is always a predecessor of (uhP) if:

$$0 < A_{uhP} + S_{uhP} + T_{uhP,whD} - B_{whD}$$

- the node (whD) is never a predecessor of (uhP) if:

$$0 \geq A_{uhD} - T_{uhP,uhD} + T_{uhP,whD} - B_{whP} - S_{whP} - T_{whP,whD}$$

- the node (whD) becomes a predecessor of (uhP) if one of the three cases is valid:

$$\begin{cases} l_{uh'} > \overline{L_u} - A_{uhD} - T_{uhP,whD} + B_{whD} \\ l_{wh'} > \overline{L_w} + B_{whP} + S_{whP} - A_{uhP} - S_{uhP} - T_{uhP,whD} \\ l_{uh'} + l_{wh'} > \overline{L_u} + \overline{L_w} - A_{uhD} - T_{uhP,whD} + B_{whP} + S_{whP} \end{cases}$$

Drop-off/pickup

- the node (whP) is always a predecessor of (uhD) if:

$$0 < A_{uhD} + S_{uhD} + T_{uhD,whP} - B_{whP}$$

- the node (whP) is never a predecessor of (uhD) if:

$$0 \geq A_{uhD} + S_{uhD} + T_{uhD,whP} - B_{whP}$$

5.4.3.3 Precedence filter implementation

We present some guidelines to implement the precedence filter. When testing the insertion of a request in a route, all the possible positions for pickup and drop-off are tested. The precedence filter allows us to limit the number of positions to test by finding a minimal and maximal feasible position for both pickup and drop-off.

First, the minimal position for the pickup is computed. The route is traversed in reverse order and for each node, the precedence filter tests if the current node is a predecessor of the pickup. If true, the minimal position for the pickup is set to the position just after the position of the current node. Traversing the route in reversed order allows us to stop as soon as a precedence has been found, for the minimal position for the pickup is just after the latest predecessor in the route.

The maximal position for the drop-off is then computed. The route is traversed in order, and for each node the precedence filter tests if the drop-off is a predecessor of the current node. The maximal position for the drop-off is set to the position just before the first successor. At that point, if the minimal position for the pickup is greater than the maximal position for the drop-off, then the insertion is infeasible and the test stops.

Next, the same procedure is applied for the maximal position for the pickup and for the minimal position for the drop-off. To limit the number of tests, the precedence filter tests only the positions between the minimal position for the pickup and the maximal position for the drop-off.

To conclude, the precedence filter allows us to sort out any infeasible positions for the pickup and the drop-off when testing the insertion of a request in a route. This is based on the precedence relations between nodes presented in Section 5.4.3.2.

5.4.4 Feasibility testing

The feasibility of each candidate insertion must be tested. As a large number of insertions are performed while solving the problem, the feasibility test efficiency directly impacts the speed of the search. The tests must therefore be as efficient as possible. The order in which the different tests are performed is paramount. When testing the insertion of a request in a route, we first use the precedence filter to eliminate as many infeasible positions as possible. Then, for each position we first execute the preliminary test and finally the schedule test.

First, the precedence test ensures that the precedence constraints are respected. When inserting a node i in a route, we check that i is not a predecessor of any previous nodes and that i is not a successor of any following node. To speed up this test, the precedence data are preprocessed at the beginning of the resolution with a $O(n^2)$ complexity, with n being the number of nodes. In practice, we pre-compute thresholds as described in Section 5.4.3 for each pair of nodes. Then, during the resolution, we compare the ride times with these thresholds to check the precedence without any additional computation. This comparison is performed in constant time thanks to the preprocessing. Overall, testing the precedence relations for the insertion of one request in one route has a $O(n)$ complexity, with n being the size of the route.

Second, the capacity of the vehicle and some requirements on the time windows are checked in the preliminary tests. These tests are required to be valid in order to have a feasible solution. However, they are not sufficient to ensure the feasibility of a solution. They have a $O(1)$ complexity. These tests are summarized in Braekers et al. (2014);

Finally, the schedule test ensures that there is a feasible schedule for the route. In other words, it checks that the itinerary respects all the time windows and the maximum ride times. As checking the

route schedule is a major challenge in solving the DARP, many papers have contributed to improving this test. In this work, we use the procedure introduced in Tang et al. (2010) as presented in Gschwind (2019). However, this procedure only checks the feasibility of the route. To include the route duration minimization, we add the forward slack time computation introduced by Cordeau and Laporte (2003). The complete algorithm is presented in Tellez et al. (2022). This test has a $O(n^2)$ complexity. In the case of Com-DARP, the reader must keep in mind that a request’s maximum ride time changes dynamically according to the insertion of its pair request.

5.5 Computational experiments

In the following experiments, we compare our metaheuristic to other metaheuristics for solving the Com-DARP problem. Our metaheuristic is called Small and Large Neighborhood Search with Precedence Filter (SLNS-PF). The two other metaheuristics are the classic LNS and the SLNS. These acronyms are used throughout the rest of this section. The algorithms are coded in C++ and compiled into a 64-bit single-thread code using g++ 11.3.0. The algorithm has been adapted to solve both the classic DARP and the Com-DARP. The experiments are performed on a 2.0 GHz AMD EPYC 7702 CPU under a Linux Ubuntu 22.04.1 LTS operating system. For each instance, 5 independent runs are performed.

Concerning the parameter setting, we use the setting from Tellez et al. (2018) for the LNS and the setting from Dumez et al. (2021b) for the SLNS. The parameter setting is presented in Table 5.1.

| value | description |
|--|---|
| $\chi = 5\%$ | record-to-record acceptance percentage |
| $[\delta, \Delta] = [0.1, 0.45]$ | destruction size for the LNS |
| $[\delta_{small}, \Delta_{small}] = [0.01, 0.1]$ | destruction size for a small iteration of the SLNS |
| $[\delta_{large}, \Delta_{large}] = [0.1, 0.3]$ | destruction size for the large iteration of the SLNS |
| $\omega = 10 N ^{1.5}$ | small iterations limit for the SLNS |
| $\psi = 6$ | roulette wheel parameter for the history removal operator |

Table 5.1: parameter setting for the metaheuristics

The results are presented as follows. First, the benchmark instances are described in Section 5.5.1. Then, the results on the Com-DARP instances are analyzed in Section 5.5.2. Finally, we discuss the managerial insights in Section 5.5.3.

5.5.1 Benchmark instances

To our knowledge, there is no benchmark for our problem in the literature. We have therefore generated instances using the following method. A user has two requests: one morning request and one afternoon request. The pickup and drop-off locations are randomly distributed on a $[-10, 10] \times [-10, 10]$ kilometers grid. Distances and durations are equal to the Euclidean distance between two nodes. For a user, the morning pickup location corresponds to the afternoon drop-off location and the morning drop-off location corresponds to the afternoon pickup location (commuting). The morning and the afternoon loads are both equal to 1 and the service duration is always equal to 3 minutes at pickup and drop-off. The quality of service is ensured by the maximum ride time coefficient (MRTC), for a user’s maximum ride time is set to their direct ride times multiplied by the MRTC. In this work, we used $MRTC = 1.1, 1.2, 1.3, 1.4, 1.5$. Small coefficients induce a better quality of service but also reduce the number of feasible solutions. A request maximum ride time is set to the user’s maximum ride time minus the pair request direct ride time. Additionally, we raise to 15 minutes any maximum ride time lower than this value. The time windows for morning drop-off and afternoon pickup nodes are set first. The maximum time window for the morning drop-off is randomly selected according to a normal distribution centered on 8am with a variance of 90 minutes. Similarly, the minimum time window for the afternoon pickup is randomly selected according to a normal distribution centered on 4pm with a variance of 90 minutes. Time windows are open for 10 minutes. The morning pickup and the afternoon drop-off time windows are set according to the time

window tightening procedure described in Cordeau (2006). There is one depot located in the center of the map. The service time is equal to zero minutes at the depot. A fleet of infinite homogeneous vehicles leaves and returns to this depot. Each vehicle has a capacity of 6 passengers, costs 0.25€ per kilometer to run, costs 0.5€ per minute to run, and has a fixed cost of 50€ if used during the day. These costs correspond to the real transportation cost of sanitary transportation in France. Because our work is motivated by real problems, our instances are fairly large compared to the literature. In total, we generated 24 instances: 8 instances with 100 users, 8 instances with 200 users, and 8 instances with 400 users.

The computational time limit is set to 5 minutes for instances with 100 users, 20 minutes for instances with 200 users, and 60 minutes for instances with 400 users. These time limits include pre-computed operations and the optimization part.

5.5.2 Performance evaluation

The performances of the three metaheuristics are compared using the following metrics: the average gap, the average number of iterations, and the average computation duration in seconds to reach a gap of 5% relative to the best-known solution. We chose to measure the computation duration to reach a 5% gap because, due to the randomness of the search, some runs might not have been able to achieve a better gap. As the 5% gap threshold is not too easy nor too hard to reach, it is sufficient to give us a good insight into the convergence speed of the different algorithms. This 5% gap objective can also be seen as a good objective for real applications. The results are summarized in Table 5.2. The computational time to reach the 5% gap is not reported in the table when the gap is not reached for the five runs within the computation time limit (5 minutes for 100 users, 20 minutes for 200 users, and 60 minutes for 400 users).

| nb users | MRTC | LNS | | | SLNS | | | SLNS-PF | | |
|----------|------|------|-------|-----|------|--------|----|---------|--------|----|
| | | gap | nb it | 5% | gap | nb it | 5% | gap | nb it | 5% |
| 100 | 1.1 | 1.37 | 43131 | 1 | 0.45 | 199153 | 4 | 0.44 | 256700 | 3 |
| | 1.2 | 2.23 | 40385 | 3 | 0.81 | 196763 | 4 | 0.55 | 251645 | 3 |
| | 1.3 | 2.73 | 37875 | 7 | 0.76 | 202018 | 5 | 0.70 | 258013 | 4 |
| | 1.4 | 3.25 | 35582 | 19 | 1.04 | 197766 | 7 | 0.97 | 256135 | 5 |
| | 1.5 | 3.52 | 33670 | - | 1.03 | 194112 | 7 | 0.81 | 263476 | 4 |
| 200 | 1.1 | 2.15 | 51779 | 4 | 0.63 | 212108 | 13 | 0.44 | 327917 | 9 |
| | 1.2 | 2.52 | 45998 | 11 | 0.49 | 209591 | 15 | 0.45 | 325856 | 12 |
| | 1.3 | 3.57 | 42273 | 84 | 0.80 | 191089 | 19 | 0.53 | 346985 | 10 |
| | 1.4 | 4.03 | 39067 | - | 0.98 | 181717 | 22 | 0.72 | 322177 | 11 |
| | 1.5 | 4.22 | 36313 | - | 0.83 | 173603 | 22 | 0.73 | 315963 | 13 |
| 400 | 1.1 | 2.31 | 23632 | 24 | 0.59 | 149195 | 37 | 0.46 | 296121 | 29 |
| | 1.2 | 3.42 | 20595 | 123 | 0.65 | 127606 | 56 | 0.40 | 258638 | 32 |
| | 1.3 | 4.27 | 18542 | - | 0.84 | 129610 | 59 | 0.50 | 247424 | 34 |
| | 1.4 | 5.07 | 16978 | - | 0.95 | 119089 | 43 | 0.64 | 268687 | 28 |
| | 1.5 | 5.37 | 15644 | - | 0.75 | 115205 | 63 | 0.56 | 251023 | 33 |
| average | | 3.34 | 33431 | - | 0.77 | 173242 | 25 | 0.59 | 283117 | 15 |

Table 5.2: results on the Com-DARP benchmark

First, the SLNS-PF obtains the best results across all the instances. On average, the SLNS improves the gap from the LNS by 2.57% ($= 3.34 - 0.77$). The precedence filter component further improves this gap by an additional 0.18% ($= 0.77 - 0.59$). At best, the difference gap between LNS and SLNS-PF reaches 4.78% for instances with 400 users and a maximum ride time coefficient of 1.5.

Secondly, adding the precedence filter to the SLNS allows us to perform on average 63% ($= \frac{283117 - 173242}{173242}$) more iterations and up to 126% more iterations for instances with 400 users and a

maximum ride time coefficient of 1.4. These additional iterations give an edge to the SLNS-PF for finding better solutions faster than the other metaheuristics.

Thirdly, while the precedence filter does not provide as much gap improvement as the SLNS does, it allows good solutions to be obtained more quickly. In other words, solutions of similar quality are reached but the SLNS-PF reaches them faster than the SLNS. This is the consequence of performing more iterations when the precedence filter is activated. On average, solutions with a 5% gap are obtained 39% ($= \frac{25-15}{15}$) sooner with the precedence filter and up to 48% sooner for instances with 400 users and a maximum ride time coefficient of 1.5. This convergence speed is illustrated in Figure 5.5. On one instance with 200 users and a 1.5 maximum ride time coefficient, the best solution for each metaheuristic is represented by its convergence. As suggested by the results in Table 5.2, the SLNS and the SLNS-PF outperform the LNS. In addition, while both the SLNS and the SLNS-PF reach a good solution, the SLNS-PF is much faster.

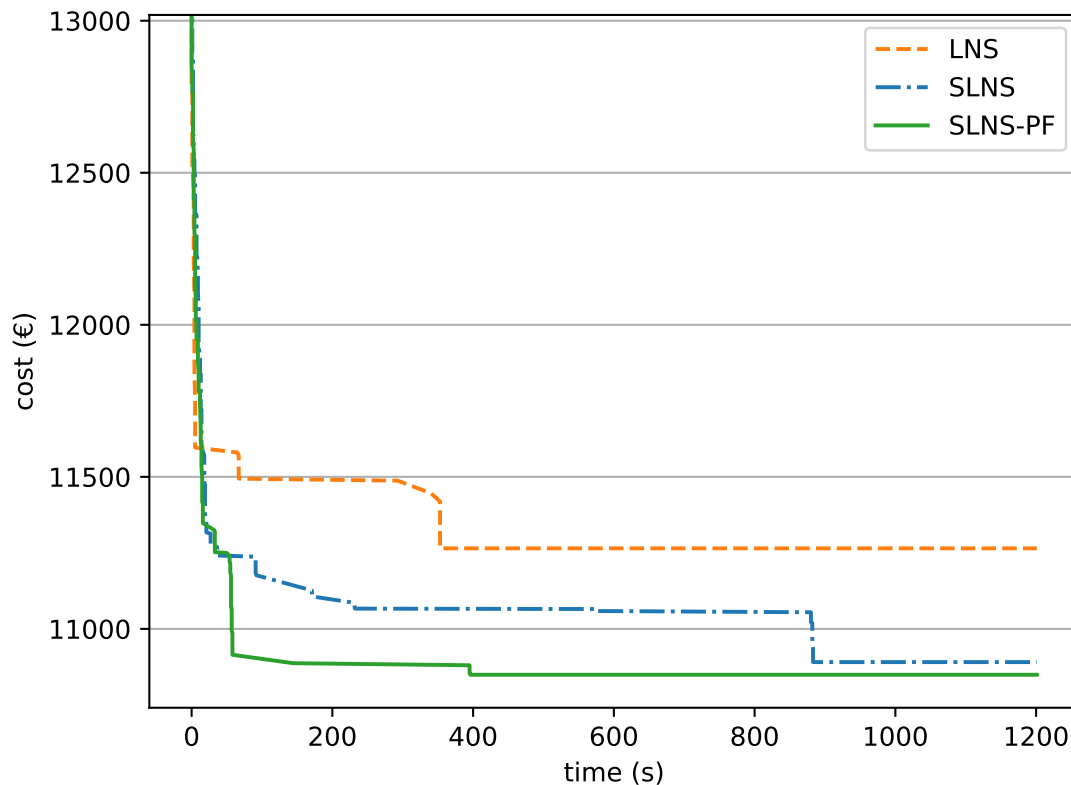


Figure 5.5: convergence speed of the LNS, SLNS, and SLNS-PF (200 users, MRTC = 1.5)

5.5.3 Managerial insights

With the proof that the SLNS-PF works on the Com-DARP, we ask the following question: is it useful? The Com-DARP can be divided into two independent DARPs if each user's daily maximum ride time is shared equally between its morning and its afternoon request. Thus, combining the routing from the two independent DARPs gives a valid solution for the Com-DARP. If the Com-DARP is useful, then it should provide solutions with better costs compared to solutions from the two corresponding DARPs.

To test this, we build a benchmark with pairs of DARPs that corresponds to the Com-DARP benchmark. As previously described, maximum daily ride times are shared equally between morning and

afternoon requests. The results are presented in Table 5.3. This table compares the results between the two independent DARPs (2-DARP) and the Com-DARP. Columns “RT (min)” give the average daily ride time (RT) in minutes, columns “cost (€)” give the cost of the solution in euros. The comparison, Com-DARP relative to 2-DARP, is made in columns “cost (%€)” and “RT (Δ min)”. The first one corresponds to the cost gap in percentage and the second corresponds to the ride time difference in minute. To illustrate the results, we plot the best cost relative to the maximum ride time coefficient for each method for the instances with 100 users (see Figure 5.6).

| nb users | MRTC | 2-DARP | | Com-DARP | | gap | |
|----------|------|----------|----------|----------|----------|-----------|--------------------|
| | | cost (€) | RT (min) | cost (€) | RT (min) | cost (%€) | RT (Δ min) |
| 100 | 1.1 | 6 905 | 46.39 | 6 757 | 47.37 | -2.14 | 0.98 |
| | 1.2 | 6 713 | 47.36 | 6 402 | 49.47 | -4.62 | 2.11 |
| | 1.3 | 6 394 | 49.06 | 6 129 | 52.67 | -4.14 | 3.61 |
| | 1.4 | 6 198 | 50.57 | 5 830 | 55.25 | -5.94 | 4.69 |
| | 1.5 | 5 967 | 53.41 | 5 688 | 58.04 | -4.68 | 4.63 |
| 200 | 1.1 | 12 995 | 47.01 | 12 601 | 48.16 | -3.04 | 1.16 |
| | 1.2 | 12 449 | 48.25 | 11 875 | 50.93 | -4.61 | 2.69 |
| | 1.3 | 11 693 | 50.77 | 11 329 | 53.77 | -3.11 | 3.01 |
| | 1.4 | 11 320 | 52.74 | 10 880 | 56.48 | -3.89 | 3.74 |
| | 1.5 | 10 917 | 55.37 | 10 660 | 59.01 | -2.36 | 3.63 |
| 400 | 1.1 | 25 117 | 47.25 | 24 188 | 48.39 | -3.70 | 1.15 |
| | 1.2 | 23 686 | 48.81 | 22 213 | 51.58 | -6.22 | 2.77 |
| | 1.3 | 21 886 | 51.74 | 21 158 | 54.51 | -3.33 | 2.77 |
| | 1.4 | 20 765 | 54.11 | 20 308 | 57.34 | -2.20 | 3.22 |
| | 1.5 | 19 993 | 56.55 | 19 739 | 60.48 | -1.27 | 3.93 |
| average | | 13 533 | 50.62 | 13 051 | 53.56 | -3.68 | 2.94 |

Table 5.3: comparison between the Com-DARP and the corresponding 2-DARP

First of all, the Com-DARP always gives results with better costs. As depicted in Figure 5.6, the Com-DARP solutions always dominate the 2-DARP ones. With a gain of 3.68% on average and 6.22% at best (400 users, MRTC = 1.2), savings are possible just by grouping together the morning and afternoon DARPs into one daily Com-DARP. Second, the financial gains result in a slight increase in the users’ average ride time. Indeed, users spend an average of 2.94 extra minutes in transport for the Com-DARP. As the daily ride time is lengthened by 2.94 minutes, the morning and evening trips would be lengthened by less than 1.5 minutes, which is almost imperceptible for the user. Even in the worst case, daily ride times are lengthened by 4.69 minutes on average (100 users, MRTC = 1.4). Considering that daily ride times amount to more than 50 minutes on average, this increase seems acceptable.

These previous observations give the following possibilities for decision-makers. On one hand, it is possible to reduce the transportation cost for the same maximum ride time coefficient. With this option, users’ trips are lengthened by less than 1.5 minutes on average. In other words, the Com-DARP gives solutions with better costs and maintains the quality of service related to the maximum ride time coefficient, which comes at the expense of a small increase in the users’ effective ride times. It is up to decision-makers to assess if the gains are worth a small extra ride time. On the other hand, it is possible to reduce the maximum ride time coefficient for similar transportation costs. As the quality of service is linked with the maximum ride time coefficient, its decrease would improve this aspect of the quality of service and reduce inequalities between users. For example, Figure 5.6 shows that the Com-DARP with MRTC = 1.4 gives a solution with a better cost than the 2-DARP with MRTC = 1.5. While having a better cost, the Com-DARP solution ensures that users will not have detours that exceed 40% of their direct ride time. On the opposite, the 2-DARP solution is more expensive and users can have detours up

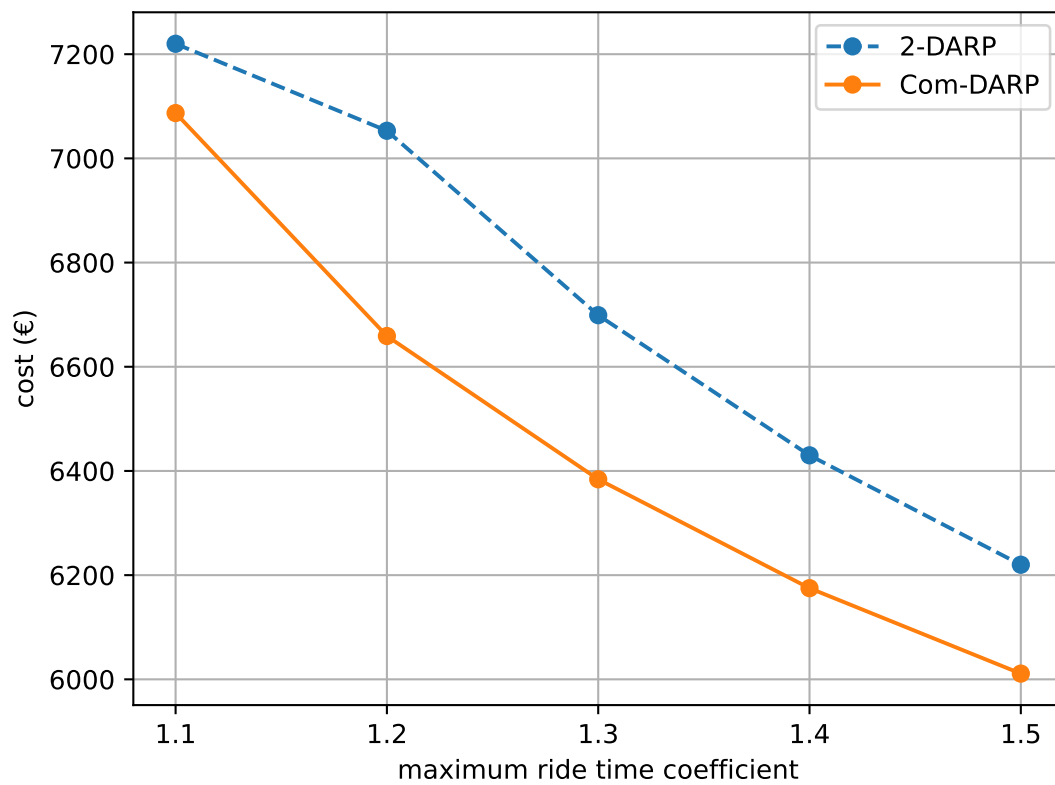


Figure 5.6: best cost relative to the maximum ride time coefficient for instances with 100 users

to 50% of their direct ride time. According to Table 5.3, the users' average ride time is longer by 1.84 (= 55.25 – 53.41) minutes for the Com-DARP. A small global increase in ride times would therefore allow to limit unfair situations for some users. This tendency is observed across instances and across maximum ride time coefficients.

5.6 Conclusion

In this chapter, we introduced the Commuting Dial-A-Ride Problem (Com-DARP) and a specific metaheuristic to solve it. In this new variant of the Dial-A-Ride Problem (DARP), a user has two requests, one in the morning and the other in the afternoon, each with its respective maximum ride time. In addition, each user has a maximum daily ride time. This feature was extracted from the door-to-door transportation of disabled children in France. However, the model and the solution method can be applied to any transportation system with commuting. We transposed the Small and Large Neighborhood Search (SLNS) metaheuristic from the Vehicle Routing Problem (VRP) to the Com-DARP. In addition, we developed a new feasibility testing procedure that takes advantage of the precedence constraints and adapts dynamically during the search. The complete algorithm is called SLNS-PF. The experiments on generated Com-DARP instances from 100 to 400 users showed a 2.75% gap improvement from the SLNS-PF on average and up to 4.78%. In addition, the introduced precedence test increases the computation speed. Under 5% gap solutions are obtained 39% sooner on average and up to 48%. From a managerial point of view, solving the Com-DARP instead of the two corresponding morning and afternoon DARPs has three advantages. First, the cost is reduced by 3.68% on average. Second, it is also possible to lower the limit on the users' maximum ride time without a significant change in the transportation cost. Finally, solutions are obtained faster.

This work opens two main perspectives for future research. First, more features could be integrated into this model such as heterogeneous users and reconfigurable vehicles, in order to get closer to real transportation systems. Second, it would be interesting to study the multi-request multi-trip multi-depot heterogeneous fleet DARP introduced in this work as a generalization of the Com-DARP.

Chapter 6

The Assignment Dial-a-Ride Problem (ADARP)¹

Highlights:

- This chapter presents the Assignment Dial-A-Ride Problem (ADARP).
- We propose a solution method based on a metaheuristic combining a large neighborhood search and an exact method.
- A comparison with the literature and case studies on real instances in the healthcare context is provided.
- In terms of managerial insight, this new approach reduces the current cost up to 24% without degrading the service quality for users.

6.1 Introduction

In France, the medico-social sector is structured by a set of institutions and services that support people with disabilities and other vulnerabilities. Every day, each user is picked up at home and taken to their assigned medico-social institution (MSI) where they receive personal care throughout the day. At the end of the day, users are taken from their respective MSI to their homes. The assignment process is long and labor intensive. Indeed, each demand has particular needs and each MSI has particular services. As a consequence, satisfying all the demands requires to test many different configurations of assignment.

The assignment process is carried out by the MDPH (*Maison Départementale des Personnes Handicapées*), the home for the disabled in each French *département*. A multidisciplinary team (EP) within this institution analyzes the applications of people with disabilities and, where relevant, refers them to a list of institutions and medical-social services (ESMS) adapted to their needs. More precisely, each

¹This chapter is an updated version of the journal article Chane-Haï et al. (2023a) (to be published in Health Care Management Science).

user receives medical care throughout the week. Indeed, there is a care planning for each user that repeats weekly. As a consequence, the EP must check if the MSIs' weekly planning can accommodate the candidate and try to build a valid planning for each weekday. An EP must process by hand over 200 referrals per month. Thus, it must change assignments and plannings accordingly every month. The referrals are processed one by one in order of arrival. As a consequence, the referrals' order have a considerable impact on the overall performance of the system. Using a solver that processes all the demands simultaneously will facilitate this decision process. As a result, the quality of the assignment will be improved and the EP will spend less time on the applications' processing. Based on this analysis, the CDAPH, the commission for the rights and autonomy of the disabled (*Commission des Droits et de l'Autonomie des Personnes Handicapées*) validates or refuses each application. If a candidate is accepted, the CDAPH also informs them about the reasons of acceptance and the care protocol. The person must apply for admission to one or more institutions listed by the CDAPH. This assignment process is more detailed in Mosquera Varela et al. (2017). Because the resources of MSIs are limited and the assignment process is so laborious, previous decisions are not revised even if the context changes. Therefore, the impact of these decisions lasts several years. For example, if a space becomes available in a MSI closer to a patient's home, no steps are taken to review the current assignment. Moreover, the transportation criterion is not explicitly taken into account in the assignment decision.

The daily transportation in the context of MSIs entails several problems. The first is related to the users' quality of life. Because they are in a vulnerable position due to their disabilities, their daily transportation should not take too long. However, a study presented by Tellez et al. (2020) shows the long duration of these users' ride times and the real need to reduce them. This study, carried out in the city of Lyon and surrounding areas, shows that trips are on average 48 kilometers long and last on average 74 minutes. The second issue is that the transportation also has an environmental impact of 1.42 million kg CO₂ eq and 4 million km per year for 245,000 trips. This environmental impact has to be limited. The final issue we address is the economic dimension. In the case of patients under the age of 18, who make up the majority here, the cost of this transportation system is funded directly by the MSIs. In 2017, medico-social transportation costs totaled 500 million euros, at national level, for all medico-social institutions put together (Gonzalez et al., 2017). This cost increased by 40% between 2006 and 2014 Lesteven et al. (2016) and has had a significant impact on their global budget since this expense is the second largest after wages ANAP (2016). It follows that more money spent on transportation means less money spent on supporting people with disabilities and on the MSIs' services.

The assignment process also has an impact on the institutions' resources. Each institution has accreditation for a certain number of places and a predefined number of employees. Each employee has specific competencies. When a user is assigned to the facility, that individual uses one space allocated in the accreditation and also requires time from several appropriate caregivers. If this time is greater than the available resource, the excess will be counted as overtime. In practice, overtimes last until the next assignment at least or until additional staff is recruited. Additionally, temporary workers can be recruited for overtime.

This work proposes a decision-making approach that integrates three aspects of the problem. The first aspect is the assignment of people with disabilities to a given MSI. The second aspect is the staff sizing at MSIs to be able to satisfy the needs of their users. The last aspect is the management of the shared door-to-door transportation service. This approach makes it possible to jointly optimize the proper use of resources and logistics costs. This is called Assignment Dial-A-Ride Problem (ADARP) in the rest of the chapter. If used by the EP, some users may need to change MSI. However, this change would occur only once a month. Moreover, a monthly reassignment is also a chance for them to move to a closer MSI and thus, reduce their commuting time. For the sake of clarity, we focus our study only on the transportation from home to MSIs for one day. However, the results can easily be transposed to the reverse - transportation from MSIs to home - by reversing the routes in the solution. Combining the morning and afternoon problems would be ideal but is still challenging as the size of the problem would be multiplied by two. As a result, the computation duration would increase exponentially due to the NP-hardness of the DARP. To build the weekly planning, the problem must be solved for each weekday. Aspects related to route consistency are not considered in this study but constitute interesting research perspectives. We refer the reader to Braekers and Kovacs (2016), Tellez et al. (2022), and Lindstrøm and Røpke (2022) for respectively, drivers, service times, and groups consistency.

The chapter is structured as follows: Section 6.2 reviews the literature on related problems and presents our main contributions. Section 6.3 defines the problem with its route-based formulation. The method developed to solve the ADARP is detailed in Section 6.4 and the case study is presented in Section 6.5. In Section 6.6, the parameters of the method are tuned, the components of the method are validated, and the method is challenged with results from the literature. Section 6.7 then presents managerial results derived from numerical analyses of the case study. In Section 6.8, we finally summarize our main findings and indicate some interesting extensions of the model presented here.

6.2 Literature review

Despite having no added value to the care itself, transportation represents an essential part of healthcare systems. Indeed, transportation is not a medical act. In that regard, this support activity must be as efficient as possible to improve the quality of service with a minimal cost. As described in Tellez (2019), healthcare transportation is a complex problem due to its multiple aspects. As this domain has been extensively studied in the literature, we provide only a few references in order to be concise. The first aspect of medical transportation is the nature of what is transported. It can be patients (e.g., Qu and Bard (2013), Tellez et al. (2018)), medical staff (e.g., Genet et al. (2011), Mankowska et al. (2014), Malagodi et al. (2021)) or medical supplies (e.g., Kelle et al. (2012), Uthayakumar and Priyan (2013)). In our study, we focus on the patient transportation. Second, it is possible to distinguish emergency transportation (e.g., Coster et al. (2017), Li et al. (2019)), where the patient's life is directly in danger, from the non-emergency transportation (e.g., Mahon (2011), Huggins and Shugg (2008), Fogue et al. (2016)). In our case, we study non-emergency transportation. Finally, the transportation can be occasional (e.g., Qu and Bard (2013), Qu and Bard (2015)) or regular (e.g., Feillet et al. (2014), Tellez et al. (2022)). Our problem is about regular transportation. To summarize, the assignment dial-a-ride problem addresses the regular non-urgent patient transportation. As shown in Hains et al. (2011), the regular non-urgent transportation is growing with the global increase in life expectancy. As a consequence, long-term care has received increasing attention (e.g., Worrall and Chaussalet (2015)). In order to improve the efficiency of healthcare systems in the long run, we must be able to revise previous decisions that are no more relevant.

Motivated by a para-transit application, many variants of the Dial-A-Ride Problem (DARP), as defined in Cordeau and Laporte (2003), have emerged in the last two decades. The reader may refer to Molenbruch et al. (2017b) and Ho et al. (2018) for recent reviews. Because passengers are vulnerable, several specific constraints or objectives related to the quality of service are considered. For instance, the multi-criteria optimization of the total ride time and the routing cost is considered in Lehuédé et al. (2014), among others. The solution consistency may also be considered during the week. Some passengers prefer to have the same driver every day Braekers and Kovacs (2016), or to have the same pickup hours every day Tellez et al. (2022). These two papers study the impact of consistency on the total cost of the solution with a bi-objective analysis. Because some users are in wheelchairs, Parragh (2011) introduces the impact of heterogeneous users and vehicles. In the same way, Qu and Bard (2013) study the possibility to configure the number of seats and wheelchairs in a vehicle before the departure, and Tellez et al. (2018) study the same possibility but during the route. Other extensions of the DARP consider the transfer of passengers from one vehicle to another Masson et al. (2014) or the multidepot-DARP Braekers et al. (2014) and Malheiros et al. (2021).

The possibility to choose between several destinations in a Vehicle Routing Problem (VRP) is introduced by Ghiani and Improta (2000). In this paper, the authors present the Generalized-VRP (GVRP) which is a VRP where a client demand can be fulfilled by visiting one point in a list of alternatives. By adding time window constraints, Moccia et al. (2012) extend the previous model and introduce the GVRPTW. In the context of urban logistics, Dumez et al. (2021a) and Dumez et al. (2021b) have extended the GVRPTW to the VRP-with Delivery Options (VRPDO). In the latter, a delivery can be made in certain lockers selected by the client, but the capacities of the lockers are finite and have to be taken into account for the delivery point selections.

In the context of dial-a-ride transportation, the Generalized Dial-A-Ride Problem (GDARP) presented by Pinson et al. (2016) was the first paper to consider the possibility of alternative pickup and drop-off

points. In that study, each passenger could be picked up at their initial position or could walk for a certain distance to reach an alternative pickup point. The same mechanism existed for alternative drop-off points. The optimization model jointly optimizes the selection of the pickup and drop-off points, and the selection of the routes.

The specificities of assignment problems in a medical context (i.e., long-term decisions, compatibility between needs and competencies, etc.) make it inappropriate to transpose insights from other assignment studies with a different context. This is the reason why this paragraph focuses on other research works that also consider a medical context. The assignment of people with disabilities was studied by Osorio et al. (2017). This study demonstrated the impact of alternative assignment policies on the performance of the healthcare system and on the quality of care. Transport was not taken into account. The study identified a number of points for improvement, including the long processing time due to the lack of computerization, and the failure to take logistical criteria into account in the choice of an assignment. In another work, Mosquera Varela et al. (2017) proposed a decision model for this assignment. In this model, a maximum distance between a patient’s home and institutions was considered, but the management of the transportation service and the staff sizing were not.

With its contributions, this work tries to bridge research gap RG4². First, we introduce the first Assignment Dial-A-Ride Problem (ADARP). This problem stems from a real-life issue faced by medico-social institutions which have to take three interconnected decisions pertaining to: 1) the routing for the ride-sharing services; 2) the assignment of users to geographical points; and 3) the staff sizing at the geographical points. Second, we present a matheuristic that iteratively generates and selects routes. Routes are generated with a large neighborhood search that includes an assignment problem, and are selected using a mixed integer linear program. After being validated on instances from the literature for the GDARP and the VRPDO, the solution method is applied to real instances obtained from the sanitary transportation services in three different areas in France. Finally, we provide managerial insights derived from the numerical experiments. The cost reduction induced by the user (re)assignment is equivalent to the cost reduction induced by the sharing of the vehicles between institutions. Savings can thus be generated without increasing ride times, with less than a quarter of the users reassigned, and with small changes in institutions’ staff.

6.3 Model

The model is presented with its route-based formulation. In this section, we introduce the notations used, the single route problem, and finally the route selection problem coupled with staff sizing. To facilitate its comprehension, the complete list of all the notations used in this chapter is provided in Appendix A.4. In order to better fit the problem, The notation differs from the one used in Section 3.3.1 for the classic DARP.

6.3.1 Notations

Users We consider a set of users \mathcal{U} . A user $u \in \mathcal{U}$ has to be served by the selection and execution of one of their itineraries \mathcal{I}_u . The pickup and drop-off nodes of user $u \in \mathcal{U}$ and itinerary $i \in \mathcal{I}_u$ are denoted $P(u, i)$ and $D(u, i)$ respectively.

Nodes Let $\mathcal{N} = \mathcal{P} \cup \mathcal{D} \cup \mathcal{O}^+ \cup \mathcal{O}^-$ be the set of all the nodes. From a user’s perspective, $\mathcal{P} = \{P(u, i) \mid u \in \mathcal{U}, i \in \mathcal{I}_u\}$ is the set of pickup nodes, $\mathcal{D} = \{D(u, i) \mid u \in \mathcal{U}, i \in \mathcal{I}_u\}$ is the set of drop-off nodes. From a driver’s perspective, $\mathcal{O}^+ = \{o_k^+ \mid k \in \mathcal{K}\}$ is the set of starting depots for the beginning of the routes, and $\mathcal{O}^- = \{o_k^- \mid k \in \mathcal{K}\}$ is the set of ending depots for the end of the routes. Thus, any driver must start a route from a starting depot and finish it at an ending depot. In practice, the starting and ending depots are usually the same. However, we propose a formulation with differentiated starting and ending depots as it easily encompasses our case and others (e.g., drivers directly return home with the service vehicle).

²RG4: There is no model that encompasses the entire scope of the regular non-urgent sanitary transportation of disabled children in France and considers all the aspects related to the different stakeholders.

Times The time and the distance to go from node i to node j are denoted $\Gamma_{i,j}$ and $\Delta_{i,j}$ respectively. If a node $n \in \mathcal{N}$ is visited, the visit takes a service time S_n and the beginning of the service time has to be within a time window $[A_n, B_n]$. Then each user $u \in \mathcal{U}$ has a maximum ride time H_u .

Vehicles Let \mathcal{K} be the set of heterogeneous vehicles. Each vehicle is assigned to a specific depot by the solution. The capacity of vehicle $k \in \mathcal{K}$ is denoted \bar{V}_k . The load variation corresponding to user $u \in \mathcal{U}$ in a vehicle is denoted V_u (it may be more than one in case of obesity or an in-wheelchair user). By extension, we define the load variation for nodes such that $V_{P(u,i)} = -V_{D(u,i)} = V_u$ for all $i \in \mathcal{I}$. The load variation for nodes related to depots is zero. Then, when using vehicle k , the fixed cost, the distance cost, and the time cost are denoted C_k^f , C_k^d , and C^t respectively.

Geography The set of geographical locations is denoted \mathcal{G} , and $G(n) \in \mathcal{G}$ is the geographical location of node $n \in \mathcal{N}$.

Resources A user $u \in \mathcal{U}$ needs a quantity $E_{u,r}$ of resources $r \in \mathcal{R}$ with \mathcal{R} the set of resources. A resource corresponds to the amount of working time available for a particular care from caregivers with the corresponding competency. The availability of resource $r \in \mathcal{R}$ in geographical locality $g \in \mathcal{G}$ without paying for overtime is denoted $\bar{E}_{g,r}$ and the maximal quantity of resources with overtime (with paying) is denoted $\bar{E}_{g,r}^{ext}$. Then, C_r^{ext} is the resource extension cost per unit of resource $r \in \mathcal{R}$.

Route For a given route ω , let $\mathcal{UI}(\omega)$ be the set of tuples (u, i) , where u is the user and $i \in \mathcal{I}_u$ the itinerary chosen for user $u \in \mathcal{U}$ in route ω . The ordered set of visited nodes is denoted $\mathcal{N}(\omega)$, and the first and last visited nodes of route $\omega \in \Omega$ are denoted $F(\omega) \in \mathcal{O}^+$ and $L(\omega) \in \mathcal{O}^-$ respectively. Then, the node visited just before node n in route ω is denoted $Q(\omega, n)$ with $n \in \mathcal{N}(\omega) \setminus \{F(\omega)\}$.

6.3.2 Single-route problem

The single route problem is formulated as follows. Each route needs to comply with these constraints while minimizing its costs.

The decision variables are: $t_{\omega,n}$, the beginning of service time for node n in route ω ; v_n , the load of the vehicle when leaving the node n ; and x_k that indicates whether or not the vehicle k is selected for the route ω .

$$\begin{aligned} \min f_\omega = & \sum_{k \in \mathcal{K}} C_k^f x_k + C^t (t_{\omega, L(\omega)} - t_{\omega, F(\omega)}) \\ & + \sum_{k \in \mathcal{K}} C_k^d x_k \sum_{n \in \mathcal{N}_\omega \setminus \{F(\omega)\}} \Delta_{Q(\omega, n), n} \end{aligned} \quad (6.1)$$

s.t.

$$t_{\omega, Q(\omega, n)} + \Gamma_{Q(\omega, n), n} + S_{Q(\omega, n)} \leq t_{\omega, n} \quad \forall n \in \mathcal{N}(\omega) \setminus \{F(\omega)\} \quad (6.2)$$

$$H_u \geq t_{\omega, D(ui)} - t_{\omega, P(ui)} - S_{P(ui)} \quad \forall ui \in \mathcal{UI}(\omega) \quad (6.3)$$

$$A_n \leq t_{\omega, n} \leq B_n \quad \forall n \in \mathcal{N}(\omega) \quad (6.4)$$

$$t_{\omega, P(ui)} \leq t_{\omega, D(ui)} \quad \forall ui \in \mathcal{UI}(\omega) \quad (6.5)$$

$$v_{Q(\omega, n)} + V_n = v_n \quad \forall n \in \mathcal{N}(\omega) \setminus \{F(\omega)\} \quad (6.6)$$

$$v_n \leq \bar{V}_k x_k \quad \forall n \in \mathcal{N}(\omega), k \in \mathcal{K} \quad (6.7)$$

$$\sum_{k \in \mathcal{K}} x_k = 1 \quad (6.8)$$

$$t_{\omega, n}, v_n \in \mathbb{R}^+ \quad \forall n \in \mathcal{N}(\omega) \quad (6.9)$$

$$x_k \in \{0, 1\} \quad \forall k \in \mathcal{K} \quad (6.10)$$

The objective function (6.1) is a sum of the fixed cost related to the depreciation expense of the selected vehicle, the driver wage that is proportional to the duration of the route, and the distance cost.

In practice, drivers' contracts truly relate to the duration of their routes and are not based on a fixed weekly salary.

Constraints (6.2) to (6.5) ensure respectively that the driving time and the service time are respected between the visits of the nodes, the ride times do not exceed the limit, the time windows are satisfied, and each pickup is before the associated drop-off. Constraints (6.6) to (6.8) ensure respectively that the load increases when leaving pickup nodes and decreases for drop-off nodes, the maximum capacity of the selected vehicle is not exceeded, and one vehicle is selected to serve the route.

6.3.3 Route selection problem and staff sizing

Even though the set of routes that satisfy the single-route problem (6.1)-(6.10) is huge, here we formulate the route selection problem by assuming that it is possible to list them all in a set Ω . Note that all combinations of itineraries have to be considered. A route $\omega \in \Omega$ is characterized by its cost f_ω and its set of tuples $\mathcal{UI}(\omega)$. The latter contains the list of users served by the route and the itineraries chosen for each user served. If two routes have the same set of users and itineraries, then only the cheapest can be kept in Ω and the other can be dropped.

The route selection model is now presented. The model consists in selecting some routes, taking into account the quantity of resources available in each geographical locality. The first decision variable y_ω (6.14) is a Boolean indicating whether or not the route ω is selected. The second decision variable $e_{g,r}$ (6.15) is the quantity of extra resources $r \in \mathcal{R}$ that must be spent to ensure that the needs of each user assigned to geographical locality $g \in \mathcal{G}$ will be satisfied.

$$\min f = \sum_{\omega \in \Omega} f_\omega y_\omega + \sum_{r \in \mathcal{R}} \sum_{g \in \mathcal{G}} C_r^{ext} e_{g,r} \quad (6.11)$$

s.t.

$$\bar{E}_{g,r} + e_{g,r} \geq \sum_{\omega \in \Omega} \sum_{\substack{u,i \in \mathcal{UI}(\omega) \\ g \in \left\{ \begin{array}{l} G(P(u,i)), \\ G(D(u,i)) \end{array} \right\}}} E_{u,r} y_\omega \quad \forall r \in \mathcal{R}, g \in \mathcal{G} \quad (6.12)$$

$$\sum_{\omega \in \Omega | u, i \in \mathcal{UI}(\omega)} y_\omega = 1 \quad \forall u \in \mathcal{U} \quad (6.13)$$

$$y_\omega \in \{0, 1\} \quad \forall \omega \in \Omega \quad (6.14)$$

$$0 \leq e_{g,r} \leq \bar{E}_{g,r}^{ext} \quad \forall g \in \mathcal{G}, r \in \mathcal{R} \quad (6.15)$$

The objective (6.11) is the sum of the cost of individually selected routes and the sum of all costs related to the payment of the extra resources in each geographical locality.

The model contains only two constraints. The first constraint (6.12) is related to the consumption of each resource r in each geographical locality g . The left-hand side of the inequality represents the amount of available resources r in locality g . This quantity is defined by the initial amount of available resource $\bar{E}_{g,r}$ and the quantity of extra resources $e_{g,r}$ that have to be paid. The right-hand side of the inequality represents the consumption related to the selected routes. With ω a selected route and $u, i \in \mathcal{UI}(\omega)$, if the itinerary i has its pickup or its drop-off in geographical position g (i.e., $g \in \{G(P(u,i)), G(D(u,i))\}$), then user u consumes a quantity $E_{u,r}$ of resource r in locality g . The second constraint (6.13) is a classic set partitioning problem where all users have to be served once.

6.4 Solution method

The formulation presented in the previous section specifies our problem. However, the number of routes in Ω makes the problem intractable if formulated in that way. In this section, we will see how we can generate a subset of routes with a heuristic, and solve the route selection problem on this subset called "pool", denoted by Ω' . Because the pool Ω' is a subset of Ω , the routes in Ω' have to be solutions to the single route problem. The heuristic used to fill the subset of routes is the Large Neighborhood Search

(LNS). For each iteration of LNS, the routes of the solution found are added to Ω' . The route selection problem is then solved when the pool of routes contains ν routes. The rest of this section details this solution method.

6.4.1 General framework: Iterative Route Selection (IRS)

The general framework is presented in Algorithm 5 and is denoted by IRS (Iterative Route Selection). This is a hybrid column generation where the columns (i.e., the routes) are generated with an LNS and selected with a master problem formulated as a Mixed-Integer Linear Program (MILP).

The input parameters of IRS are: ν the initial number of routes used to solve the route selection problem; ε the constant positive multiplier for changing the pool size defined by ν ; τ the total time limit; and τ' the time limit for the MILP solver which solves the route selection problem.

Algorithm 5 general framework: IRS

Parameters: $\nu, \varepsilon, \tau, \tau'$
Result: Best solution S^*
 * Initialization * \
 1: $S^* := \emptyset$
 2: **while** execution time $< \tau$ **do**
 * Route generation with an LNS (see Section 6.4.2) * \
 3: $\Omega', S^* := \text{LNS}(S^*, \nu)$
 * Solve the route selection problem * \
 4: Solve (6.11)-(6.15) with an MILP Solver, taking $\Omega = \Omega'$; warm start with S^* ; time limit τ'
 * Update ν * \
 5: **if** the route selection problem has been solved to optimality on two consecutive iterations **then**
 6: $\nu := \lceil \nu \times (1 + \varepsilon) \rceil$
 7: **else if** optimality was not proven on two consecutive iterations and S^* was not improved **then**
 8: $\nu := \lfloor \max\{1, \nu/(1 + \varepsilon)\} \rfloor$
 9: **end if**
 10: **end while**

First, the best solution is initialized empty (line 1). During a given execution time τ (line 2), the framework performs some iterations (lines 3 to 8). For each iteration of the framework, the LNS creates a pool of routes Ω' (line 3). This pool is used to solve the route selection problem (line 4).

The parameter ν is dynamically adjusted according to the capacity of the MILP solver to solve the selection route problem (lines 5 to 8). As ν corresponds to the number of routes in Ω' , it correlates to the difficulty to solve the route selection problem. On the one hand, the optimal solution has a better potential when there are more routes in Ω' . But on the other hand, having more routes makes it more difficult to find this optimal solution. So, our goal is to have a pool of routes Ω' as large as possible but sufficiently small so the MILP solver finds the optimal solution. Hence, this dynamic adjustment of ν ensures that the MILP solver will reach proven optimality approximately half the time.

6.4.2 Large Neighborhood Search (LNS)

The Large Neighborhood Search (LNS) metaheuristic, introduced by Pisinger and Ropke (2007), has been successfully used to solve several DARP variants (e.g., Masson et al. (2014) and Gschwind and Drexel (2019)). It has also been combined to set covering/partitioning problems to create LNS-based metaheuristics in several routing problems (e.g., Parragh and Schmid (2013), Tellez et al. (2018), and Dumez et al. (2021b)). The framework we used for the LNS is presented in Algorithm 6.

The LNS works with three solutions: best (S^*), current (S), and new (S'). At each beginning of the LNS, the current solution and the new solution are initialized with the best solution (line 1). For each iteration, the current solution is copied into the new solution, after which the new solution is destroyed

Algorithm 6 Large Neighborhood Search (LNS) algorithm

Parameters: S^*, ν
Result: Ω', S^*

```

\* Initialization * \
1:  $S := S^*, S' := S^*$ 
2: reset  $\Omega'$  with the routes in  $S^*$ 
3: while  $|\Omega'| < \nu$  do
4:    $S' := S$ 

   \* Removal * \
5:   select a random percentage of users  $\mu$ 
6:   remove  $\mu$  percent of users from  $S'$  using a random removal operator

   \* Insertion * \
7:   solve the “assignment problem”
8:   insert half of the uninserted users in  $S'$  using a random insertion operator
9:   solve the “assignment problem”
10:  insert the remaining uninserted users in  $S'$  using a random insertion operator

   \* Update  $\Omega'$  * \
11:  add the routes in  $S'$  to  $\Omega'$ 

   \* Acceptation criteria * \
12:  if  $S' < S^*$  then
13:     $S^* := S', S := S'$ 
14:    reset  $\Omega'$  with the routes in  $S^*$ 
15:  else if  $S' < S * 1.05$  then
16:     $S := S'$ 
17:  end if
18: end while

```

and repaired (lines 5 to 10). If the newly repaired solution outperforms the best, then it updates the best and the current ones (line 13). In our case, we use the simple “record-to-record” acceptance criterion Dueck (1993) to manage the current solution. If the new solution under-performs the best solution with less than a 5% difference, then it updates the current solution (line 16). If the new solution under-performs the best solution with more than a 5% difference, the new solution is restored to be identical to the current one at the beginning of the new iteration (line 4).

During the LNS iterations, all the routes in the new solution are recorded in Ω' (line 11). The LNS stops when the number of routes in Ω' reaches ν (line 3). At the beginning of an LNS, the pool is initialized with the routes of the best solution S^* only (line 2), and when the LNS improves the best solution S^* (mainly at the beginning of IRS), the pool Ω' is reinitialized (line 14).

At the beginning of each LNS iteration, a new solution identical to the current solution is partially destroyed. A percentage of the users, randomly chosen in $[\underline{\mu}, \bar{\mu}]$, is removed from the new solution (lines 5 to 6). The destroy operators used are the random removal and the historical node-pair removal introduced by Pisinger and Ropke (2007). To repair the solution, the non-inserted users are inserted one by one using a k -regret operator Pisinger and Ropke (2007) with $k \in \{1, 2, 3, 4\}$. In a regular LNS, the users are inserted one by one just after the destroy step. However, the current procedure used here is: 1) choose one itinerary per non-inserted user by solving the “assignment problem” (detailed in Section 6.4.3); 2) insert 50% of non-inserted users; 3) solve the assignment problem again; and 4) insert the remaining non-inserted users (lines 7 to 10).

6.4.3 Assignment problem

The set of non-inserted users is denoted by $U^{no} \subseteq U$. To solve the assignment problem, we first compute an insertion cost estimation $F_{u,i}$ for all non-inserted users $u \in U^{no}$ and for all itineraries $i \in \mathcal{I}_u$. The cost retained for $F_{u,i}$ is the minimum of the insertion costs over all positions of the inserted pickup and drop-off in every route. The insertion cost is defined as the difference between the cost of the route after insertion and before insertion. We also compute the resulting availability $\bar{E}'_{g,r}$ of resources $r \in \mathcal{R}$ in geographical locality $g \in \mathcal{G}$, considering the users who are still inserted in the partially destroyed new solution.

$$\bar{E}'_{g,r} = \bar{E}_{g,r} - \sum_{u \in \left\{ \begin{array}{l} \text{users in the sol.} \\ \text{assigned to } g \end{array} \right\}} E_{u,r} \quad \forall u \in U, g \in \mathcal{G}$$

Then, knowing $F_{u,i}$ and $\bar{E}'_{g,r}$, the following assignment problem is solved to choose the itineraries.

$$\min f^a = \sum_{u \in U^{no}} \sum_{i \in \mathcal{I}_u} F_{u,i} z_{u,i} + \sum_{r \in \mathcal{R}} \sum_{g \in \mathcal{G}} C_r^{ext} e_{g,r} \quad (6.16)$$

s.t.

$$\sum_{i \in \mathcal{I}_u} z_{u,i} = 1 \quad \forall u \in U^{no} \quad (6.17)$$

$$\bar{E}'_{g,r} + e_{g,r} \geq \sum_{u \in U^{no}} \sum_{\substack{i \in \mathcal{I}_u \\ g \in \left\{ \begin{array}{l} G(P(u,i)) \\ G(D(u,i)) \end{array} \right\}}} E_{u,r} z_{u,i} \quad \forall r \in \mathcal{R}, g \in \mathcal{G} \quad (6.18)$$

$$z_{u,i} \in \{0, 1\} \quad \forall u \in U^{no}, i \in \mathcal{I}_u \quad (6.19)$$

$$0 \leq e_{g,r} \leq \bar{E}_{g,r}^{ext} \quad \forall g \in \mathcal{G}, r \in \mathcal{R} \quad (6.20)$$

This problem looks like the route selection problem. However, these are not the routes that are selected, but one itinerary per user. The decision variable $z_{u,i}$ (6.19) indicates whether or not the itinerary i is selected for user u . As in the route selection problem, $e_{g,r}$ (6.20) represents the quantity of extra resource $r \in \mathcal{R}$ that has to be paid to ensure that each user assigned to geographical locality $g \in \mathcal{G}$ will be satisfied. The objective function (6.16) is the sum of the insertion cost estimation, that depends on selected itineraries, and the cost of the extra resources. Constraint (6.17) ensures the selection of

one itinerary per user. Then, constraint (6.18) computes the extra resources needed, depending on the selected itineraries.

It is obvious that the assignment solution is not optimal. First, the insertion cost estimations $F_{u,i}$ are updated only twice per LNS iteration, once at the beginning and once after 50% of the insertions have been completed. Moreover, the insertion cost estimation is just the cheapest insertion at the moment of computation. However, Section 6.6.1 shows numerically that solving the assignment problem more or less frequently during the insertion procedure has little impact and does not significantly improve the overall performance of the method.

6.5 Case study

As presented in the introduction, ADARP is introduced in this chapter to solve a real case of a dial-a-ride with user assignment and MSIs staff sizing.

Instances We have 3 real instances for our problem. Table 6.1 presents their main characteristics. The first instance “Loire” is located in the Loire *département*, France. Despite being centered around a medium-sized city (Saint-Étienne), this area is mainly rural, so there is a low-density demand. The second instance “Rhône” is centered around the city of Lyon and the whole Rhône *département*. The area is urban and densely populated. The last instance “Savoie” covers the whole Savoie *département*. The study area is stretched out, the demand has a low density, and there are many mountains (French Alps) and valleys.

| inst. | users | MSI | depots |
|--------|-------|-----|--------|
| Loire | 42 | 2 | 1 |
| Rhône | 125 | 4 | 2 |
| Savoie | 208 | 5 | 3 |

Table 6.1: number of users, MSIs, and depots for the three instances

Current assignment For these three instances, we know the current user assignment to MSIs. We also know the average resources needed per user, i.e., we know $E_{u,r}$ for all users u and resources r .

Resource availability For these experiments, it was not possible to obtain precisely the real availability of the resources of each MSI, i.e., current staff size. These were therefore estimated by adding all the average resources needed by users.

$$\bar{E}_{g,r} = \sum_{u \in \left\{ \begin{array}{l} \text{users currently} \\ \text{assigned to } g \end{array} \right\}} E_{u,r} \quad \forall r \in \mathcal{R}, g \in \mathcal{G}$$

Resources There are 6 resources. Resources C0 to C4 are different human resources needed by people with disabilities. It is possible to extend these resources up to a certain limit (i.e., $\bar{E}_{g,r}^{ext}$) using overtimes and temporary workers, but these extensions have costs (i.e., C_r^{ext}). The extensions related to C0 to C4 are tactical decisions. Resource C5 represents the MSI’s accreditation, i.e., the number of users that can be assigned to an institution. In our basic instances, this resource cannot be changed because it seems to be a strategic decision to change the size of the building, the dining hall, the facilities, etc. However, a sensitivity analysis on these values is still presented in Section 6.7.1.

Time and distance The time and distance values from node i to j (i.e., $\Gamma_{i,j}$ and $\Delta_{i,j}$) have been computed using the API detailed in Mapotempo (n.d.).

Maximal ride time The theoretical maximal ride time of a user is equal to the duration of the direct itinerary of its current assignment, multiplied by a parameter called Direct Time Factor (DTF). MSIs would theoretically like to observe $DTF = 1.5$ but in practice, the max ride time is not really controlled by them. That is why the following experiments are done with $DTF = 1.25, 1.5, \text{ and } 1.75$.

At the same time, any ride time up to 15 minutes is accepted by MSIs. Therefore, for each user, the maximal ride time used in the following experiments is the maximal value between 15 minutes and the current direct duration multiplied by DTF ($H_u = \max\{15; \max\{\Gamma_{P_{ui}, D_{ui}}\} \times DTF\}, \forall u \in \mathcal{U}, i \in \mathcal{I}_u$).

Time Windows The reception time of MSIs are between 8:00 and 9:15. Therefore, all drop-offs happen within this time frame. For a DTF of 1.75, the width of the users' time windows follows the distribution presented in Table 6.2. Obviously, the distribution is shifted toward the smaller time windows when the DTF is decreased.

| inst. | width of the time window in minutes | | | | | |
|--------|-------------------------------------|----------|----------|----------|----------|----------|
| | [40, 44] | [45, 49] | [50, 54] | [55, 59] | [60, 64] | [65, 69] |
| Loire | 0 | 10 | 7 | 6 | 17 | 2 |
| Rhône | 13 | 5 | 1 | 4 | 98 | 4 |
| Savoie | 9 | 46 | 39 | 47 | 66 | 1 |

Table 6.2: number of users relative to the width of the time windows for a $DTF = 1.75$

Itineraries For each user, all of its potential itineraries originate from his/her home. By default, all MSIs can receive all users. However, the maximum ride time precludes some itineraries from home to MSIs when the direct ride time for the itinerary is longer than the maximum ride time. Thus, the number of feasible itineraries increases with the DTF, as shown in Table 6.3. For the Savoie instance with a DTF of 1.5, there are on average 1.97 feasible itineraries per user. If the DTF is reduced to 1.25, this number decreases to 1.57. On the opposite, if the DTF is increased to 1.75, then the average number of feasible itineraries increases to 2.33 per user. According to Table 6.4 whatever the chosen DTF, there is always at least one user with the maximum number of feasible itineraries. In other words, at least one user can go to any of the MSIs.

| inst. | DTF | | |
|--------|------|------|------|
| | 1.25 | 1.50 | 1.75 |
| Loire | 1.60 | 1.71 | 1.79 |
| Rhône | 2.14 | 2.51 | 2.68 |
| Savoie | 1.57 | 1.97 | 2.33 |

Table 6.3: average number of feasible itineraries (i.e., MSIs) per user depending on DTF

Vehicles There are 4 types of vehicles with different capacities and costs. The number of each type of vehicle available at each depot is not limited. In our case, buying a vehicle is actually a tactical decision by MSIs because it is easy for them to sell, buy, exchange, or reconfigure vehicles to adapt their fleet as needed.

Depots The depots are located at the car park of some MSIs. At the beginning of the day, the drivers travel to these MSIs with their own means of transport and take the MSI's vehicles. The drivers start and end their routes at the same MSI. As these MSIs have dedicated car parks, we can consider that there is no maximal number of vehicles at the depots.

| inst. | DTF | nb of feasible itineraries | | | | |
|--------|------|----------------------------|----|----|----|----|
| | | 1 | 2 | 3 | 4 | 5 |
| Loire | 1.25 | 17 | 25 | | | |
| | 1.50 | 12 | 30 | | | |
| | 1.75 | 9 | 33 | | | |
| Rhône | 1.25 | 54 | 26 | 19 | 26 | |
| | 1.50 | 44 | 19 | 16 | 46 | |
| | 1.75 | 36 | 24 | 9 | 56 | |
| Savoie | 1.25 | 128 | 53 | 16 | 10 | 1 |
| | 1.50 | 114 | 37 | 22 | 20 | 15 |
| | 1.75 | 97 | 34 | 18 | 29 | 30 |

Table 6.4: number of users with i feasible itineraries (i.e., MSIs) depending on DTF

6.6 Validation of the method

The solution method was coded in C++ and compiled with g++ 11. CPLEX 22.1.0 was used for solving the route selection problem. The experiments were performed on an AMD EPYC 7702 2.1GHz processor. As in many articles in this research field, 5 runs are done per instance for numerical experiments. The values of the solution method parameters are the following. The initial number of routes ν in the pool Ω' that stops the LNS iterations is set to 100 routes. The increasing or decreasing percentage ε of the number of routes ν is set to 25%. The MILP solver limit τ' for each IRS iteration is set to 10 seconds. The proportion of users removed for each iteration of LNS is randomly chosen from a uniform distribution between $[\underline{\mu}, \bar{\mu}] = [10\%, 45\%]$. Lastly, the total time τ available to run IRS on instances Loire, Rhône, and Savoie are 30, 60, and 120 minutes respectively. Each instance uses a different value for τ because the number of users significantly differs among them. This value ensures a sufficient number of iterations in order to reach good solutions.

6.6.1 Validation of the components and tuning of parameters

The route and the assignment components of the IRS are evaluated in Table 6.5. All values are the percentage deviation from the Best-Known Solutions (BKS, presented in Appendix A.5). The first column presents the solutions obtained with the IRS framework presented in this work. As we can see, the average gap obtained by the IRS is 0.32%. The results are pretty stable as the worst average gap over five runs is only 0.77% (for the Loire instance with DFT = 1.75)

The noRSP (no route selection problem) column presents the case where the LNS is used to generate solutions and the route selection problem is never solved. For instance, in Algorithm 5 line 3 is used but line 4 is not. In this case, the average gap with the BKS increases to 2.06% and the results are not really stable. As an interpretation, the pool of routes acts as a memory that can be used to build better solutions. By using the LNS without the route selection problem, the solver is short-sighted as it just moves from one solution to another. Having a pool of saved routes allows the solver to simultaneously consider many routes that were generated at different moments.

In IRS, two assignment problems are solved at each iteration of LNS: at the beginning of the insertion procedures and after 50% of re-inserted users. The noAP (no Assignment Problem) column presents the case where an itinerary is assigned to a user by the k-regret insertion operator for each insertion during the LNS, and not by solving assignment problems with a MILP solver. In other words, instead of having only one candidate itinerary per user thanks to the assignment problem, all itineraries are candidates for insertion. Thus, the number of insertions that must be tested increases. As a consequence, the LNS takes more time for each insertion. In this case, the k-regret choice is based on the cheapest itinerary that maintains the capacity constraints. We can see the noAP method under-perform the IRS method by

$1.70 - 0.32 = 1.38\%$.

Lastly, the LNS column corresponds to noRSP and noAP together. The method is a simple LNS with a record-to-record acceptance criterion and the k-regret insertion operator is used to select the best assignment of an itinerary to a user. With an average gap of 11.32%, this method is clearly not efficient.

| inst. | DTF | IRS | noRSP | noAP | LNS |
|---------|------|------|-------|------|-------|
| Loire | 1.25 | 0.00 | 0.00 | 0.00 | 1.70 |
| | 1.5 | 0.16 | 1.34 | 1.37 | 3.34 |
| | 1.75 | 0.77 | 1.30 | 1.57 | 1.57 |
| Rhône | 1.25 | 0.61 | 1.28 | 0.56 | 7.57 |
| | 1.5 | 0.26 | 1.43 | 1.29 | 13.96 |
| | 1.75 | 0.28 | 1.70 | 1.87 | 11.71 |
| Savoie | 1.25 | 0.24 | 1.93 | 1.08 | 11.63 |
| | 1.5 | 0.34 | 4.88 | 3.57 | 23.23 |
| | 1.75 | 0.22 | 4.71 | 3.99 | 27.19 |
| average | | 0.32 | 2.06 | 1.70 | 11.32 |

Table 6.5: evaluation of the components, average gap with the BKS in % for 5 runs

Table 6.6 shows how the number of times the assignment problem is solved, influences the overall performance of our matheuristic. Table 6.7 shows this influence on the number of iterations of the IRS and the LNS, and as well as on the size of the pool of routes ν . For ν , its initial value at the beginning of the resolution, its average value across the resolution, and its final value at the end of the resolution are provided. In this table, the metrics are averaged over all the instances for all the DTF's values for the 5 runs. In both tables, column "1" represents the case where the assignment problem is solved only once after the destroy phase and at the beginning of the insertion phase; columns "2", "3", and "4" represent the cases where the assignment problem is solved 2, 3 or 4 times during one iteration, respectively, i.e after the destroy phase and every 50%, 33%, and 25% of inserted users. Then, column "all" is the case where the assignment problem is solved before each user insertion.

According to Table 6.6, the best result is obtained when the assignment problem is solved twice during an iteration of the LNS. Thus, this value is used in the rest of the experiments and serves as a basis for the following analysis. The main observation is that, irrespective of the number of assignment problems we solve from 1 to 4, the performance is fairly good and stable. Indeed on average, the gap difference is 0.2% ($= 0.52 - 0.32$) or less. The second observation is that solving the assignment problem each time a user is inserted gives poor results. In this case, the number of IRS iterations decreases by 71% ($= \frac{24.11-82.51}{82.51}$) and the number of LNS iterations decreases by 79%. This leads to a 0.84% ($= 1.16 - 0.32$) gap difference which is fairly larger than the one observed among columns 1, 2, 3, 4. While Table 6.6 shows the importance of the assignment problem, Table 6.7 shows that it is not required to solve the assignment problem often. Solving it too often will reduce the number of iterations of the IRS, but on the opposite, solving it too rarely will decrease the quality of the selected itineraries during the insertion phase of the LNS. The numerical results show that the balance is obtained when the assignment problem is solved twice during the LNS. However, these are the results for using the minimum insertion cost as a criterion for the route selection problem. Another criterion (e.g., average insertion cost) may result in different performances. As the results were good with minimum insertion cost, we decided to continue with it. Nevertheless, criterion selection is an interesting topic for future work. Our last observation is that the size of the pool of routes ν for the route selection problem increases during the resolution. It means that the MILP solver is able to handle a growing number of routes. This can be explained by the following. The MILP solver uses a warm start with the best solution. In addition, the proportion of bad routes relative to the best solution increases as the best solution improves. As a consequence, it becomes easier for the MILP solver to eliminate bad solutions during the route selection problem.

| inst. | DTF | number of solved assignment problems | | | | |
|---------|------|--------------------------------------|------|------|------|------|
| | | 1 | 2* | 3 | 4 | all |
| Loire | 1.25 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | 1.5 | 0.16 | 0.16 | 0.16 | 0.00 | 0.65 |
| | 1.75 | 1.54 | 0.77 | 1.03 | 1.88 | 1.54 |
| Rhône | 1.25 | 0.79 | 0.61 | 0.38 | 0.49 | 0.74 |
| | 1.5 | 0.13 | 0.26 | 0.19 | 0.39 | 0.92 |
| | 1.75 | 0.19 | 0.28 | 0.29 | 0.29 | 1.03 |
| Savoie | 1.25 | 0.07 | 0.24 | 0.26 | 0.38 | 0.79 |
| | 1.5 | 0.60 | 0.34 | 0.61 | 0.65 | 2.81 |
| | 1.75 | 0.26 | 0.22 | 0.27 | 0.63 | 1.92 |
| average | | 0.42 | 0.32 | 0.36 | 0.52 | 1.16 |

*: chosen parameter

Table 6.6: number of times the assignment problem is solved during an iteration of the LNS, average gap with the BKS in % for 5 runs per line

| number of solved assignment problems | 1 | 2* | 3 | 4 | 0 |
|--------------------------------------|------------|-----------|-----------|-----------|-----------|
| nb LNS iterations | 113,416.31 | 76,173.96 | 68,450.20 | 57,601.27 | 15,932.73 |
| nb IRS iterations | 98.38 | 82.51 | 68.33 | 60.24 | 24.11 |
| initial ν | 100 | 100 | 100 | 100 | 100 |
| average ν | 3,147.55 | 2,909.77 | 2,795.80 | 2,722.17 | 1,447.42 |
| final ν | 5,761.82 | 5,465.62 | 5,320.42 | 5,187.24 | 3,801.73 |

*: chosen parameter

Table 6.7: metrics relative to the number of times the assignment problem is solved during an iteration of the LNS

6.6.2 Comparison with the literature

Generalized Dial-A-Ride Problem (GDARP) The closest problem in the literature to ours is the Generalized Dial-A-Ride Problem (GDARP) presented by Pinson et al. (2016). The authors use a Greedy Randomized Adaptive Search Procedure (GRASP) to solve their problems. In the GDARP, each user has one origin and one destination. The specificity is that a user can walk from his/her origin to one of the possible pickup locations and again walk from one of the possible drop-off locations to his/her destination. In this case, the ride happens only between the selected pickup and drop-off. Similarly to our problem, a user has multiple possible pickup and drop-off locations (i.e., multiple possible itineraries). However, there is no resource allocation similar to the staff sizing in our problem. In the model presented by Pinson et al. (2016), the objective is to minimize the itinerary duration of drivers and users from their respective origins to their respective destinations. To establish a proper comparison, our algorithm has been slightly adapted to use the same objective function.

Table 6.8 presents the results of the comparison between their results with a GRASP with 500 iterations and our IRS algorithm with 500 iterations on an instance with 50 users. To be fair, the LNS iterations are counted in the 500 iterations of the IRS. For each algorithm, results are given with one run. For each instance, IRS ameliorates the BKS slightly.

| inst. | GRASP | IRS |
|-------|--------|--------|
| CM 51 | 16,232 | 16,202 |
| CM 52 | 14,532 | 14,505 |
| CM 53 | 13,912 | 13,896 |
| CM 54 | 13,821 | 13,790 |
| CM 55 | 13,824 | 13,814 |

Table 6.8: comparison of costs for Pinson et al. (2016) instances with 500 iterations for GRASP and IRS

Vehicle Routing Problem with Delivery Options (VRPDO) Another problem close to ours is the Vehicle Routing Problem with Delivery Options (VRPDO) introduced by Dumez et al. (2021a) and Dumez et al. (2021b). Clients have several preferences for their delivery options, the total number of first and second preferred options is constrained, some delivery options have capacities, and the objective is to minimize first the number of vehicles and then the cost, based on the distance only. Our algorithm has been adapted to these specific features. Table 6.9 presents the comparison between the best solution found with two methods (LNS-SPP and MathHeu) combining different LNSs and set partitioning problems with 5 runs in Dumez et al. (2021a) and Dumez et al. (2021b) and the best solution found with 5 runs of the IRS algorithm. IRS’s computation times are 30, 60, and 120 minutes for instance sizes 50, 100, and 200 respectively. The table presents for each best solution its number of routes (“Nb routes”), its cost (“Cost”), and the cost gap between the LNS-SPP & MathHeu and the IRS.

Instance types U and V have 2 and 1.5 delivery options per client on average respectively. We note that these instance types are fairly well solved with IRS. However, we find that the IRS algorithm performs poorly for the instance type UBC. The explanation is that the UBC instances are the U instances with large vehicle capacity. With large vehicle capacity, the length of the routes increases, so the time needed to perform an insertion with the k-regret procedure also increases. For this reason, the number of investigated routes during the IRS algorithm is reduced. This case, with large vehicle capacity and long routes, is rather far from our case study.

6.7 Managerial results

At this point, it is reasonable to consider that the IRS algorithm is able to provide fairly good results on our instances. In this section, several analyses will be presented on our case study to derive managerial results from the MSIs’ perspective and the users’ perspective.

| type | size | LNS-SPP & MathHeu | | IRS | | Gap |
|------|------|-------------------|----------|-----------|----------|----------|
| | | nb routes | cost (€) | nb routes | cost (€) | cost (%) |
| U | 50 | 54 | 3,864 | 54 | 3,869 | 0.13 |
| | 100 | 105 | 6,499 | 105 | 6,511 | 0.19 |
| | 200 | 205 | 11,726 | 205 | 11,763 | 0.32 |
| UBC | 50 | 20 | 2,293 | 21* | 2,272 | -0.92* |
| | 100 | 40 | 3,580 | 40 | 3,617 | 1.03 |
| | 200 | 80 | 6,048 | 80 | 6,181 | 2.19 |
| V | 50 | 54 | 3,742 | 54 | 3,720 | -0.59 |
| | 100 | 104 | 7,001 | 104 | 6,964 | -0.53 |
| | 200 | 203 | 13,669 | 203 | 13,471 | -1.45 |

*: As the number of routes is not the same, the cost cannot be compared

Table 6.9: comparison with the best solutions in the literature Dumez et al. (2021a) and Dumez et al. (2021b) and the best solutions for 5 runs of IRS

Four scenarios are considered. Scenario (A) refers to the current real case. Most of the time in the real case, a vehicle transports only those users who go to the same MSI, and the users are never reassigned from their current MSI to another one to improve transportation performance (including the user's ride time). Scenario (B) refers to an emerging practice where the transportation facilities of the MSIs are shared. Therefore, users from different MSIs can be in the same vehicle at the same time. In this scenario, users are not reassigned. Without being able to measure it precisely, we find that this sharing remains infrequent but is gradually being practiced more between the MSIs. Scenario (C) refers to the case where the vehicles are not shared but where users are assigned to the MSIs, taking into account the transportation cost. Scenario (D) refers to the case of ADARP, with users' assignments and vehicles sharing between MSIs.

6.7.1 From the MSIs' perspective

Cost and routes Table 6.10 presents the cost and the number of routes for the 4 scenarios. As Scenario (A) is taken as a reference, the cost and the number of routes are presented as absolute values for this scenario and as relative values (gap in %) for the other scenarios. We note that the average saving from the vehicles sharing without assignment (B) is 8.12% of the cost of the current practice (A). By comparison, assignment without sharing (C) represents a 12.26% gain, and sharing and allocating (D) represents a 16.91% gain. We can easily conclude that the users' reassignment from one facility to another is a fine opportunity to reduce the expenses of socio-medical facilities. We also find that the evolution of the number of routes is strongly correlated to the cost savings in Scenarios (B) (C) and (D). Beyond the cost reduction, the reduction of the number of routes is an objective for the MSIs that would enable them to manage fewer vehicles and therefore also fewer drivers. We find a clear route number reduction tendency with the introduction of assignment flexibility. In some cases, this decrease can amount to more than 36% owing to the possibility of assignment only (Scenario (C), Inst. Sa, DTF=1.25). To conclude, the benefit related to a reassignment policy seems equivalent to the cost reduction related to vehicle sharing. Moreover, these two changes can be combined to further improve economic performance. However, from a managerial point of view, reassignment could be more advantageous than vehicle sharing which could potentially be unpractical when the MSIs have to charge one another for the transport service. The reassignment option should then be considered a priority by MSIs wishing to collaborate with one another.

| | | cost | | | | nb routes | | | |
|---------|------|-------|-------|--------|--------|-----------|--------|--------|--------|
| | | A | B | C | D | A | B | C | D |
| | | no | no | yes | yes | no | no | yes | yes |
| | | no | yes | no | yes | no | yes | no | yes |
| | | unit | € | % | % | # | % | % | % |
| inst. | DTF | | | | | | | | |
| Loire | 1.25 | 1,248 | -7.16 | -13.44 | -19.66 | 13 | -15.38 | -15.38 | -30.77 |
| | 1.5 | 1,023 | -9.89 | -17.04 | -20.26 | 11 | -18.18 | -27.27 | -36.36 |
| | 1.75 | 885 | -8.73 | -14.13 | -16.03 | 9 | -11.11 | -22.22 | -22.22 |
| Rhône | 1.25 | 1,931 | -9.52 | -9.49 | -16.74 | 26 | -15.38 | -15.38 | -26.92 |
| | 1.5 | 1,601 | -9.52 | -6.90 | -14.22 | 20 | -15.00 | -15.00 | -15.00 |
| | 1.75 | 1,408 | -5.58 | -3.57 | -8.97 | 16 | 6.25 | 0.00 | 0.00 |
| Savoie | 1.25 | 4,225 | -9.19 | -19.95 | -23.93 | 46 | -15.22 | -36.96 | -34.78 |
| | 1.5 | 3,458 | -7.34 | -13.73 | -18.01 | 34 | -5.88 | -26.47 | -23.53 |
| | 1.75 | 3,137 | -6.17 | -12.11 | -14.34 | 30 | -10.00 | -16.67 | -20.00 |
| average | | | -8.12 | -12.26 | -16.91 | | -11.10 | -19.48 | -23.29 |

Table 6.10: effect of user assignment and sharing of vehicles on the cost and the number of routes

Impact on the MSIs' payrolls Table 6.11 presents the evolution of the two main expenses for MSIs: the cost related to the routes, and staff wages. These two costs are compared for Scenarios (A) and (D). Wages are the sum of the current wages of the staff (1,931€, 6,910€, and 7,760€ for respectively Loire, Rhône, and Savoie) and the overtime (cost for the extension of capacity). We can see that the sharing and the assignment decrease the routing cost significantly (up to 28.63%) but increase the cost of wages only slightly (no more than 2.56%). From a managerial point of view, it is interesting to note that the increase in salaries is relatively low, and therefore does not lead to a radical change in the payrolls of the MSIs. Thus, it could be considered that the allocation of human resources to institutions would be affected only slightly by the implementation of a reassignment policy.

| | | routes | | wages | | |
|---------|------|--------|--------|-------|------|---|
| | | scena. | A | D | A | D |
| | | unit | € | % | € | % |
| inst. | DTF | | | | | |
| Loire | 1.25 | 1,247 | -22.28 | 1,931 | 1.69 | |
| | 1.5 | 1,023 | -22.36 | 1,931 | 1.11 | |
| | 1.75 | 884 | -17.94 | 1,931 | 0.87 | |
| Rhône | 1.25 | 1,931 | -21.96 | 6,909 | 1.46 | |
| | 1.5 | 1,600 | -15.74 | 6,910 | 0.35 | |
| | 1.75 | 1,408 | -10.70 | 6,910 | 0.35 | |
| Savoie | 1.25 | 4,225 | -28.63 | 7,760 | 2.56 | |
| | 1.5 | 3,457 | -22.79 | 7,760 | 2.13 | |
| | 1.75 | 3,137 | -19.17 | 7,760 | 1.95 | |
| average | | | -20.17 | | 1.39 | |

Table 6.11: evolution of the two main expenses for MSIs: routes and wages between Scenarios (A) and (D)

Number of user places Until now, the capacity of the MSIs (i.e., the number of users served) has been fixed as it is difficult to increase this value in real life. However, we studied the impact of such an increase on Scenario (D). Hence, Table 6.12 presents the cost of Scenario (A) compared with the different increase of C5 in Scenario (D). As we can see, most of the assignment gain can be obtained without increasing their capacities. In addition, this gain does not seem significant above a 5% capacity increase.

| | | scena. | A | | D | | | |
|---------|------|--------|--------|--------|--------|--------|--------|--------|
| | | C5 | 0% | +0% | +5% | +10% | +20% | +∞ |
| | | unit | € | % | % | % | % | % |
| inst. | DTF | | | | | | | |
| Loire | 1.25 | 1,247 | -19.60 | -22.40 | -24.98 | -26.65 | -28.42 | |
| | 1.5 | 1,023 | -20.23 | -22.29 | -23.46 | -24.05 | -25.15 | |
| | 1.75 | 884 | -15.94 | -17.45 | -20.24 | -21.27 | -24.17 | |
| Rhône | 1.25 | 1,931 | -16.72 | -18.08 | -19.43 | -19.50 | -19.71 | |
| | 1.5 | 1,600 | -14.12 | -14.86 | -15.68 | -15.66 | -15.69 | |
| | 1.75 | 1,408 | -8.95 | -11.15 | -10.65 | -10.57 | -10.67 | |
| Savoie | 1.25 | 4,225 | -23.84 | -26.32 | -27.00 | -27.72 | -27.72 | |
| | 1.5 | 3,457 | -17.97 | -20.70 | -21.61 | -23.19 | -23.22 | |
| | 1.75 | 3,137 | -14.34 | -17.38 | -18.77 | -19.29 | -19.20 | |
| average | | | | -16.86 | -18.96 | -20.20 | -20.88 | -21.55 |

Table 6.12: impact of increasing the number of user places (Capacity C5)

6.7.2 From people with disabilities and other vulnerabilities perspective

Table 6.13 presents two indicators from a user’s perspective. The first one is the ride time, which has to be as short as possible. The second one is the number of reassignments, which should also be as low as possible to avoid upsetting the previous habits of the user and the staff. These indicators are important as they are directly linked to the quality of service. We find that sharing vehicles (B) does not have a significant impact on the ride time. However, the reassignment does, with an average reduction of ride time of more than 5% (C, D). On the second indicator, we find that the cost saving for Scenario (D) can be obtained by reassigning less than a quarter of the users.

6.8 Conclusion

In this chapter, we addressed an important issue in the medico-social sector that impacts the quality of care, the cost of the system, and the system’s environmental footprint. The goal is to integrate logistic aspects into users’ assignment decisions by simultaneously solving three interrelated problems: the allocation of the users, the transportation of the users, and the assignment of the medical staff. We named this new problem the Assignment Dial-A-Ride Problem (ADARP). It can be seen as an extension of the GDARP and the VRPDO. This problem-solving approach jointly optimizes the routing for the shared transportation service, the assignment of users with disabilities and other vulnerabilities to geographical locations, and the extra resources needed at these locations.

To address this problem, we developed a matheuristic, the iteration route selection algorithm (IRS) which combines an LNS and a MILP solver. The LNS generates a pool of routes and the MILP solver chooses the best combination of these routes. For each iteration of the LNS, an assignment problem is solved twice during the insertion procedure. The method was validated on instances from the literature for the GDARP and the VRPDO. We then considered 3 real instances from the sanitary transportation of 3 *départements* in France.

| | | ride time | | | | reassigned user | |
|---------|---------|-----------|-------|--------|--------|-----------------|-------|
| | | A | B | C | D | C | D |
| | scena. | no | no | yes | yes | yes | yes |
| | assign. | no | yes | no | yes | no | yes |
| | shared | no | yes | no | yes | no | yes |
| | unit | min. | % | % | % | % | % |
| inst. | DTF | | | | | | |
| Loire | 1.25 | 36.44 | -1.10 | -1.77 | -3.06 | 19.05 | 23.81 |
| | 1.5 | 41.06 | 5.76 | -2.48 | -1.01 | 23.81 | 19.05 |
| | 1.75 | 44.97 | 10.64 | 3.34 | 4.30 | 19.05 | 14.29 |
| Rhône | 1.25 | 25.14 | 0.60 | -3.10 | -3.27 | 12.80 | 12.80 |
| | 1.5 | 28.09 | -0.67 | -2.38 | -3.96 | 9.60 | 8.00 |
| | 1.75 | 30.79 | -3.87 | -3.95 | -8.42 | 8.00 | 8.00 |
| Savoie | 1.25 | 31.04 | -1.12 | -12.30 | -11.03 | 24.04 | 21.63 |
| | 1.5 | 34.64 | 0.53 | -11.26 | -7.51 | 23.56 | 17.79 |
| | 1.75 | 40.09 | -4.24 | -16.29 | -13.63 | 19.23 | 17.31 |
| average | | 34.69 | 0.73 | -5.58 | -5.29 | 17.68 | 15.85 |

Table 6.13: ride time evolution and proportion of users moved from their current assignment

The key finding is that a proper user assignment is paramount for improving the current transportation system. The saving induced by the assignment is equivalent to that obtained by the transport sharing. Assignment and sharing can be combined to achieve a greater cost reduction. Compared with the current practice, this combination reduces costs by 17% on average and up to 24% without degrading the service quality. From the institutions' perspective, transportation costs can be reduced with very small changes in the staff size. From the perspective of users in vulnerable situations, the savings come with a slight decrease in transportation time and less than a quarter of users are reassigned.

For future research, we plan to develop a model to take human resources into account more precisely. For example, the ability to reallocate existing human resources between MSIs, considering the cost, the skills, and the distances, may be taken into account. A second line of research is related to the location of institutions themselves. From the public health policy point of view, it would be relevant to measure the impact of the opening or closing of an institution, or to modify its capacity in terms of access to care in a territory.

Chapter 7

Conclusions and perspectives

In this final chapter, we summarize the main findings of this thesis. Additionally, we point out the limitations and propose future directions for research.

7.1 Conclusions

The Dial-A-Ride Problem (DARP) is a classic problem of Operations Research (OR). It is one of the most advanced variants of routing problems. In this thesis, we worked on 3 variants of the DARP. To solve these DARPs, we developed four solution methods based on state-of-the-art metaheuristics from the OR literature and Machine Learning (ML) techniques.

Thanks to the advancements in computational power and artificial intelligence over the last decade, methods developed within the machine learning community are spreading to other disciplinary fields. Today, more and more researchers use ML techniques as tools to solve their problems. In Chapter 4, we explore this new possibility for solving the DARP in its classic form. As such, we developed two solution methods based on machine learning and tested them on the state-of-the-art benchmark A from Cordeau (2006).

The first solution method tackled Research Question RQ1¹ in an attempt to bridge Research Gaps RG1², RG2³, and RG3⁴. The method is based on an offline learning procedure. The main goal is to learn the characteristics of good DARP solutions and arcs. We first generated a large data set, and then trained classifiers to differentiate good and bad solutions/arcs. Based on these predictions, we derived four guidelines to build good DARP solutions: we should aim for a balanced distribution of requests between routes, low connection ranks, low Forward Time Slacks (FTS), and high number of apparitions to number of successful scheduling tests ratios. For each guideline, we created a new repair heuristic and included it in a Large Neighborhood Search (LNS) metaheuristic. Among the new operators, the best

¹RQ1: How can we build an algorithm that computes fast and optimal solutions for the regular non-urgent sanitary transportation?

²RG1: The current state-of-the-art solution methods for the DARP are not able to tackle real-life sized instances.

³RG2: The characteristics of good DARP solutions have yet to be identified through a rigorous process and used to solve the DARP.

⁴RG3: Many machine learning methods have yet to be tested on the DARP and compared to state-of-the-art solution methods from the operations research literature.

results were obtained with the FTS insertion and connection rank insertion. In addition, the operators were able to generalize the learning obtained on medium-sized instances to large instances. Unfortunately, none of the new knowledge-guided heuristics were able to improve the performance of the initial LNS. Despite interesting results and new insights, outperforming the well tried and tested LNS is still a real challenge for this offline ML-based method.

The second solution method also tackled Research Question RQ1⁵, trying to bridge Research Gaps RG1⁶ and RG3⁷. The second solution method uses the Nested Rollout Policy Adaptation (NRPA) at its core. The NRPA is an online learning method based on the Monte Carlo Tree Search (MCTS). Inspired by the very good results on games and some preliminary works on simpler routing problems, we adapt the NRPA for solving the DARP. The NRPA is used within a greedy insertion framework. It is in charge of creating the ordered list of requests to be inserted sequentially. The NRPA is a recursive algorithm that intensifies the search in the neighbors of the best solution. At the same time, it uses different levels of recursion to diversify the search. The NRPA demonstrates good learning capabilities on small and medium-sized instances (i.e., less than 50 users). However, the gap between the NRPA and the LNS grows with larger instances. In addition, the NRPA requires significantly longer computation times. As a consequence, although the NRPA is able to solve the DARP, it is still far from a current state-of-the-art algorithm from the OR literature.

In Chapter 5, we introduce a new variant of the DARP: the Commuting Dial-A-Ride Problem (Com-DARP). This work provided new insights on Research Question RQ1⁵ and its related Research Gap RG1⁶. At the same time, it successfully addressed Research Question RQ3⁸ and bridged Research Gap RG5⁹. This new variant is inspired by the real-life transportation of disabled children but can easily be translated to any transportation system based on commuting. The novelty lies in the dependency between requests. Indeed, a daily maximal ride time is introduced to take advantage of a possible asymmetry between morning and afternoon trips. For example, a long morning trip can be compensated by a short afternoon trip. Furthermore, we provide a mathematical formulation that encompasses and extends the Com-DARP. This formulation corresponds to a multi-request multi-trip multi-depot heterogeneous fleet DARP. We solve the Com-DARP using a metaheuristic based on the Small And Large Neighborhood Search (SLNS) and add a speed-up mechanism. This mechanism takes advantage of the dynamic precedence relations between the nodes. The complete algorithm with the SLNS and the Precedence Filter (PF) is called SLNS-PF. From an algorithmic perspective, the SLNS-PF improves upon a classic LNS and the PF component speeds up the computation by 39% on average (up to 49%). From a managerial perspective, it is beneficial to use the Com-DARP formulation instead of two independent DARPs. Indeed, it allows either to reduce the cost or to improve the quality of service in terms of maximal ride time allowed.

In Chapter 6, another variant of the DARP is introduced: the Assignment Dial-A-Ride Problem (ADARP). This work focused on Research Gap RG4¹⁰, successfully providing new insights on Research Question RQ2¹¹. This work is motivated by three problems faced by the medico-social sector: the assignment of users to medical facilities, the transportation of the users, and the allocation of the medical staff. Prior to this study, the three problems were never considered simultaneously but as three independent problems. Merging the problems allows to explore a bigger research space. On the one hand, better solutions are now available. On the other hand, the increased size of the research space makes it more challenging to find these solutions. We developed a new solution method called the Iterative Route Search (IRS). The IRS is a matheuristic based on an LNS and a MILP solver. The LNS generates a pool of routes and the MILP solver chooses the best combination of these routes. One key finding is that the sharing of vehicles between medical facilities is the main contributor to cost reduction.

⁵RQ1: How can we build an algorithm that computes fast and optimal solutions for the regular non-urgent sanitary transportation?

⁶RG1: The current state-of-the-art solution methods for the DARP are not able to tackle real-life sized instances.

⁷RG3: Many machine learning methods have yet to be tested on the DARP and compared to state-of-the-art solution methods from the operations research literature.

⁸RQ3: How the dependence between requests can be used to improve the transportation?

⁹RG5: There are few works on shared constraints for the DARP. In particular, there is no study on a shared maximum ride time constraint.

¹⁰RG4: There is no model that encompasses the entire scope of the regular non-urgent sanitary transportation of disabled children in France and considers all the aspects related to the different stakeholders.

¹¹RQ2: How can we extend the scope of the optimization and consider the different aspects related to all the stakeholders?

Similarly, it is the proper assignment of users to medical facilities that improves the quality of service in terms of ride time. Put together, the ADARP allows to significantly decrease the transportation cost, and slightly decrease the ride time of the users. In addition, these improvements do not require reassigning more than a quarter of the users.

7.2 Perspectives

During this thesis, we had to choose between which idea to investigate and which idea to put aside. At the same time, the investigation of some ideas gave birth to new ones. Among the ones that we put aside and the new ones, the following presents the most interesting perspectives for future work.

The two ML-based methods we developed in Chapter 4 showed interesting results despite their lower performance compared to the state-of-the-art solution method from the OR literature.

In the first approach, the good predictions from the classifiers did not translate to good LNS operators. Here are some possible reasons to explain the results. 1) Were the predictions good enough? We had an accuracy score between 60 and 75%. This is better than random guessing, but far from a 100% accuracy. Finding better metrics or classifiers could improve the accuracy and thus the reliability of the guidelines. 2) Can more elaborated guidelines improve the performance? We created operators based on guidelines, themselves based on the classifiers' predictions. In the end, the knowledge-guided operators are rather far from the classifiers. Using the classifiers directly in a metaheuristic framework could be a solution to keep the learning from the training process as untouched as possible. For example, insertions could be filtered using a decision tree directly integrated into an LNS. 3) Is the scope of our work adequate? In this thesis, we study the DARP at solution and arc level. However, it is possible to repeat this study using routes or groups of users as the main scope. 4) How should the learning be used? We created new operators for the LNS based on the results of the learning process. However, it could be possible to integrate the learning differently. Among the possibilities, bad arcs could be penalized or knowledge-guided operators could be used only during particular phases of the search.

In the second approach, the NRPA was not on par with the LNS both in terms of solution quality and in terms of computational speed. 1) To improve the computation speed, the NRPA could complete partial solutions instead of starting from scratch. 2) In our implementation, the NRPA only adds one task: choosing the next request to insert. The NRPA could be charged with additional decisions such as selecting the insertion heuristics or the route for the next insertion. Adding a removal component to the NRPA could also be possible. However, adding tasks to the NRPA inevitably increases the size of its decision space.

In Chapter 5, the Com-DARP was presented with classic features except for the daily maximum ride time. Although it showed good results, this work could easily be improved in order to be deployed in real-life applications. 1) The first step would be to include more realistic features such as a heterogeneous fleet and users, multiple depots, or (re)configurable vehicles. 2) Then, testing the algorithm on real-life instances would allow to test its robustness over instances with specific features. For example, the demand distribution is dense in cities, sparse in rural areas, and clustered in mountainous regions.

We point out two perspectives from Chapter 6. 1) Currently, medical staff is just considered as a resource. However, we could consider a deeper level of integration by taking into account their trips to the medical facilities, their planning, or their skill level. 2) The study was focused on decisions at operational and tactical levels (i.e., user allocation, vehicle routing, staff assignment, and overtime work). Nevertheless, we derived some insights at the strategic level with a brief study on the size of the medical facilities. To complete this analysis at the strategic level, it could be interesting to question the geographical location of the facilities themselves. Indeed, proper location could allow more users to benefit from these institutions while reducing transportation costs.

Chapter 8

Final words

As a young explorer on the vast ocean of knowledge, I modestly push back, ever so slightly, the boundary between what we know and what we don't know. Despite facing many hardships, the journey has been a source of joy and fulfillment. I sincerely hope that other young researchers will set forth in a similar direction with outstanding crew members such as Samuel, Céline, and Thibaud. Even though we embark on separate journeys, our shared aspiration to make the world a better place will undoubtedly continue to guide us on the vast ocean of knowledge.

Bibliography

- Accorsi, Luca, Andrea Lodi, and Daniele Vigo (2022). “Guidelines for the computational testing of machine learning approaches to vehicle routing problems”. In: *Operations Research Letters* 50.2, pp. 229–234.
- Accorsi, Luca and Daniele Vigo (2021). “A Fast and Scalable Heuristic for the Solution of Large-Scale Capacitated Vehicle Routing Problems”. In: *Transportation Science* 55.4, pp. 832–856.
- Ackermann, Christian and Julia Rieck (2022). “New Optimization Guidance for Dynamic Dial-a-Ride Problems”. In: *Operations Research Proceedings 2021*. Ed. by Norbert Trautmann and Mario Gnägi. Lecture Notes in Operations Research. Cham: Springer International Publishing, pp. 283–288.
- Alders, Peter and Frederik T. Schut (2019). “Trends in ageing and ageing-in-place and the future market for institutional care: scenarios and policy implications”. In: *Health Economics, Policy and Law* 14.1, pp. 82–100.
- Alzubi, Jafar, Anand Nayyar, and Akshi Kumar (2018). “Machine Learning from Theory to Algorithms: An Overview”. In: *Journal of Physics: Conference Series* 1142.1, p. 012012.
- ANAP (2016). *Transport - Improving the Management of Transport for Disabled People - Volume 1: Feedback (fr: Transports - Améliorer la gestion des transports de personnes handicapées - Tome 1 : Retour d’expériences)*. ANAP.
- Anenberg, Susan, Joshua Miller, DAVEN Henze, and Ray Minjares (2019). “A global snapshot of the air pollution-related health impacts of transportation sector emissions in 2010 and 2015”. In: *International Council on Clean Transportation: Washington, DC, USA*.
- Aringhieri, R., M.E. Bruni, S. Khodaparasti, and J.T. Van Essen (2017). “Emergency medical services and beyond: Addressing new challenges through a wide literature review”. In: *Computers & Operations Research* 78, pp. 349–368.
- Armbrust, Philipp, Philipp Hungerländer, Kerstin Maier, and Veronika Pachatz (2022). “Case study of Dial-a-Ride Problems arising in Austrian rural regions”. In: *Transportation Research Procedia*. 24th Euro Working Group on Transportation Meeting 62, pp. 197–204.
- Arnold, Florian and Kenneth Sörensen (2019). “What makes a VRP solution good? The generation of problem-specific knowledge for heuristics”. In: *Computers & Operations Research* 106, pp. 280–288.
- Asta, Shahriar, Daniel Karapetyan, Ahmed Kheiri, Ender Özcan, and Andrew J. Parkes (2016). *Combining Monte-Carlo and Hyper-heuristic methods for the Multi-mode Resource-constrained Multi-project Scheduling Problem*.

- Atahran, Ahmed, Christophe Lenté, and Vincent T'kindt (2014). "A Multicriteria Dial-a-Ride Problem with an Ecological Measure and Heterogeneous Vehicles". In: *Journal of Multi-Criteria Decision Analysis* 21.5, pp. 279–298.
- Baltean-Lugojan, R, R Misener, P Bonami, and A Tramontani (2018). "Strong sparse cut selection via trained neural nets for quadratic semidefinite outer-approximations". In: *Imperial College, London, Tech. Rep.*
- Banister David, Moshe Givoni, ed. (2010). *Integrated Transport: From Policy to Practice*. London: Routledge. 368 pp.
- Basso, Rafael, Balázs Kulcsár, Ivan Sanchez-Diaz, and Xiaobo Qu (2022). "Dynamic stochastic electric vehicle routing with safe reinforcement learning". In: *Transportation Research Part E: Logistics and Transportation Review* 157, p. 102496.
- Baty, Léo, Kai Jungel, Patrick S. Klein, Axel Parmentier, and Maximilian Schiffer (2023). *Combinatorial Optimization enriched Machine Learning to solve the Dynamic Vehicle Routing Problem with Time Windows*.
- Baugh Jr, John W, Gopala Krishna Reddy Kakivaya, and John R Stone (1998). "Intractability of the dial-a-ride problem and a multiobjective solution using simulated annealing". In: *Engineering Optimization* 30.2, pp. 91–123.
- Beaudry, Alexandre, Gilbert Laporte, Teresa Melo, and Stefan Nickel (2010). "Dynamic transportation of patients in hospitals". In: *OR Spectrum* 32.1, pp. 77–107.
- Belhaiza, Slim (2019). "A Hybrid Adaptive Large Neighborhood Heuristic for a Real-Life Dial-a-Ride Problem". In: *Algorithms* 12.2, p. 39.
- Bellamy, Vanessa, Fabrice Lenglard, Valérie Bauer-Eubriet, and Élisabeth Castaing (2023). "Le handicap en chiffres – Édition 2023". In: *Panoramas de la DRESS*.
- Bello, Irwan, Hieu Pham, Quoc V. Le, Mohammad Norouzi, and Samy Bengio (2017). "Neural Combinatorial Optimization with Reinforcement Learning". In: arXiv:1611.09940.
- Bengio, Yoshua, Andrea Lodi, and Antoine Prouvost (2021). "Machine learning for combinatorial optimization: A methodological tour d'horizon". In: *European Journal of Operational Research* 290.2, pp. 405–421.
- Berbeglia, Gerardo, Jean-François Cordeau, and Gilbert Laporte (2012). "A Hybrid Tabu Search and Constraint Programming Algorithm for the Dynamic Dial-a-Ride Problem". In: *INFORMS Journal on Computing* 24.3, pp. 343–355.
- Bischi, Bernd et al. (2016). "ASlib: A benchmark library for algorithm selection". In: *Artificial Intelligence* 237, pp. 41–58.
- Bogyrbayeva, Aigerim, Meraryslan Meraliyev, Taukekhan Mustakhov, and Bissenbay Dauletbayev (2022). *Learning to Solve Vehicle Routing Problems: A Survey*.
- Bongiovanni, Claudia, Mor Kaspi, Jean-François Cordeau, and Nikolaos Geroliminis (2020). "A predictive large neighborhood search for the dynamic electric autonomous dial-a-ride problem". In: *9th Symposium of the European Association for Research in Transportation*.
- Bongiovanni, Claudia, Mor Kaspi, Jean-François Cordeau, and Nikolaos Geroliminis (2022). "A machine learning-driven two-phase metaheuristic for autonomous ridesharing operations". In: *Transportation Research Part E: Logistics and Transportation Review* 165, p. 102835.
- Bongiovanni, Claudia, Mor Kaspi, and Nikolaos Geroliminis (2019). "The electric autonomous dial-a-ride problem". In: *Transportation Research Part B: Methodological* 122, pp. 436–456.
- Braekers, Kris, An Caris, and Gerrit K. Janssens (2014). "Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots". In: *Transportation Research Part B: Methodological* 67, pp. 166–186.
- Braekers, Kris and Attila A. Kovacs (2016). "A multi-period dial-a-ride problem with driver consistency". In: *Transportation Research Part B: Methodological* 94, pp. 355–377.
- Browne, Cameron B. et al. (2012). "A Survey of Monte Carlo Tree Search Methods". In: *IEEE Transactions on Computational Intelligence and AI in Games* 4.1, pp. 1–43.
- Brůhová Foltýnová, Hana, Eliška Vejchodská, Kristýna Rybová, and Viktor Květoň (2020). "Sustainable urban mobility: One definition, different stakeholders' opinions". In: *Transportation Research Part D: Transport and Environment* 87, p. 102465.

- Burke, E.K., G. Kendall, and E. Soubeiga (2003). “A Tabu-Search Hyperheuristic for Timetabling and Rostering”. In: *Journal of Heuristics* 9.6, pp. 451–470.
- Calvin, Katherine et al. (2023). *IPCC, 2023: Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, H. Lee and J. Romero (eds.)]*. IPCC, Geneva, Switzerland. Intergovernmental Panel on Climate Change (IPCC).
- Cappart, Quentin, Charles Thomas, Pierre Schaus, and Louis-Martin Rousseau (2018). “A Constraint Programming Approach for Solving Patient Transportation Problems”. In: *Principles and Practice of Constraint Programming* 11008. Ed. by John Hooker, pp. 490–506.
- Carnes, Timothy A., Shane G. Henderson, David B. Shmoys, Mahvareh Ahghari, and Russell D. MacDonald (2013). “Mathematical Programming Guides Air-Ambulance Routing at Ornge”. In: *Interfaces* 43.3, pp. 232–239.
- Carotenuto, Pasquale and Fabio Martis (2017). “A double dynamic fast algorithm to solve multi-vehicle Dial a Ride Problem”. In: *Transportation Research Procedia*. 20th EURO Working Group on Transportation Meeting, EWGT 2017, 4-6 September 2017, Budapest, Hungary 27, pp. 632–639.
- Cattaruzza, Diego, Nabil Absi, and Dominique Feillet (2018). “Vehicle routing problems with multiple trips”. In: *Annals of Operations Research* 271.1, pp. 127–159.
- Cazenave, Tristan (2009). “Nested monte-carlo search”. In: *Twenty-First International Joint Conference on Artificial Intelligence*.
- Cazenave, Tristan, Jean-Yves Lucas, Thomas Triboulet, and Hyoseok Kim (2021). “Policy adaptation for vehicle routing”. In: *AI Communications* 34.1, pp. 21–35.
- Chane-Haï, Timothée, Samuel Vercaene, and Thibaud Monteiro (2023a). “The Assignment Dial-A-Ride Problem”. In: *Health Care Management Science*.
- Chane-Haï, Timothée, Samuel Vercaene, Céline Robardet, and Thibaud Monteiro (2022). “Guidage des métaheuristiques par machine-learning, application au transport d’enfants en situation de handicap”. In: *23ème congrès annuel de la Société Française de Recherche Opérationnelle et d’Aide à la Décision*. Villeurbanne - Lyon, France: INSA Lyon.
- Chane-Haï, Timothée, Samuel Vercaene, Céline Robardet, and Thibaud Monteiro (2023b). “The Commuting Dial-A-Ride Problem (Com-DARP)”. In.
- Chapman, Lee (2007). “Transport and climate change: a review”. In: *Journal of Transport Geography* 15.5, pp. 354–367.
- Chassaing, M., C. Duhamel, and P. Lacomme (2016). “An ELS-based approach with dynamic probabilities management in local search for the Dial-A-Ride Problem”. In: *Engineering Applications of Artificial Intelligence* 48, pp. 119–133.
- Chen, Li-Wen, Ta-Yin Hu, and Yu-Wen Wu (2022). “A bi-objective model for eco-efficient dial-a-ride problems”. In: *Asia Pacific Management Review* 27.3, pp. 163–172.
- Chen, Xinyun and Yuandong Tian (2019). “Learning to Perform Local Rewriting for Combinatorial Optimization”. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc.
- Chevrier, Rémy, Arnaud Liefoghe, Laetitia Jourdan, and Clarisse Dhaenens (2012). “Solving a dial-a-ride problem with a hybrid evolutionary multi-objective approach: Application to demand responsive transport”. In: *Applied Soft Computing* 12.4, pp. 1247–1258.
- Chlebíková, Janka, Clément Dallard, and Niklas Paulsen (2023). “Impact of soft ride time constraints on the complexity of scheduling in Dial-A-Ride Problems”. In: *Theoretical Computer Science* 960, p. 113923.
- Choong, Shin Siang, Li-Pei Wong, and Chee Peng Lim (2018). “Automatic design of hyper-heuristic based on reinforcement learning”. In: *Information Sciences* 436-437, pp. 89–107.
- Christiaens, Jan and Greet Vanden Berghe (2020). “Slack Induction by String Removals for Vehicle Routing Problems”. In: *Transportation Science* 54.2, pp. 417–433.
- Christodoulou, Aris and Panayotis Christidis (2020). *Measuring congestion in European cities: a focus on Brussels, Seville and Krakow*.
- Clarke, G. and J. W. Wright (1964). “Scheduling of Vehicles from a Central Depot to a Number of Delivery Points”. In: *Operations Research* 12.4, pp. 568–581.

- Coppi, Alberto, Paolo Detti, and Jessica Raffaelli (2013). “A planning and routing model for patient transportation in health care”. In: *Electronic Notes in Discrete Mathematics* 41, pp. 125–132.
- Cordeau, J-F, M. Gendreau, G. Laporte, J-Y Potvin, and F. Semet (2002). “A guide to vehicle routing heuristics”. In: *Journal of the Operational Research Society* 53.5, pp. 512–522.
- Cordeau, Jean-François (2006). “A Branch-and-Cut Algorithm for the Dial-a-Ride Problem”. In: *Operations Research* 54.3, pp. 573–586.
- Cordeau, Jean-François and Gilbert Laporte (2003). “A tabu search heuristic for the static multi-vehicle dial-a-ride problem”. In: *Transportation Research Part B: Methodological* 37.6, pp. 579–594.
- Cordeau, Jean-François and Gilbert Laporte (2007). “The dial-a-ride problem: models and algorithms”. In: *Annals of Operations Research* 153.1, pp. 29–46.
- Corne, David, Clarisse Dhaenens, and Laetitia Jourdan (2012). “Synergies between operations research and data mining: The emerging use of multi-objective approaches”. In: *European Journal of Operational Research* 221.3, pp. 469–479.
- Coster, Joanne E., Janette K. Turner, Daniel Bradbury, and Anna Cantrell (2017). “Why Do People Choose Emergency and Urgent Care Services? A Rapid Review Utilizing a Systematic Literature Search and Narrative Synthesis”. In: *Academic Emergency Medicine: Official Journal of the Society for Academic Emergency Medicine* 24.9, pp. 1137–1149.
- Coulom, Rémi (2007). “Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search”. In: *Computers and Games*. Ed. by H. Jaap Van Den Herik, Paolo Ciancarini, and H. H. L. M. Donkers. Red. by David Hutchison et al. Vol. 4630. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 72–83.
- Cubillos, Claudio, Nivaldo Rodriguez, and Broderick Crawford (2007). “A Study on Genetic Algorithms for the DARP Problem”. In: *Bio-inspired Modeling of Cognitive Tasks*. Ed. by José Mira and José R. Álvarez. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 498–507.
- Cubillos, Claudio, Enrique Urrea, and Nivaldo Rodríguez (2009). “Application of Genetic Algorithms for the DARPTW Problem”. In: *International Journal Of Computers Communications & Control* 4.2, pp. 127–136.
- Curtis, Frank E. and Katya Scheinberg (2017). “Optimization Methods for Supervised Machine Learning: From Linear Models to Deep Learning”. In: *Leading Developments from INFORMS Communities*. INFORMS TutORials in Operations Research. INFORMS, pp. 89–114.
- Cyr, Melissa E., Anna G. Etchin, Barbara J. Guthrie, and James C. Benneyan (2019). “Access to specialty healthcare in urban versus rural US populations: a systematic literature review”. In: *BMC Health Services Research* 19.1, p. 974.
- Dantzig, G. B. and J. H. Ramser (1959). “The Truck Dispatching Problem”. In: *Management Science* 6.1, pp. 80–91.
- de la Rue du Can, Stephane et al. (2022). *Climate Change 2022: Mitigation of Climate Change. Contribution of Working Group III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change, Chapter 11*, 1973106, ark:/13030/qt23n103jv.
- De San Miguel, Kristen and Gill Lewin (2008). “Brief Report: Personal emergency alarms: What impact do they have on older people’s lives?” In: *Australasian Journal on Ageing* 27.2, pp. 103–105.
- De Vries, Jan and Robbert Huijsman (2011). “Supply chain management in health services: an overview”. In: *Supply Chain Management: An International Journal* 16.3. Ed. by Jan De Vries, pp. 159–165.
- Deasy, C and I O’Sullivan (2007). “Transfer of patients—from the spoke to the hub”. In: *Irish medical journal* 100.7, pp. 538–539.
- Degroote, Hans (2018). “Applying Algorithm Selection – a Case Study for the Generalised Assignment Problem”. In: *Electronic Notes in Discrete Mathematics* 69, pp. 205–212.
- Deng, Yong, Yang Liu, and Deyun Zhou (2015). “An Improved Genetic Algorithm with Initial Population Strategy for Symmetric TSP”. In: *Mathematical Problems in Engineering* 2015, e212794.
- Detti, Paolo, Francesco Papalini, and Garazi Zabalo Manrique de Lara (2017). “A multi-depot dial-a-ride problem with heterogeneous vehicles and compatibility constraints in healthcare”. In: *Omega* 70, pp. 1–14.
- Di Gaspero, Luca and Tommaso Urli (2012). “Evaluation of a Family of Reinforcement Learning Cross-Domain Optimization Heuristics”. In: *Learning and Intelligent Optimization*. Ed. by Youssef Hamadi and Marc Schoenauer. Vol. 7219. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 384–389.

- Diana, Marco and Maged M. Dessouky (2004). “A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows”. In: *Transportation Research Part B: Methodological* 38.6, pp. 539–557.
- Dong, Sharon (2022). “New Formulations and Solution Methods for the Dial-a-ride Problem”. Thesis. UNSW Sydney.
- Dong, Xiaotong, David Rey, and S. Travis Waller (2020). “Dial-a-Ride Problem with Users’ Accept/Reject Decisions Based on Service Utilities”. In: *Transportation Research Record* 2674.10, pp. 55–67.
- Droogh, Joep M., Marije Smit, Anthony R. Absalom, Jack JM Ligtenberg, and Jan G. Zijlstra (2015). “Transferring the critically ill patient: are we there yet?”. In: *Critical Care* 19.1, p. 62.
- Dueck, Gunter (1993). “New Optimization Heuristics: The Great Deluge Algorithm and the Record-to-Record Travel”. In: *Journal of Computational Physics* 104.1, pp. 86–92.
- Dumez, Dorian, Fabien Lehuédé, and Olivier Péton (2021a). “A large neighborhood search approach to the vehicle routing problem with delivery options”. In: *Transportation Research Part B: Methodological* 144, pp. 103–132.
- Dumez, Dorian, Christian Tilk, Stefan Irnich, Fabien Lehuédé, and Olivier Péton (2021b). “Hybridizing large neighborhood search and exact methods for generalized vehicle routing problems with time windows”. In: *EURO Journal on Transportation and Logistics* 10, p. 100040.
- Euchi, Jalel, Malek Masmoudi, and Patrick Siarry (2022). “Home health care routing and scheduling problems: a literature review”. In: *4OR* 20.3, pp. 351–389.
- European Commission. Directorate General for Economic and Financial Affairs. and Economic Policy Committee of the European Communities. (2019). *Joint report on health care and long-term care systems & fiscal sustainability: country documents : 2019 update*. LU: Publications Office.
- Fahmy, Sherif A. (2023). “Hybrid and approximate algorithms for the dial-a-ride problem with adaptive ride times considering different service strategies”. In: *Engineering Optimization* 55.8, pp. 1263–1277.
- Fassbender, Klaus et al. (2013). “Streamlining of prehospital stroke management: the golden hour”. In: *The Lancet Neurology* 12.6, pp. 585–596.
- Feillet, Dominique, Thierry Garaix, Fabien Lehuédé, Olivier Péton, and Dominique Quadri (2014). “A new consistent vehicle routing problem for the transportation of people with disabilities”. In: *Networks* 63.3, pp. 211–224.
- Feng, Lei, Elise Miller-Hooks, Paul Schonfeld, and Matthew Mohebbi (2014). “Optimizing Ridesharing Services for Airport Access”. In: *Transportation Research Record* 2467.1, pp. 157–167.
- Firat, Murat and Gerhard J. Woeginger (2011). “Analysis of the dial-a-ride problem of Hunsaker and Savelsbergh”. In: *Operations Research Letters* 39.1, pp. 32–35.
- Fleischmann, Bernhard (1990). “The vehicle routing problem with multiple use of the vehicles”. In: *Expert Systems with Applications* 61, pp. 262–271.
- Foguel, Manuel et al. (2016). “Non-emergency patient transport services planning through genetic algorithms”. In: *Expert Systems with Applications* 61, pp. 262–271.
- Fortun, M. and S. S. Schweber (1993). “Scientists and the Legacy of World War II: The Case of Operations Research (OR)”. In: *Social Studies of Science* 23.4, pp. 595–642.
- Gaans, Deborah van and Elsa Dent (2018). “Issues of accessibility to health services by older Australians: a review”. In: *Public Health Reviews* 39.1, p. 20.
- Gambella, Claudio, Bissan Ghaddar, and Joe Naoum-Sawaya (2021). “Optimization Problems for Machine Learning: A Survey”. In: *European Journal of Operational Research* 290.3, pp. 807–828.
- Gao, Lei et al. (2020). *Learn to Design the Heuristics for Vehicle Routing Problem*.
- Garaix, Thierry, Christian Artigues, Dominique Feillet, and Didier Josselin (2011). “Optimization of occupancy rate in dial-a-ride problems via linear fractional column generation”. In: *Computers & Operations Research* 38.10, pp. 1435–1442.
- Garey, Michael R. and David S. Johnson (1979). *Computers and intractability: a guide to the theory of NP-completeness*. 27. print. A series of books in the mathematical sciences. New York [u.a]: Freeman. 338 pp.
- Gaul, Daniela, Kathrin Klamroth, and Michael Stiglmayr (2021). “Solving the Dynamic Dial-a-Ride Problem Using a Rolling-Horizon Event-Based Graph”. In: *21st Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2021)*. Ed. by Matthias Müller-Hannemann and Federico Perea. Vol. 96. Open Access Series in Informatics (OASIs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 8:1–8:16.

- Gelly, Sylvain et al. (2012). “The grand challenge of computer Go: Monte Carlo tree search and extensions”. In: *Communications of the ACM* 55.3, pp. 106–113.
- Gendreau, Michel and Jean-Yves Potvin (2005). “Metaheuristics in Combinatorial Optimization”. In: *Annals of Operations Research* 140.1, pp. 189–213.
- Genet, Nadine et al. (2011). “Home care in Europe: a systematic literature review”. In: *BMC Health Services Research* 11.1, p. 207.
- Ghiani, Gianpaolo and Gennaro Improta (2000). “An efficient transformation of the generalized vehicle routing problem”. In: *European Journal of Operational Research* 122.1, pp. 11–17.
- Glover, Fred and Claude McMillan (1986). “The general employee scheduling problem. An integration of MS and AI”. In: *Computers & Operations Research. Applications of Integer Programming* 13.5, pp. 563–573.
- Gogna, Anupriya and Akash Tayal (2013). “Metaheuristics: review and application”. In: *Journal of Experimental & Theoretical Artificial Intelligence* 25.4, pp. 503–526.
- Gomes, Carla (2001). “Algorithm portfolios”. In: *Artificial Intelligence* 126.1, pp. 43–62.
- Gonzalez, Lucie, Romain Roussel, Jean-Cyprien Héam, Myriam Mikou, and (2017). *Health expenditure in 2017 - Results of the health accounts - 2018 ed. (fr: Les dépenses de santé en 2017 - Résultats des comptes de la santé - Édition 2018.)* Direction de la recherche, des études, de l'évaluation et des statistiques.
- Gschwind, Timo (2019). “Route feasibility testing and forward time slack for the Synchronized Pickup and Delivery Problem”. In: *OR Spectrum* 41.2, pp. 491–512.
- Gschwind, Timo and Michael Drexl (2019). “Adaptive Large Neighborhood Search with a Constant-Time Feasibility Test for the Dial-a-Ride Problem”. In: *Transportation Science* 53.2, pp. 480–491.
- Gschwind, Timo and Stefan Irnich (2015). “Effective Handling of Dynamic Time Windows and Its Application to Solving the Dial-a-Ride Problem”. In: *Transportation Science* 49.2, pp. 335–354.
- Guerriero, Francesca, Ferdinando Pezzella, Ornella Pisacane, and Luigi Trollini (2014). “Multi-objective Optimization in Dial-a-ride Public Transportation”. In: *Transportation Research Procedia*. 17th Meeting of the EURO Working Group on Transportation, EWGT2014, 2-4 July 2014, Sevilla, Spain 3, pp. 299–308.
- Guo, Shuocheng, Iman Dayarian, Jian Li, and Xinwu Qian (2023). *A branch-cut-and-price algorithm for a dial-a-ride problem with minimum disease-transmission risk*.
- Hains, Isla M., Anne Marks, Andrew Georgiou, and Johanna I. Westbrook (2011). “Non-emergency patient transport: what are the quality and safety issues? A systematic review”. In: *International Journal for Quality in Health Care: Journal of the International Society for Quality in Health Care* 23.1, pp. 68–75.
- Häll, Carl H., Henrik Andersson, Jan T. Lundgren, and Peter Värbrand (2009). “The Integrated Dial-a-Ride Problem”. In: *Public Transport* 1.1, pp. 39–54.
- Häll, Carl H. and Anders Peterson (2013). “Improving paratransit scheduling using ruin and recreate methods”. In: *Transportation Planning and Technology* 36.4, pp. 377–393.
- Hanne, Thomas, Teresa Melo, and Stefan Nickel (2009). “Bringing Robustness to Patient Flow Management Through Optimized Patient Transports in Hospitals”. In: *Interfaces* 39.3, pp. 241–255.
- Hee-Su Kim and Sung-Bae Cho (2001). “An efficient genetic algorithm with less fitness evaluation by clustering”. In: *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*. 2001 Congress on Evolutionary Computation. Vol. 2. Seoul, South Korea: IEEE, pp. 887–894.
- Heilporn, Géraldine, Jean-François Cordeau, and Gilbert Laporte (2011). “An integer L-shaped algorithm for the Dial-a-Ride Problem with stochastic customer delays”. In: *Discrete Applied Mathematics* 159.9, pp. 883–895.
- Ho, Sin C. and Dag Haugland (2011). “Local search heuristics for the probabilistic dial-a-ride problem”. In: *OR Spectrum* 33.4, pp. 961–988.
- Ho, Sin C. et al. (2018). “A survey of dial-a-ride problems: Literature review and recent developments”. In: *Transportation Research Part B: Methodological* 111, pp. 395–421.
- Holland, JohnH (1975). “Adaptation in natural and artificial systems, univ. of mich. press”. In: *Ann Arbor* 7, pp. 390–401.

- Holmes, Wendy (2021). *Projecting the Need for and Cost of Long-Term Care for Older Persons*. Asian Development Bank.
- Hoos, Holger H. (2011). “Automated Algorithm Configuration and Parameter Tuning”. In: *Autonomous Search*. Ed. by Youssef Hamadi, Eric Monfroy, and Frédéric Saubion. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 37–71.
- Hopfield, John and D Tank (1985). “Neural Computation of Decisions in Optimization Problems”. In: *Biological cybernetics* 52, pp. 141–52.
- Hu, Ta-Yin and Chin-Ping Chang (2015). “A revised branch-and-price algorithm for dial-a-ride problems with the consideration of time-dependent travel cost”. In: *Journal of Advanced Transportation* 49.6, pp. 700–723.
- Hu, Ta-Yin, Guan-Chun Zheng, and Tsai-Yun Liao (2019). “Multi-Objective Model for Dial-a-Ride Problems with Vehicle Speed Considerations”. In: *Transportation Research Record* 2673.11, pp. 161–171.
- Huggins, Chris and David Shugg (2008). “Non-Emergency Patient Transport in Victoria: An overview”. In: *Australasian Journal of Paramedicine* 6.4.
- Hunger, J. and G. Huttner (1999). “Optimization and analysis of force field parameters by combination of genetic algorithms and neural networks”. In: *Journal of Computational Chemistry* 20.4, pp. 455–471.
- Hunsaker, Brady and Martin Savelsbergh (2002). “Efficient feasibility testing for dial-a-ride problems”. In: *Operations Research Letters* 30.3, pp. 169–173.
- Jagtenberg, C. J., P. L. van den Berg, and R. D. van der Mei (2017). “Benchmarking online dispatch algorithms for Emergency Medical Services”. In: *European Journal of Operational Research* 258.2, pp. 715–725.
- Jain, Siddhartha and Pascal Van Hentenryck (2011). “Large Neighborhood Search for Dial-a-Ride Problems”. In: *Principles and Practice of Constraint Programming – CP 2011*. Ed. by Jimmy Lee. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 400–413.
- James, Spencer L et al. (2020). “Morbidity and mortality from road injuries: results from the Global Burden of Disease Study 2017”. In: *Injury Prevention* 26 (Suppl 1), pp. i46–i56.
- Jaques, John (1859). “The Icosian Game”. In: *London.[1: 2]*.
- Jaw, Jang-Jei, Amedeo R. Odoni, Harilaos N. Psaraftis, and Nigel H. M. Wilson (1986). “A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows”. In: *Transportation Research Part B: Methodological* 20.3, pp. 243–257.
- Jin, Scarlett T., Hui Kong, Rachel Wu, and Daniel Z. Sui (2018). “Ridesourcing, the sharing economy, and the future of cities”. In: *Cities* 76, pp. 96–104.
- Jorgensen, R M, J Larsen, and K B Bergvinsdottir (2007). “Solving the Dial-a-Ride problem using genetic algorithms”. In: *Journal of the Operational Research Society* 58.10, pp. 1321–1331.
- Joshi, Chaitanya K., Thomas Laurent, and Xavier Bresson (2019). “An Efficient Graph Convolutional Network Technique for the Travelling Salesman Problem”. In: *arXiv:1906.01227 [cs, stat]*.
- Juan, Angel A., Javier Faulin, Josep Jorba, Jose Caceres, and Joan Manuel Marquès (2013). “Using parallel & distributed computing for real-time solving of vehicle routing problems with stochastic demands”. In: *Annals of Operations Research* 207.1, pp. 43–65.
- Kairanbay, Magzhan and Hajar Mat Jani (2013). “A Review and Evaluations of Shortest Path Algorithms”. In: *International Journal of Scientific & Technology Research* 2, pp. 99–104.
- Kelle, Peter, John Woosley, and Helmut Schneider (2012). “Pharmaceutical supply chain specifics and inventory solutions for a hospital case”. In: *Operations Research for Health Care* 1.2, pp. 54–63.
- Kergosien, Y., Ch. Lenté, D. Piton, and J. -C. Billaut (2011). “A tabu search heuristic for the dynamic transportation of patients between care units”. In: *European Journal of Operational Research* 214.2, pp. 442–452.
- Kerschke, Pascal, Holger H. Hoos, Frank Neumann, and Heike Trautmann (2018). *Automated Algorithm Selection: Survey and Perspectives*.
- Khalil, Elias B., Bistra Dilkina, George L. Nemhauser, Shabbir Ahmed, and Yufen Shao (2017). “Learning to Run Heuristics in Tree Search”. In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. Twenty-Sixth International Joint Conference on Artificial Intelligence. Melbourne, Australia: International Joint Conferences on Artificial Intelligence Organization, pp. 659–666.

- Khanassov, Vladimir et al. (2016). “Organizational interventions improving access to community-based primary health care for vulnerable populations: a scoping review”. In: *International Journal for Equity in Health* 15.1, p. 168.
- Kim, Minsu, Jinkyoo Park, and Joungho Kim (2021). “Learning Collaborative Policies to Solve NP-hard Routing Problems”. In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., pp. 10418–10430.
- Kirchler, Dominik and Roberto Wolfler Calvo (2013). “A Granular Tabu Search algorithm for the Dial-a-Ride Problem”. In: *Transportation Research Part B: Methodological* 56, pp. 120–135.
- Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi (1983). “Optimization by Simulated Annealing”. In: *Science* 220.4598, pp. 671–680.
- Kironji, Antony Gatebe et al. (2018). “Identifying barriers for out of hospital emergency care in low and low-middle income countries: a systematic review”. In: *BMC Health Services Research* 18.1, p. 291.
- Kocsis, Levente and Csaba Szepesvári (2006). “Bandit Based Monte-Carlo Planning”. In: *Machine Learning: ECML 2006*. Ed. by Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou. Red. by David Hutchison et al. Vol. 4212. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 282–293.
- Kool, Wouter, Herke van Hoof, and Max Welling (2019). “Attention, Learn to Solve Routing Problems!” In: *arXiv:1803.08475 [cs, stat]*.
- Kraus, Mathias, Stefan Feuerriegel, and Asil Oztekin (2020). “Deep learning in business analytics and operations research: Models, applications and managerial implications”. In: *European Journal of Operational Research*. Featured Cluster: Business Analytics: Defining the field and identifying a research agenda 281.3, pp. 628–641.
- Krumke, Sven O. et al. (2006). “On Minimizing the Maximum Flow Time in the Online Dial-a-Ride Problem”. In: *Approximation and Online Algorithms*. Ed. by Thomas Erlebach and Giuseppe Persinao. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 258–269.
- Kulshrestha, Ashish and Jasveer Singh (2016). “Inter-hospital and intra-hospital patient transfer: Recent concepts”. In: *Indian Journal of Anaesthesia* 60.7, pp. 451–457.
- Laporte, Gilbert (1992). “The traveling salesman problem: An overview of exact and approximate algorithms”. In: *European Journal of Operational Research* 59.2, pp. 231–247.
- Laporte, Gilbert (2009). “Fifty Years of Vehicle Routing”. In: *Transportation Science* 43.4, pp. 408–416.
- Larsen, Eric et al. (2022). “Predicting Tactical Solutions to Operational Planning Problems Under Imperfect Information”. In: *INFORMS Journal on Computing* 34.1, pp. 227–242.
- Lehuédé, Fabien, Renaud Masson, Sophie N Parragh, Olivier Péton, and Fabien Tricoire (2014). “A multi-criteria large neighbourhood search for the transportation of disabled people”. In: *Journal of the Operational Research Society* 65.7, pp. 983–1000.
- Lesteven, P., E. Robert, T. Wahl, and P.E. Grimonprez (2016). *Health Transportation - 2016 Expenditure Review (fr: Les transports sanitaires - Revue de dépenses 2016.)* Inspection générale des affaires sociales (IGAS).
- Levesque, Jean-Frederic, Mark F. Harris, and Grant Russell (2013). “Patient-centred access to health care: conceptualising access at the interface of health systems and populations”. In: *International Journal for Equity in Health* 12.1, p. 18.
- Lewis, Mark O, S Balaji, and Arun Rai (2010). “Rfid-enabled capabilities and their impact on healthcare process performance”. In: *Organization Theory*.
- Li, Jingpeng, Edmund K. Burke, and Rong Qu (2011). “Integrating neural networks and logistic regression to underpin hyper-heuristic search”. In: *Knowledge-Based Systems* 24.2, pp. 322–330.
- Li, Jingpeng, Edmund K. Burke, and Rong Qu (2012). “A pattern recognition based intelligent search method and two assignment problem case studies”. In: *Applied Intelligence* 36.2, pp. 442–453.
- Li, Jingwen et al. (2022). “Heterogeneous Attentions for Solving Pickup and Delivery Problem via Deep Reinforcement Learning”. In: *IEEE Transactions on Intelligent Transportation Systems* 23.3, pp. 2306–2315.
- Li, Mengyu, Peter Vanberkel, and Alix J. E. Carter (2019). “A review on ambulance offload delay literature”. In: *Health Care Management Science* 22.4, pp. 658–675.
- Li, Sirui, Zhongxia Yan, and Cathy Wu (2021). “Learning to delegate for large-scale vehicle routing”. In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., pp. 26198–26211.

- Liang, Xiao, Gonçalo Homem de Almeida Correia, Kun An, and Bart van Arem (2020). “Automated taxis’ dial-a-ride problem with ride-sharing considering congestion-based dynamic travel times”. In: *Transportation Research Part C: Emerging Technologies* 112, pp. 260–281.
- Lim, Andrew, Zhenzhen Zhang, and Hu Qin (2017). “Pickup and Delivery Service with Manpower Planning in Hong Kong Public Hospitals”. In: *Transportation Science* 51.2, pp. 688–705.
- Lindstrøm, Cebirina and Stefan Røpke (2022). “Group Consistent Dial-A-Ride using Adaptive Large Neighborhood Search”. In: *13th International Conference on Computational Logistics*.
- Liu, Mengyang, Zhixing Luo, and Andrew Lim (2015). “A branch-and-cut algorithm for a realistic dial-a-ride problem”. In: *Transportation Research Part B: Methodological* 81, pp. 267–288.
- Lodi, Andrea and Giulia Zarpellon (2017). “On learning and branching: a survey”. In: *TOP* 25.2, pp. 207–236.
- Lopes, Leo and Kate Smith-Miles (2013). “Generating Applicable Synthetic Instances for Branch Problems”. In: *Operations Research* 61, pp. 563–577.
- Lu, Chang, Yuehui Wu, and Shanchuan Yu (2022). “A Sample Average Approximation Approach for the Stochastic Dial-A-Ride Problem on a Multigraph with User Satisfaction”. In: *European Journal of Operational Research* 302.3, pp. 1031–1044.
- Lu, Hao, Xingwen Zhang, and Shuang Yang (2019). “A Learning-based Iterative Method for Solving Vehicle Routing Problems”. In: *International Conference on Learning Representations*.
- Luo, Zhixing, Mengyang Liu, and Andrew Lim (2019). “A Two-Phase Branch-and-Price-and-Cut for a Dial-a-Ride Problem in Patient Transportation”. In: *Transportation Science* 53.1, pp. 113–130.
- Ma, Yining et al. (2022). “Efficient Neural Neighborhood Search for Pickup and Delivery Problems”. In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 4776–4784.
- Madkour, Amgad, Walid G. Aref, Faizan Ur Rehman, Mohamed Abdur Rahman, and Saleh Basalamah (2017). *A Survey of Shortest-Path Algorithms*.
- Madsen, Oli B. G., Hans F. Ravn, and Jens Moberg Rygaard (1995). “A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives”. In: *Annals of Operations Research* 60.1, pp. 193–208.
- Mahon, A. G. Mac (2011). “Non-emergency patient transport: an integral part of accessible comprehensive health care”. In: *SAMJ: South African Medical Journal* 101.10, pp. 684–685.
- Malagodi, Laura, Ettore Lanzarone, and Andrea Matta (2021). “Home care vehicle routing problem with chargeable overtime and strict and soft preference matching”. In: *Health Care Management Science* 24.1, pp. 140–159.
- Malheiros, Igor, Rodrigo Ramalho, Bruno Passeti, Teobaldo Bulhões, and Anand Subramanian (2021). “A hybrid algorithm for the multi-depot heterogeneous dial-a-ride problem”. In: *Computers & Operations Research* 129, p. 105196.
- Malitsky, Yuri, Ashish Sabharwal, Horst Samulowitz, and Meinolf Sellmann (2013). *Algorithm Portfolios Based on Cost-Sensitive Hierarchical Clustering*.
- Mańdziuk, Jacek and Maciej Świechowski (2017). “UCT in Capacitated Vehicle Routing Problem with Traffic Jams”. In: *Information Sciences* 406.
- Mankowska, Dorota Slawa, Frank Meisel, and Christian Bierwirth (2014). “The home health care routing and scheduling problem with interdependent services”. In: *Health Care Management Science* 17.1, pp. 15–30.
- Mapotempo (n.d.). *Mapotempo - Route Planning, Route Optimization, Live Tracking*.
- Marrauld, Laurie, Thomas Rambaud, Mathis Egnell, and Jean-Noël Geist (2023). *Décarboner la santé pour soigner durablement : édition 2023 du rapport du Shift Project*.
- Martinez-Estudillo, A.C., C. Hervas-Martinez, F.J. Martinez-Estudillo, and N. Garcia-Pedrajas (2006). “Hybridization of evolutionary algorithms and local search by means of a clustering method”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 36.3, pp. 534–545.
- Masmoudi, Mohamed Amine, Kris Braekers, Malek Masmoudi, and Abdelaziz Dammak (2017). “A hybrid Genetic Algorithm for the Heterogeneous Dial-A-Ride Problem”. In: *Computers & Operations Research* 81, pp. 1–13.
- Masmoudi, Mohamed Amine, Manar Hosny, Kris Braekers, and Abdelaziz Dammak (2016). “Three effective metaheuristics to solve the multi-depot multi-trip heterogeneous dial-a-ride problem”. In: *Transportation Research Part E: Logistics and Transportation Review* 96, pp. 60–80.

- Masmoudi, Mohamed Amine, Manar Hosny, Emrah Demir, Konstantinos N. Genikomsakis, and Naoufel Cheikhrouhou (2018). “The dial-a-ride problem with electric vehicles and battery swapping stations”. In: *Transportation Research Part E: Logistics and Transportation Review* 118, pp. 392–420.
- Masson, Renaud, Fabien Lehuédé, and Olivier Péton (2014). “The Dial-A-Ride Problem with Transfers”. In: *Computers & Operations Research* 41, pp. 12–23.
- Mauri, Geraldo Regis, Luiz Antonio, and Nogueira Lorena (2009). “Customers’ satisfaction in a dial-a-ride problem”. In: *IEEE Intelligent Transportation Systems Magazine* 1.3, pp. 6–14.
- Mei-Ko, Kwan (1962). “Graphic programming using odd or even points”. In: *Chinese Math* 1, pp. 237–277.
- Melachrinoudis, Emanuel, Ahmet B. Ilhan, and Hokey Min (2007). “A dial-a-ride problem for client transportation in a health-care organization”. In: *Computers & Operations Research. Logistics of Health Care Management* 34.3, pp. 742–759.
- Melachrinoudis, Emanuel and Hokey Min (2011). “A tabu search heuristic for solving the multi-depot, multi-vehicle, double request dial-a-ride problem faced by a healthcare organisation”. In: *International Journal of Operational Research* 10.2, pp. 214–239.
- Menger, Karl (1932). “Das botenproblem”. In: *Ergebnisse eines mathematischen kolloquiums* 2.4, pp. 11–12.
- Metropolis, Nicholas, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller (1953). “Equation of State Calculations by Fast Computing Machines”. In: *The Journal of Chemical Physics* 21.6, pp. 1087–1092.
- Miskolczi, Márk, Dávid Földes, András Munkácsy, and Melinda Jászberényi (2021). “Urban mobility scenarios until the 2030s”. In: *Sustainable Cities and Society* 72, p. 103029.
- Mladenović, N. and P. Hansen (1997). “Variable neighborhood search”. In: *Computers & Operations Research* 24.11, pp. 1097–1100.
- Moccia, Luigi, Jean-François Cordeau, and Gilbert Laporte (2012). “An incremental tabu search heuristic for the generalized vehicle routing problem with time windows”. In: *Journal of the Operational Research Society* 63.2, pp. 232–244.
- Moghdani, Reza, Khodakaram Salimifard, Emrah Demir, and Abdelkader Benyettou (2021). “The green vehicle routing problem: A systematic literature review”. In: *Journal of Cleaner Production* 279, p. 123691.
- Molenbruch, Yves, Kris Braekers, and An Caris (2017a). “Benefits of horizontal cooperation in dial-a-ride services”. In: *Transportation Research Part E: Logistics and Transportation Review* 107, pp. 97–119.
- Molenbruch, Yves, Kris Braekers, and An Caris (2017b). “Typology and literature review for dial-a-ride problems”. In: *Annals of Operations Research* 259.1, pp. 295–325.
- Molenbruch, Yves, Kris Braekers, An Caris, and Greet Vanden Berghe (2017c). “Multi-directional local search for a bi-objective dial-a-ride problem in patient transportation”. In: *Computers & Operations Research* 77, pp. 58–71.
- Molenbruch, Yves, Kris Braekers, Ohad Eisenhandler, and Mor Kaspi (2023). “The Electric Dial-a-Ride Problem on a Fixed Circuit”. In: *Transportation Science* 57.3, pp. 594–612.
- Moll, Robert, Andrew Barto, Theodore Perkins, and Richard S Sutton (1998). “Learning Instance-Independent Value Functions to Enhance Local Search”. In: *Advances in Neural Information Processing Systems* 11.
- Montoya, Alejandro, Christelle Guéret, Jorge E. Mendoza, and Juan G. Villegas (2016). “A multi-space sampling heuristic for the green vehicle routing problem”. In: *Transportation Research Part C: Emerging Technologies* 70, pp. 113–128.
- Moons, Karen, Geert Waeyenbergh, and Liliane Pintelon (2019). “Measuring the logistics performance of internal hospital supply chains – A literature study”. In: *Omega* 82, pp. 205–217.
- Mortazavi, Amir, Milad Ghasri, Hossein Haghshenas, and Tapabrata Ray (2022). “Adaptive Logit Models for Constructing Feasible Solution for the Dial-a-Ride Problem”. In: *SSRN Electronic Journal*.
- Mosquera Varela, Anderson Fabian, Lorraine Trilling, and Thibaud Monteiro (2017). “Managing the accompanying path of people with disabilities in a medico-social network: a method to create a process model based on autonomous actors point of views,” in: *20th IFAC World Congress of the International Federation of Automatic Control (IFAC 2017)*. Toulouse, France.

- Muelas, Santiago, Antonio LaTorre, and José-María Peña (2013). “A variable neighborhood search algorithm for the optimization of a dial-a-ride problem in a large city”. In: *Expert Systems with Applications* 40.14, pp. 5516–5531.
- Münzel, Thomas, Mette Sørensen, and Andreas Daiber (2021). “Transportation noise pollution and cardiovascular disease”. In: *Nature Reviews Cardiology* 18.9, pp. 619–636.
- Nazari, MohammadReza, Afshin Oroojlooy, Lawrence Snyder, and Martin Takac (2018). “Reinforcement Learning for Solving the Vehicle Routing Problem”. In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc.
- Osorio, Geovanny, Thibaud Monteiro, Lorraine Trilling, and Frédéric Albert (2017). “Multi-criteria assignment policies to improve global effectiveness of medico-social service sector”. In: *Engineering Applications of Artificial Intelligence* 61, pp. 21–34.
- Owens, Avalon C. S. et al. (2020). “Light pollution is a driver of insect declines”. In: *Biological Conservation* 241, p. 108259.
- Paquette, Julie, François Bellavance, Jean-François Cordeau, and Gilbert Laporte (2012). “Measuring quality of service in dial-a-ride operations: the case of a Canadian city”. In: *Transportation* 39.3, pp. 539–564.
- Paquette, Julie, Jean-François Cordeau, and Gilbert Laporte (2009). “Quality of service in dial-a-ride operations”. In: *Computers & Industrial Engineering* 56.4, pp. 1721–1734.
- Paquette, Julie, Jean-François Cordeau, Gilbert Laporte, and Marta M. B. Pascoal (2013). “Combining multicriteria analysis and tabu search for dial-a-ride problems”. In: *Transportation Research Part B: Methodological* 52, pp. 1–16.
- Park, Junheung and Kyoung-Yun Kim (2017). “Meta-modeling using generalized regression neural network and particle swarm optimization”. In: *Applied Soft Computing* 51, pp. 354–369.
- Parragh, Sophie N. (2011). “Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem”. In: *Transportation Research Part C: Emerging Technologies*. Freight Transportation and Logistics (selected papers from ODYSSEUS 2009 - the 4th International Workshop on Freight Transportation and Logistics) 19.5, pp. 912–930.
- Parragh, Sophie N., Jean-François Cordeau, Karl F. Doerner, and Richard F. Hartl (2012). “Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints”. In: *OR Spectrum* 34.3, pp. 593–633.
- Parragh, Sophie N., Karl F. Doerner, and Richard F. Hartl (2008). “A survey on pickup and delivery problems”. In: *Journal für Betriebswirtschaft* 58.2, pp. 81–117.
- Parragh, Sophie N., Karl F. Doerner, and Richard F. Hartl (2010). “Variable neighborhood search for the dial-a-ride problem”. In: *Computers & Operations Research* 37.6, pp. 1129–1138.
- Parragh, Sophie N., Karl F. Doerner, Richard F. Hartl, and Xavier Gandibleux (2009). “A heuristic two-phase solution approach for the multi-objective dial-a-ride problem”. In: *Networks* 54.4, pp. 227–242.
- Parragh, Sophie N., Jorge Pinho De Sousa, and Bernardo Almada-Lobo (2015). “The Dial-a-Ride Problem with Split Requests and Profits”. In: *Transportation Science* 49.2, pp. 311–334.
- Parragh, Sophie N. and Verena Schmid (2013). “Hybrid column generation and large neighborhood search for the dial-a-ride problem”. In: *Computers & Operations Research* 40.1, pp. 490–497.
- Pimenta, Victor, Alain Quilliot, Hélène Toussaint, and Daniele Vigo (2017). “Models and algorithms for reliability-oriented Dial-a-Ride with autonomous electric vehicles”. In: *European Journal of Operational Research* 257.2, pp. 601–613.
- Pinson, Corentin, H. Murat Afsar, and Caroline Prodhon (2016). “Heuristic Approaches to solve a Generalized Dial-a-Ride Problem Applied to Car-pooling”. In: *IFAC-PapersOnLine*. 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2016 49.12, pp. 1187–1191.
- Pisinger, David and Stefan Ropke (2007). “A general heuristic for vehicle routing problems”. In: *Computers & Operations Research* 34.8, pp. 2403–2435.
- Pisinger, David and Stefan Ropke (2019). “Large Neighborhood Search”. In: *Handbook of Metaheuristics*. Ed. by Michel Gendreau and Jean-Yves Potvin. International Series in Operations Research & Management Science. Cham: Springer International Publishing, pp. 99–127.
- Posada, Marcus, Henrik Andersson, and Carl H. Häll (2017). “The integrated dial-a-ride problem with timetabled fixed route service”. In: *Public Transport* 9.1, pp. 217–241.

- Psarafitis, Harilaos (1980). “A Dynamic Programming Solution to the Single Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem”. In: *Transportation Science* 14, pp. 130–154.
- Qu, Yuan and Jonathan F. Bard (2013). “The heterogeneous pickup and delivery problem with configurable vehicle capacity”. In: *Transportation Research Part C: Emerging Technologies* 32, pp. 1–20.
- Qu, Yuan and Jonathan F. Bard (2015). “A Branch-and-Price-and-Cut Algorithm for Heterogeneous Pickup and Delivery Problems with Configurable Vehicle Capacity”. In: *Transportation Science* 49.2, pp. 254–270.
- Ray, Susmita (2019). “A Quick Review of Machine Learning Algorithms”. In: *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*. 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), pp. 35–39.
- Reinhardt, Line Blander, Tommy Clausen, and David Pisinger (2013). “Synchronized dial-a-ride transportation of disabled passengers at airports”. In: *European Journal of Operational Research* 225.1, pp. 106–117.
- Rekiek, Brahim, Alain Delchambre, and Hussain Aziz Saleh (2006). “Handicapped Person Transportation: An application of the Grouping Genetic Algorithm”. In: *Engineering Applications of Artificial Intelligence* 19.5, pp. 511–520.
- Riley, Connor, Antoine Legrain, and Pascal Van Hentenryck (2019). “Column Generation for Real-Time Ride-Sharing Operations”. In: *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Ed. by Louis-Martin Rousseau and Kostas Stergiou. Vol. 11494. Cham: Springer International Publishing, pp. 472–487.
- Rist, Yannik and Michael A. Forbes (2021). “A New Formulation for the Dial-a-Ride Problem”. In: *Transportation Science* 55.5, pp. 1113–1135.
- Ritzinger, Ulrike, Jakob Puchinger, and Richard F. Hartl (2016). “Dynamic programming based metaheuristics for the dial-a-ride problem”. In: *Annals of Operations Research* 236.2, pp. 341–358.
- Ritzinger, Ulrike, Jakob Puchinger, Christian Rudloff, and Richard F. Hartl (2022). “Comparison of anticipatory algorithms for a dial-a-ride problem”. In: *European Journal of Operational Research* 301.2, pp. 591–608.
- Robinson, Victoria, Vivek Goel, Russell D. MacDonald, and Doug Manuel (2009). “Inter-Facility Patient Transfers in Ontario: Do You Know What Your Local Ambulance Is Being Used For?”. In: *Healthcare Policy* 4.3, pp. 53–66.
- Ropke, Stefan and Jean-François Cordeau (2009). “Branch and Cut and Price for the Pickup and Delivery Problem with Time Windows”. In: *Transportation Science* 43.3, pp. 267–286.
- Ropke, Stefan, Jean-François Cordeau, and Gilbert Laporte (2007). “Models and branch-and-cut algorithms for pickup and delivery problems with time windows”. In: *Networks* 49.4, pp. 258–272.
- Ropke, Stefan and David Pisinger (2006). “An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows”. In: *Transportation Science* 40.4, pp. 455–472.
- Rosin, Christopher D (2011). “Nested rollout policy adaptation for Monte Carlo tree search”. In: *Ijcai*. Vol. 2011, pp. 649–654.
- Sabharwal, Ashish, Horst Samulowitz, and Chandra Reddy (2012). “Guiding Combinatorial Optimization with UCT”. In: *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*. Ed. by Nicolas Beldiceanu, Narendra Jussien, and Éric Pinson. Red. by David Hutchison et al. Vol. 7298. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 356–361.
- Santini, Alberto, Stefan Ropke, and Lars Magnus Hvattum (2018). “A comparison of acceptance criteria for the adaptive large neighbourhood search metaheuristic”. In: *Journal of Heuristics* 24.5, pp. 783–815.
- Savelsbergh, Martin W. P. (1992). “The Vehicle Routing Problem with Time Windows: Minimizing Route Duration”. In: *ORSA Journal on Computing* 4.2, pp. 146–154.
- Schilde, M., K. F. Doerner, and R. F. Hartl (2011). “Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports”. In: *Computers & Operations Research* 38.12, pp. 1719–1730.
- Schilde, M., K. F. Doerner, and R. F. Hartl (2014). “Integrating stochastic time-dependent travel speed in solution methods for the dynamic dial-a-ride problem”. In: *European Journal of Operational Research* 238.1, pp. 18–30.

- Schrumpf, Gerhard, Johannes Schneider, Hermann Stamm-Wilbrandt, and Gunter Dueck (2000). “Record Breaking Optimization Results Using the Ruin and Recreate Principle”. In: *Journal of Computational Physics* 159.2, pp. 139–171.
- Shaw, Paul (1998). “Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems”. In: *Principles and Practice of Constraint Programming — CP98*. Ed. by Michael Maher and Jean-Francois Puget. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 417–431.
- Sikka, Neal and Gregg Margolis (2005). “Understanding diversity among prehospital care delivery systems around the world”. In: *Emergency Medicine Clinics of North America* 23.1, pp. 99–114.
- Smith, Kate A. (1999). “Neural Networks for Combinatorial Optimization: A Review of More Than a Decade of Research”. In: *INFORMS Journal on Computing* 11.1, pp. 15–34.
- Solomonoff, Raymond J (1957). “An inductive inference machine”. In: *IRE Convention Record, Section on Information Theory*. Vol. 2. Institute of Radio Engineers New York, pp. 56–62.
- Song, Heda, Isaac Triguero, and Ender Özcan (2019). “A review on the self and dual interactions between machine learning and optimisation”. In: *Progress in Artificial Intelligence* 8.2, pp. 143–165.
- Souza, André L. S., Marcella Bernardo, Puca H. V. Penna, Jürgen Pannek, and Marccone J. F. Souza (2022). “Bi-objective optimization model for the heterogeneous dynamic dial-a-ride problem with no rejects”. In: *Optimization Letters* 16.1, pp. 355–374.
- Streichert, Felix, Gunnar Stein, Holger Ulmer, and Andreas Zell (2003). “A Clustering Based Niching Method for Evolutionary Algorithms”. In: *Genetic and Evolutionary Computation — GECCO 2003*. Ed. by Erick Cantú-Paz et al. Red. by Gerhard Goos, Juris Hartmanis, and Jan Van Leeuwen. Vol. 2723. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 644–645.
- Su, Yue, Jakob Puchinger, and Nicolas Dupin (2023). “A Deterministic Annealing Local Search for the Electric Autonomous Dial-A-Ride Problem”. In: *European Journal of Operational Research* 309.3, pp. 1091–1111.
- Tang, Jiafu, Yuan Kong, Henry Lau, and Andrew W. H. Ip (2010). “A note on “Efficient feasibility testing for dial-a-ride problems””. In: *Operations Research Letters* 38.5, pp. 405–407.
- Tellez, Oscar (2019). “Optimizing the daily transport for people with disabilities”. PhD thesis.
- Tellez, Oscar, Samuel Vercaene, Fabien Lehuédé, Olivier Péton, and Thibaud Monteiro (2018). “The fleet size and mix dial-a-ride problem with reconfigurable vehicle capacity”. In: *Transportation Research Part C: Emerging Technologies* 91, pp. 99–123.
- Tellez, Oscar, Samuel Vercaene, Fabien Lehuédé, Olivier Péton, and Thibaud Monteiro (2022). “The time-consistent dial-a-ride problem”. In: *Networks* 79.4, pp. 452–478.
- Tellez, Oscar et al. (2020). “A stakeholder oriented approach to the optimization of transports of people with disabilities”. In: *Supply Chain Forum: An International Journal* 21.2, pp. 93–102.
- Thevenin, Simon and Nicolas Zufferey (2019). “Learning Variable Neighborhood Search for a scheduling problem with time windows and rejections”. In: *Discrete Applied Mathematics*. GO X Meeting, Rigi Kaltbad (CH), July 10–14, 2016 261, pp. 344–353.
- Turkeš, Renata, Kenneth Sörensen, and Lars Magnus Hvattum (2021). “Meta-analysis of metaheuristics: Quantifying the effect of adaptiveness in adaptive large neighborhood search”. In: *European Journal of Operational Research* 292.2, pp. 423–442.
- Uthayakumar, R. and S. Priyan (2013). “Pharmaceutical supply chain and inventory management strategies: Optimization for a pharmaceutical company and a hospital”. In: *Operations Research for Health Care* 2.3, pp. 52–64.
- Vachey, Laurent, Agnès Jeannet, Anne Auburtin, Frédéric Varnier, and Claire-Marie Ier Gazane (2012). “Établissements et services pour personnes handicapées: offre et besoins, modalités de financement”. In: *La Documentation française. Inspection générale des finances*.
- Václavík, Roman, Přemysl Šúcha, and Zdeněk Hanzálek (2016). “Roster Evaluation Based on Classifiers for the Nurse Rostering Problem”. In: *Journal of Heuristics* 22.5, pp. 667–697.
- Vera, Jose Manuel and Andres G. Abad (2019). *Deep Reinforcement Learning for Routing a Heterogeneous Fleet of Vehicles*.
- Vinyals, Oriol, Meire Fortunato, and Navdeep Jaitly (2015). “Pointer Networks”. In: *Advances in Neural Information Processing Systems*. Vol. 28. Curran Associates, Inc.

- Waledzik, Karol, Jacek Mandziuk, and Slawomir Zadrozny (2014). “Proactive and reactive risk-aware project scheduling”. In: *2014 IEEE Symposium on Computational Intelligence for Human-like Intelligence (CIHLI)*. 2014 IEEE Symposium on Computational Intelligence for Human-like Intelligence (CIHLI). Orlando, FL: IEEE, pp. 1–8.
- Walker, Tony R. et al. (2019). “Chapter 27 - Environmental Effects of Marine Transportation”. In: *World Seas: An Environmental Evaluation (Second Edition)*. Ed. by Charles Sheppard. Academic Press, pp. 505–530.
- Wang, Hua, Cuiqin Ma, and Lijuan Zhou (2009). “A Brief Review of Machine Learning and Its Application”. In: *2009 International Conference on Information Engineering and Computer Science*. 2009 International Conference on Information Engineering and Computer Science, pp. 1–4.
- Wieczorek, Estera, Silvia Evers, Ewa Kocot, Christoph Sowada, and Milena Pavlova (2022). “Assessing Policy Challenges and Strategies Supporting Informal Caregivers in the European Union”. In: *Journal of Aging & Social Policy* 34.1, pp. 145–160.
- Wolfler Calvo, Roberto and Nora Touati-Moungla (2011). “A Matheuristic for the Dial-a-Ride Problem”. In: *Network Optimization*. Ed. by Julia Pahl, Torsten Reiners, and Stefan Voß. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 450–463.
- Wong, K.I., A.F. Han, and C.W. Yuen (2014). “On dynamic demand responsive transport services with degree of dynamism”. In: *Transportmetrica A: Transport Science* 10.1, pp. 55–73.
- Wong, Li-Pei and Shin Siang Choong (2015). “A Bee Colony Optimization algorithm with Frequent-closed-pattern-based Pruning Strategy for Traveling Salesman Problem”. In: *2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*. 2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI), pp. 308–314.
- Worrall, Philip and Thierry J. Chausalet (2015). “A structured review of long-term care demand modelling”. In: *Health Care Management Science* 18.2, pp. 173–194.
- Wu, Yaoxin, Wen Song, Zhiguang Cao, Jie Zhang, and Andrew Lim (2020). “Learning Improvement Heuristics for Solving Routing Problems”. In: *arXiv:1912.05784 [cs]*.
- Xiang, Zhihai, Chengbin Chu, and Haoxun Chen (2006). “A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints”. In: *European Journal of Operational Research* 174.2, pp. 1117–1139.
- Xiang, Zhihai, Chengbin Chu, and Haoxun Chen (2008). “The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments”. In: *European Journal of Operational Research* 185.2, pp. 534–551.
- Xu, L., F. Hutter, H. H. Hoos, and K. Leyton-Brown (2008). “SATzilla: Portfolio-based Algorithm Selection for SAT”. In: *Journal of Artificial Intelligence Research* 32, pp. 565–606.
- Yalcinoz, Tankut and Halis Altun (2001). “Power economic dispatch using a hybrid genetic algorithm”. In: *Power Engineering Review, IEEE* 21, pp. 59–60.
- Yu, James J. Q., Wen Yu, and Jiatao Gu (2019). “Online Vehicle Routing With Neural Combinatorial Optimization and Deep Reinforcement Learning”. In: *IEEE Transactions on Intelligent Transportation Systems* 20.10, pp. 3806–3817.
- Zhang, Huaxiang and Jing Lu (2008). “Adaptive evolutionary programming based on reinforcement learning”. In: *Information Sciences* 178.4, pp. 971–984.
- Zhang, Jun, Henry Shu-Hung Chung, and Wai-Lun Lo (2007). “Clustering-Based Adaptive Crossover and Mutation Probabilities for Genetic Algorithms”. In: *IEEE Transactions on Evolutionary Computation* 11.3, pp. 326–335.
- Zhang, Xingyi, Ye Tian, Ran Cheng, and Yaochu Jin (2018). “A Decision Variable Clustering-Based Evolutionary Algorithm for Large-Scale Many-Objective Optimization”. In: *IEEE Transactions on Evolutionary Computation* 22.1, pp. 97–112.
- Zhang, Zhenzhen, Mengyang Liu, and Andrew Lim (2015). “A memetic algorithm for the patient transportation problem”. In: *Omega* 54, pp. 60–71.
- Zidi, Issam, Khaled Mesghouni, Kamel Zidi, and Khaled Ghedira (2012). “A multi-objective simulated annealing for the multi-criteria dial a ride problem”. In: *Engineering Applications of Artificial Intelligence* 25.6, pp. 1121–1131.

Appendix

A.1 List of extended arc metrics

| numerators | denominator |
|---|---|
| number of apparitions/insertions/removals | number of iterations, number of feasibility tests |
| number of feasibility tests | number of iterations, total number of feasibility tests |
| number of feasibility tests | number of iterations, number of feasibility tests |

Table A.1: list of extended arc metrics

Each extended arc metric corresponds to the ratio between two simple metrics. Not every combination are computed. The computed combination are presented in Table A.1. The metric “number of feasibility tests” is declined in four sub-metrics to improve the readability. These four sub-metrics are the number of preliminary tests computed, the number of preliminary tests succeeded, the number of scheduling tests computed, and the number of scheduling tests succeeded. Each row corresponds to a set of numerators and a set of denominators. Metrics are computed from every combination of numerator and denominator. For example, the first row gives metrics such as the ratio between the number of insertions and the number of scheduling tests computed.

A.2 List of notations for the Com-DARP

- sets:
 - \mathcal{A} : set of arcs
 - \mathcal{D} : set of drop-off nodes
 - \mathcal{E} : set of pairs of starting and ending depots
 - \mathcal{E}_k : set of pairs of starting and ending depots of the vehicle k
 - \mathcal{G} : complete directed graph
 - \mathcal{K} : set of heterogeneous vehicles

- \mathcal{N} : set of nodes
 - \mathcal{O}^+ : set of starting depot nodes
 - \mathcal{O}^- : set of ending depot nodes
 - \mathcal{O}_k^+ : set of starting depot nodes of the vehicle k
 - \mathcal{O}_k^- : set of ending depot nodes of the vehicle k
 - \mathcal{P} : set of pickup nodes
 - \mathcal{R} : set of requests
 - \mathcal{R}_{ins} : set of inserted requests
 - \mathcal{R}_{unins} : set of uninserted requests
 - \mathcal{R}_u : set of requests of the user u
 - \mathcal{S}^+ : set of insertion operators
 - \mathcal{S}^- : set of removal operators
 - \mathcal{U} : set of users
- data:
 - A_n : beginning of the time windows at node n
 - B_n : end of the time windows at node n
 - C_k^d : cost related to the ride time for the vehicle k
 - C_k^f : fixed cost of the vehicle k
 - C_k^t : cost related to the distance for the vehicle k
 - D_{ij} : travel distance from node i to node j
 - D_r : drop-off node of the request r
 - F_n : forward time slack at node n
 - H : time horizon
 - $\overline{L}_{o_k^+}$: maximum route ride time of the vehicle k when leaving from the depot node o_k^+
 - \overline{L}_k : maximum total ride time of the vehicle k
 - \overline{L}_r : maximum ride time of the request r
 - \overline{L}_u : maximum ride time of the user u
 - P_r : pickup node of the request r
 - Q_n : load variation at node n
 - \overline{Q}_k : capacity of the vehicle k
 - S_n : service duration at node n
 - T_{ij} : travel duration from node i to node j
 - W : total waiting time on a route
 - variables:
 - l_i^k : a positive integer variable which corresponds to the number of users in the vehicle k after leaving the node i
 - w_i^k : a positive continuous variable which corresponds to the service time of the vehicle k at node i
 - w_i : a positive continuous variable which corresponds to the service time at node i
 - x_{ij}^k : a binary variable which is equal to 1 if the vehicle k uses the arc (i, j) , and 0 otherwise

- y^k : a binary variable which is equal to 1 if the vehicle k is used
- z_r : a positive continuous variable which corresponds to the ride time of request r
- parameters:
 - δ : minimum destruction size in percentage of inserted requests for the LNS
 - Δ : maximum destruction size in percentage of inserted requests for the LNS
 - δ_{small} : minimum destruction size in percentage of inserted requests for a small iteration of the SLNS
 - Δ_{small} : maximum destruction size in percentage of inserted requests for a small iteration of the SLNS
 - δ_{large} : minimum destruction size in percentage of inserted requests for a large iteration of the SLNS
 - Δ_{large} : maximum destruction size in percentage of inserted requests for a large iteration of the SLNS
 - ξ : random number used in the historical node-pair removal operator
 - ϕ : proportion of requests to remove
 - χ : record-to-record acceptance percentage
 - ψ : roulette wheel parameter used in the historical node-pair removal operator
 - ω : maximum number of small iterations since the last new best solution or the last large destruction

A.3 Precedences for the Com-DARP

This section demonstrates the results presented in Section 5.4.3. To facilitate the demonstration, the following two propositions are used:

Proposition 1.

$$\min\{A, B\} = -\max\{-A, -B\}, \quad \forall A, B \in \mathbb{R} \quad (\text{A.1})$$

Proof. If $A \geq B$, $\min\{A, B\} = B$ and $-\max\{-A, -B\} = B$.

If $A < B$, $\min\{A, B\} = A$ and $-\max\{-A, -B\} = A$. □

A.3.1 Pickup/pickup

Given two pickup nodes (uhP) and (whP), (whP) is a predecessor of (uhP) if it is not possible to leave (uhP) at the earliest and arrive in time at (whP). According to equation (5.18), this is the case when:

$$0 < a_{uhP} + S_{uhP} + T_{uhP,whP} - b_{whP}$$

By replacing a_{uhP} using equation 5.24 and b_{whP} using equation 5.25, it becomes:

$$0 < \max\{A_{uhP}; A_{uhD} - S_{uhP} - \overline{l_{uh}}\} + S_{uhP} + T_{uhP,whP} - B_{whP}$$

This inequality allows us to define three cases:

- the node (whP) is always a predecessor of (uhP) if:

$$0 < \max\{A_{uhP}; A_{uhD} - S_{uhP} - \max(\overline{l_{uh}})\} + S_{uhP} + T_{uhP,whP} - B_{whP}$$

By replacing $\max(\overline{l_{uh}})$ by $\overline{L_{uh}}$, it becomes:

$$0 < \max\{A_{uhP}; A_{uhD} - S_{uhP} - \overline{L_{uh}}\} + S_{uhP} + T_{uhP,whP} - B_{whP}$$

According to equation 5.20: $A_{uhP} \geq A_{uhD} - S_{uhP} - \overline{L_{uh}}$. Thus, the inequality can be simplified to:

$$0 < A_{uhP} + S_{uhP} + T_{uhP,whP} - B_{whP} \quad (\text{A.2})$$

- the node (whP) is never a predecessor of (uhP) if:

$$0 \geq \max\{A_{uhP}; A_{uhD} - S_{uhP} - \min(\overline{l_{uh}})\} + S_{uhP} + T_{uhP,whP} - B_{whP}$$

By replacing $\min(\overline{l_{uh}})$ by $T_{uhP,uhD}$, it becomes:

$$0 \geq \max\{A_{uhP}; A_{uhD} - S_{uhP} - T_{uhP,uhD}\} + S_{uhP} + T_{uhP,whP} - B_{whP}$$

According to equation 5.22, $A_{uhD} - S_{uhP} - T_{uhP,uhD} \geq A_{uhP}$. Thus, the inequality can be simplified to:

$$0 \geq A_{uhD} - T_{uhP,uhD} + T_{uhP,whP} - B_{whP} \quad (\text{A.3})$$

- the node (whP) becomes a predecessor of (uhP) when:

$$0 < \max\{A_{uhP}; A_{uhD} - S_{uhP} - \overline{l_{uh}}\} + S_{uhP} + T_{uhP,whP} - B_{whP}$$

According to equation 5.19, $\overline{l_{uh}} = \min\{\overline{L_{uh}}; \overline{L_u} - l_{uh'}\}$. Thus using Proposition 1, the previous inequality can be written as:

$$0 < \max\{A_{uhP}; A_{uhD} - S_{uhP} + \max\{-\overline{L_{uh}}; l_{uh'} - \overline{L_u}\}\} + S_{uhP} + T_{uhP,whP} - B_{whP}$$

It can be written as:

$$0 < \max\{A_{uhP}; A_{uhD} - S_{uhP} - \overline{L_{uh}}; A_{uhD} - S_{uhP} + l_{uh'} - \overline{L_u}\} + S_{uhP} + T_{uhP,whP} - B_{whP}$$

According to equation 5.20: $A_{uhP} \geq A_{uhD} - S_{uhP} - \overline{L_{uh}}$. Thus, the inequality can be simplified to:

$$0 < \max\{A_{uhP}; A_{uhD} - S_{uhP} + l_{uh'} - \overline{L_u}\} + S_{uhP} + T_{uhP,whP} - B_{whP}$$

In the case where $A_{uhP} \geq A_{uhD} - S_{uhP} + l_{uh'} - \overline{L_u}$, the precedence does not depend on $l_{uh'}$. It corresponds to the case described by equation (A.2). The opposition case where $A_{uhP} < A_{uhD} - S_{uhP} + l_{uh'} - \overline{L_u}$, gives:

$$0 < A_{uhD} - S_{uhP} + l_{uh'} - \overline{L_u} + S_{uhP} + T_{uhP,whP} - B_{whP}$$

Thus, the inequality can be written as:

$$l_{uh'} > B_{whP} - A_{uhD} + \overline{L_u} - T_{uhP,whP} \quad (\text{A.4})$$

A.3.2 Drop-off/drop-off

Given two drop-off nodes (uhD) and (whD) , (whD) is a predecessor of (uhD) if it is not possible to leave (uhD) at the earliest and arrive in time at (whD) . According to equation (5.18) and using equation (5.26) and (5.27), this is the case when

$$0 < A_{uhD} + S_{uhD} + T_{uhD,whD} - \min\{B_{whD}; B_{whP} + S_{whP} + \overline{l_{wh}}\}$$

This inequality allows us to define three cases:

- the node (whD) is always a predecessor of (uhD) , knowing that $\max(\overline{l_{wh}}) = \overline{L_{wh}}$ and using equation (5.23), if:

$$0 < A_{uhD} + S_{uhD} + T_{uhD,whD} - B_{whD}$$

- the node (whD) is never a predecessor of (uhD) , knowing that $\min(\overline{l_{wh}}) = T_{whP,whD}$ and using equation (5.21), if:

$$0 \geq A_{uhD} + S_{uhD} + T_{uhD,whD} - B_{whP} - S_{whP} - T_{whP,whD}$$

- the node (whD) becomes a predecessor of (uhD) , using equations (5.19) and (5.23), if:

$$l_{wh'} > \overline{L_w} + B_{whP} + S_{whP} - A_{uhD} - S_{uhD} - T_{uhD,whD}$$

A.3.3 Pickup/drop-off

Given a pickup node (uhP) and a drop-off node (whD), (whD) is a predecessor of (uhP) if it is not possible to leave (uhP) at the earliest and arrive in time at (whD). According to equation (5.18) and using equation (5.24) and (5.27), this is the case when

$$0 < \max\{A_{uhP}; A_{uhD} - S_{uhP} - \overline{l_{uh}}\} + S_{uhP} + T_{uhP,whD} \\ - \min\{B_{whD}; B_{whP} + S_{whP} + \overline{l_{wh}}\}$$

This inequality allows us to define three cases:

- the node (whD) is always a predecessor of (uhP), knowing that $\max(\overline{l_{uh}}) = \overline{L_{uh}}$, $\max(\overline{l_{wh}}) = \overline{L_{wh}}$ and using equations (5.20) and (5.23), if:

$$0 < A_{uhP} + S_{uhP} + T_{uhP,whD} - B_{whD}$$

- the node (whD) is never a predecessor of (uhP), knowing that $\min(\overline{l_{uh}}) = T_{uhP,uhD}$, $\min(\overline{l_{wh}}) = T_{whP,whD}$ and using equation (5.21) and (5.22), if:

$$0 \geq A_{uhD} - T_{uhP,uhD} + T_{uhP,whD} - B_{whP} - S_{whP} - T_{whP,whD}$$

- the node (whD) becomes a predecessor of (uhP), using equations (5.19), (5.20) and (5.23) and Proposition 1, if:

$$0 < \max\{A_{uhP}; A_{uhD} - S_{uhP} - \overline{L_u} + l_{uh'}\} + S_{uhP} + T_{uhP,whD} \\ - \min\{B_{whD}; B_{whP} + S_{whP} + \overline{L_w} - l_{wh'}\}$$

This results in three cases where (whD) becomes a predecessor of (uhP):

$$\begin{cases} l_{uh'} > \overline{L_u} - A_{uhD} - T_{uhP,whD} + B_{whD} \\ l_{wh'} > \overline{L_w} + B_{whP} + S_{whP} - A_{uhP} - S_{uhP} - T_{uhP,whD} \\ l_{uh'} + l_{wh'} > \overline{L_u} + \overline{L_w} - A_{uhD} - T_{uhP,whD} + B_{whP} + S_{whP} \end{cases}$$

A.3.4 Drop-off/pickup

Given a drop-off node (uhD) and a pickup node (whP), (whP) is a predecessor of (uhD) if it is not possible to leave (uhD) at the earliest and arrive in time at (whP). According to equation (5.18) and using equation (5.25) and (5.26), this is the case when:

$$0 < A_{uhD} + S_{uhD} + T_{uhD,whP} - B_{whP}$$

This inequality allows to define two cases:

- the node (whP) is always a predecessor of (uhD) if:

$$0 < A_{uhD} + S_{uhD} + T_{uhD,whP} - B_{whP}$$

- the node (whP) is never a predecessor of (uhD) if:

$$0 \geq A_{uhD} + S_{uhD} + T_{uhD,whP} - B_{whP}$$

A.4 List of notations for the ADARP

- sets:
 - \mathcal{D} : set of drop-off nodes
 - \mathcal{G} : set of locations
 - \mathcal{I}_u : set of itineraries of user u
 - \mathcal{K} : set of vehicles
 - \mathcal{N} : set of nodes
 - $\mathcal{N}(\omega)$: ordered set of nodes visited in route ω
 - \mathcal{P} : set of pickup nodes
 - \mathcal{O}^+ : set of starting depots
 - \mathcal{O}^- : set of ending depots
 - \mathcal{R} : set of resources
 - \mathcal{U} : set of users
 - \mathcal{U}^{no} : set of uninserted users
 - $\mathcal{UI}(\omega)$: set of tuple (u, i) , where u is the user and $i \in \mathcal{I}_u$ the itinerary chosen for user $u \in \mathcal{U}$ in route ω
 - Ω : set of all possible routes
 - Ω' : set of routes for the route selection problem
- data:
 - A_n : beginning of the time windows at node n
 - B_n : end of the time windows at node n
 - C_r^{ext} : cost to extend resource r per unit of resource
 - C_k^d : distance-related cost of vehicle k
 - C_k^f : fixed cost of vehicle k
 - C^t : duration-related cost of any vehicle
 - $D(u, i)$: drop-off of itinerary i of user u
 - $\bar{E}_{g,r}^{ext}$: quantity of resource r available at geographical locality g with paying for overtime
 - $\bar{E}_{g,r}$: quantity of resource r available at geographical locality g without paying for overtime
 - $E_{u,r}$: quantity of resource r required by user u
 - $F_{u,i}$: insertion cost estimation for user u in itinerary i
 - f_ω : cost of route ω
 - $F(\omega)$: first node visited in route ω
 - $G(n)$: geographical location of node n
 - H_u : maximum ride time of user u
 - i_u : itinerary i of user u
 - $L(\omega)$: last node visited in route ω
 - $P(u, i)$: pickup of itinerary i of user u
 - $Q(\omega, n)$: node visited before node n in route ω
 - S_n : service time at node n
 - \bar{V}_k : capacity of vehicle v

- V_u : load variation corresponding to user u in a vehicle
- $\Gamma_{i,j}$: duration between node i and node j
- $\Delta_{i,j}$: distance between node i and node j
- variables:
 - $e_{g,r}$: a positive continuous variable which corresponds to the quantity of extra resources r that must be spent to ensure that the needs of each user assigned to geographical locality g will be satisfied.
 - $t_{\omega,n}$: a positive continuous variable which corresponds to beginning of service time for node n in route ω
 - v_n : a positive integer variable which corresponds load of the vehicle when leaving the node n
 - x_k : a binary variable which is equal to 1 if vehicle k is selected for the route ω , and 0 otherwise
 - y_ω : a binary variable which is equal to 1 if route ω is selected, and 0 otherwise
 - $z_{u,i}$: a binary variable which is equal to 1 if itinerary i is selected for user u , and 0 otherwise
- parameters:
 - ε : constant positive multiplier for changing the pool size
 - $\underline{\mu}$: minimum percentage of user to remove by the LNS
 - $\overline{\mu}$: maximum percentage of user to remove by the LNS
 - ν : initial number of routes used to solve the route selection problem
 - τ : total time limit
 - τ' : time limit for the MILP solver

A.5 Best-known solutions for the ADARP

Table A.2: Best-Known Solution (BKS)

| Inst. | DTF | Cost (€) |
|--------|------|----------|
| Loire | 1.25 | 1,002.53 |
| | 1.50 | 816.02 |
| | 1.75 | 743.12 |
| Rhône | 1.25 | 1,608.01 |
| | 1.50 | 1,373.32 |
| | 1.75 | 1,281.98 |
| Savoie | 1.25 | 3,214.34 |
| | 1.50 | 2,834.83 |
| | 1.75 | 2,687.22 |

FOLIO ADMINISTRATIF

THESE DE L'INSA LYON, MEMBRE DE L'UNIVERSITE DE LYON

Nom : CHANE-HAÏ

Date de soutenance : 13/12/2023

Prénoms : Timothée, Lucas

Titre : Nouvelles variantes et méthodes de résolution pour les problèmes de transport à la demande, application au transport d'enfants en situation de handicap

Nature : Doctorat

Numéro d'ordre : 2023ISAL0125

Ecole doctorale : Infomaths

Spécialité : Génie industriel

Résumé :

Cette thèse introduit de nouveaux modèles et méthodes de résolution pour les problèmes de transport à la demande (DARP). Ce travail s'applique au transport régulier d'enfants en situation de handicap entre leurs domiciles et leurs lieux de prise en charge. Pour des raisons de coûts et de qualité de service, il doit être effectué aussi efficacement que possible.

Aucune méthode de la littérature ne peut résoudre les problèmes réels car leur taille est trop importante (plusieurs milliers d'utilisateurs). De plus, les recherches se concentrent sur l'organisation des tournées de véhicules. Cependant, l'intégration d'autres éléments gravitant autour du transport serait bénéfique pour les systèmes de santé dans leur ensemble.

Nous apportons des éléments de réponse à ces enjeux dans les trois chapitres principaux de cette thèse.

Premièrement, deux méthodes d'apprentissage automatique sont appliquées : une méthode offline extrait les caractéristiques des bonnes solutions et les utilise pour créer de nouvelles heuristiques ; une méthode online dénommée NRPA construit la meilleure séquence d'utilisateurs à insérer.

Deuxièmement, nous présentons le problème journalier de transport à la demande (Com-DARP). Dans cette variante, chaque utilisateur a un trajet aller le matin, un trajet retour le soir, et un temps de trajet maximal journalier. La dépendance entre les deux demandes de trajet est utilisée pour améliorer le transport à l'échelle de la journée. Nous résolvons le problème avec une métaheuristique de recherche à petit et grand voisinage couplée à un filtre de précédences (SLNS-PF).

Troisièmement, nous introduisons le problème d'affectation et transport à la demande (ADARP). Cette variante élargit le champ d'application du problème de tournées de véhicules en incluant l'affectation des utilisateurs et l'allocation des ressources. Le problème est résolu par une nouvelle matheuristique nommée recherche itérative d'itinéraires (IRS).

Dans chaque chapitre, les résultats expérimentaux sont analysés pour fournir de nouvelles perspectives théoriques et pratiques.

Mots-clés : transport à la demande, transport sanitaire, apprentissage automatique, recherche guidée par la connaissance, nested rollout policy adaptation, déplacement pendulaire, contraintes partagées, précédence, affectation, allocation de ressources, itérative route search, recherche à voisinage large

Laboratoire (s) de recherche : DISP, EA 4570

Directeur de thèse : Thibaud MONTEIRO

Présidente de jury : Sylvie NORRE, professeur des universités, Université de Clermont-Auvergne

Composition du jury :

Maria DI-MASCOLO, directrice de recherche, Grenoble INP CNRS
Samir LOUDNI, professeur, IMT Atlantique
Sylvie NORRE, professeur des universités, Université de Clermont-Auvergne
Dominique FEILLET, professeur, Écoles des mines de Saint-Etienne

Rapporteuse
Rapporteur
Examinatrice
Examinateur

Thibaud MONTEIRO, professeur des universités, INSA Lyon
Samuel VERCRAENE, maître de conférences HDR, INSA Lyon
Céline ROBARDET, professeur des universités, INSA Lyon

Directeur de thèse
Co-directeur de thèse
Co-directrice de thèse